

---

PROGRAMMED GUIDANCE EQUATIONS FOR SKYLARK  
COMMAND MODULE EARTH ORBITAL PROGRAM

---

NAS 9-12330

14 FEBRUARY 1972

BASED ON SKYLARK 48 PROGRAM

Prepared for  
MISSION PLANNING AND ANALYSIS DIVISION  
NATIONAL AERONAUTICS AND SPACE ADMINISTRATION  
MANNED SPACECRAFT CENTER  
HOUSTON, TEXAS

**WORKING PAPER**

## Change Page Record

The original issue of 4900.5-244 represents a complete re-issue of 5520.5-205 ("Programmed Guidance Equations for Colossus 3 Command Module Earth Orbital and Lunar Program," 20 April 1971), updated to the contents of the Skylark 48 Skylab program. Because of the extensive changes required, no special change identification is provided.



## ABSTRACT

The information presented in this document on the Skylark (manned Command Module earth orbital Skylab) 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 segments called "writeups", each of which describes a basic area of guidance computations within the program. In order to assist the user in finding the computations for a particular quantity, summaries of each of the writeups, plus a brief discussion of the program computation flow, have been included. At the end of the document is an alphabetical list of the routines in the writeups, together with their location by page number in the individual writeups.

The program listing from which the "programmed equations" information was generated is identified as "Skylark 48", and is expected to be used in the computer memory for the Skylab missions.

Because of the purposes for which the information in this document was prepared, and the methods used in its production, this material under no circumstances should be considered as definitive information on the Skylark program. Contrary to an analogous document produced by MSC for the LM program, this document has not been approved by NASA management. Since this document has not been approved by NASA, therefore, it should never be used or cited as a reference for the equations in the Skylark computer program.



# CONTENTS

	Page
Introduction . . . . .	1
Summary of Individual Writeups . . . . .	3
Program Computation Flow . . . . .	9
Notation and Terminology . . . . .	31
Major Variables . . . . .	41
Channel Assignments . . . . .	67
Meanings of MODREG Values . . . . .	73
Special Program Routines . . . . .	75
Checklist and Option Codes . . . . .	77
Error Codes . . . . .	79
Job Priorities . . . . .	85
Erasable Memory Prelaunch Load . . . . .	89
Writeups	
Attitude Maneuvers . . . . .	ATTM-1
Boost Computations . . . . .	BOOS-1
Burn Control . . . . .	BURN-1
Conic Routines . . . . .	CONC-1
Coordinate Transformations . . . . .	COOR-1
Data Input/Output . . . . .	DATA-1
Display Interface Routines . . . . .	DINT-1
Display Computations . . . . .	DISP-1
Digital Autopilot Docked Jet Selection . . . . .	DPDS-1
Digital Autopilot Entry Routines . . . . .	DPEN-1
Digital Autopilot Interface Routines . . . . .	DPIR-1

# CONTENTS (Continued)

	Page
Digital Autopilot RCS Routines . . . . .	DPRC-1
Digital Autopilot TVC Routines . . . . .	DPTV-1
Entry Computations . . . . .	ENRY-1
Entry Preparation . . . . .	ENTP-1
General Program Control . . . . .	GENP-1
IMU Computations . . . . .	IMUC-1
Inflight Alignment . . . . .	INFA-1
Mathematical Functions . . . . .	MATH-1
Measurement Incorporation . . . . .	MEAS-1
Minimum Key Rendezvous . . . . .	MINK-1
Noun Definitions . . . . .	NNDF-1
Optics Computations . . . . .	OPTC-1
Orbital Integration . . . . .	ORBI-1
Orbital and Rendezvous Navigation . . . . .	ORVN-1
Prelaunch Alignment . . . . .	PREL-1
Rendezvous Computations . . . . .	REND-1
Steering Computations . . . . .	STER-1
Telemetry . . . . .	TELE-1
Testing Routines . . . . .	TEST-1
Uplink Processing . . . . .	UPLK-1
Verb Definitions . . . . .	VBDF-1
Index of Routines . . . . .	X-1

## Introduction

Under the auspices of TRW Systems MTCP Task A-201 ("Support of Apollo Guidance Document Review"), a review of certain portions of the Apollo Guidance Computer program identified as "Skylark" has been conducted. This program is a considerably modified version of the "Colossus 3" program used on J-series Apollo flights, and is intended for use with the Command Module Computer on Skylab missions.

During similar reviews of the AS-202, AS-204, AS-501, LM-1, Sundisk, Sundance, and Colossus programs, it was found desirable to assemble a set of working-paper information on the equations which seem to have been programmed for these flights. This material has proven to be useful to a few groups associated with these flights, in that it can be used to help bridge the gap between the extreme detail of the program listing and that which appears in the approved computer program documentation. Consequently, the material on the following pages has been assembled in a fashion similar to that done for previous flights, 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. To assist in such a study, the binary scale factors of equation quantities have been included, as well as the program notation employed for constants (the notation for variables follows that, in general, used in the program), and except in a few instances divisions have been made in accordance with the symbolic tags assigned to program steps in the listing. It should be realized that no complete set of approved computer program documentation was available against which the programmed equations could be validated, and in the interests 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 the programmed equations information bears the heading print:

REVISION 048 OF AGC PROGRAM SKYLARK BY NASA 2021115-011,  
18:18 NOV. 11, 1971

The function of virtually all the program steps of interest to the flight is described on the following pages or, for general computer system control (executive and waitlist system, interpretive language, and similar items) in 3420.5-27 (see Notation and Terminology). Only isolated omissions of unreferenced constants and non-executed program steps have been made.

The program, and the material on the following pages, is a modification of that for Colossus 3, and therefore some of the Colossus 3 nomenclature (particularly references to "LM") has been retained in both the software and the documentation for continuity with previous programs.

Considerable assistance in updating the Colossus 3 document to the Skylark configuration was provided by Sandra L. Woodcock. Her participation in the preparation of the Skylark programmed equations document has enabled it to be considerably more timely and accurate than would otherwise have been possible.





## Summary of Individual Writeups

The information on the programmed equations has been divided into a number of separate entities called "writeups". Each writeup is intended to be complete within itself except for definition of certain frequently used quantities (described in "Major Variables") and perhaps for reference to routines in other writeups (whose location may be found through use of the "Index of Routines"). Familiarity with the information in "Notation and Terminology" is necessary in order to understand the material presented in the individual writeups, since a somewhat specialized type of notation is necessary in order to describe adequately the computations within the guidance program. Individual writeups are assigned four-letter codes, with page numbers restarted at the beginning of each writeup, to facilitate updating. The writeups are arranged in the alphabetical order of these four-letter codes, and are also arranged in that sequence below.

### Attitude Maneuvers (ATTM)

This writeup contains the computations performed to determine the axis about which spacecraft rotation should take place and the magnitude of the rotation required to go from the present vehicle attitude to a final specified attitude. It also shows the attitude computations which are performed to provide a general-purpose vehicle attitude pointing capability.

### Boost Computations (BOOS)

This writeup contains the computations performed when the liftoff signal is received, including the control for computing display parameters during the boost phase of flight, the calculations to drive the error needle displays, and the computations associated with the Saturn mode of operation of the Digital Autopilot (DAP). Also included are the equations used for backup Saturn steering after liftoff.

### Burn Control (BURN)

This writeup contains the control computations associated with most of the pre-thrust and thrusting programs. This includes the logic associated with the generation of the various displays and some of the computations associated with the programs themselves (other computations are in the Display Computations, Rendezvous Computations, and Steering Computations writeups). The control for sequencing through the automatic rendezvous sequence is in the Minimum Key Rendezvous writeup.

### Conic Routines (CONC)

This writeup contains the various conic subroutines that are grouped together in the program and in official equation information, including the Lambert and Kepler routines.

### Coordinate Transformations (COOR)

This writeup contains the various coordinate transformation subroutines in the program, entered by the using routines as necessary. Also included are the routines for determining solar position, and the transforming between latitude/longitude/altitude and a position vector.

### Data Input/Output (DATA)

This writeup contains the computations performed in response to a keyboard (DSKY) or uplink input, as well as those routines that permit internal use of the display system. The computations associated with most of the nouns and special-purpose verbs, as well as the display interface routines, are subjects of separate writeups.

### Display Interface Routines (DINT)

This writeup contains the various display interface routines used by internal callers of the display system in order to achieve proper sequencing of the displays according to their priorities, and to achieve various special-purpose displays with a minimum of complication to the "mission program" making use of the display system.

### Display Computations (DISP)

This writeup contains the computation of those variables whose major (or sole) purpose is display for crew review or action (time of flight, special rendezvous displays, entry preparation displays, etc.). The VHF range rate filter equations are in the Measurement Incorporation writeup.

### Digital Autopilot Docked Jet Selection (DPDS)

This writeup contains the logic associated with the selection of jets to be fired for the docked (to OWS) DAP. The inputs to this logic, as well as the logic for the RCS DAP (i.e. undocked DAP), are in the computations of the Digital Autopilot RCS Routines writeup.



#### Digital Autopilot Entry Routines (DPEN)

This writeup contains the computations associated with the performance of the digital autopilot (DAP) during the entry phase of the flight, including the selection of the appropriate RCS jets for the CM and the timing of their firing duration.

#### Digital Autopilot Interface Routines (DPIR)

This writeup includes those computations performed in support of several portions of the DAP (such as derivation of mass properties information), as well as those influencing both the DAP and the non-DAP portions of the program (such as engine gimbal drive test). Also included are the computations associated with the crew use of DAP parameters (verbs 44 and 45 for docked DAP and verbs 46 and 48 for (undocked) RCS DAP).

#### Digital Autopilot RCS Routines (DPRC)

This writeup contains those computations performed for the service module RCS (Reaction Control System) DAP in both the docked (to OWS) and undocked modes, except for docked DAP jet selection which is the subject of the Digital Autopilot Docked Jet Selection writeup. The undocked DAP jet selection logic (with the effect of jet failure information) and the processing of manual controller information is also shown.

#### Digital Autopilot TVC Routines (DPTV)

This writeup contains those computations associated with the TVC (Thrust Vector Control) DAP, used with the SPS engine. This includes the computation of the trim estimation and the TVC DAP roll-control computations (which, of course, use the RCS jets).

#### Entry Computations (ENRY)

This writeup contains the computations performed for entry, once the maneuver to the entry attitude has been completed, as well as the entry display computations. The entry-DAP support computations performed at a two-second rate are also shown.

#### Entry Preparation (ENTP)

This writeup contains the computations performed when preparation for entry is started, including generation of displays for entry monitoring purposes and request for separation.

#### General Program Control (GENP)

This writeup contains those computations which have an influence upon the performance of the complete program. Included are the equations for starting accelerometer sampling, powered flight navigation ("Average-G"), the TIME4 interrupt control logic, fresh start and

restart computations, processing of a verb 37 (program change) input to start the new program, pre-standby and post-standby computations, and the alarm and software restart routines.

#### IMU Computations (IMUC)

This writeup contains the computations directly associated with the IMU (Inertial Measurement Unit), including accelerometer sampling, powered flight and free-flight accelerometer and gyro compensation, the IMU monitoring functions performed by the TIME4 interrupt logic (checks for status, alarms, and gimbal lock), and the IMU mode change and gyro torquing routines.

#### Inflight Alignment (INFA)

This writeup contains the computations required to perform the various IMU alignment functions during flight, as well as the automatic star selection routine. Also included are programs for use with the OWS hardware (P50 and P55) to determine orientation of the ATM coordinate system with respect to that of the CSM and to compute gimbal angles for the ATM star tracker. Alignment prior to launch is described in the Prelaunch Alignment writeup.

#### Mathematical Functions (MATH)

This writeup contains the various trigonometric, logarithm, and square root functions that are included in the program.

#### Measurement Incorporation (MEAS)

This writeup contains the computations which are performed to update the state vector based on optics or VHF range measurements via the rendezvous tracking data processing routine (R22). Also included are most of the computations associated with the VHF range rate filter done for display purposes (R27). Other navigation logic is in Orbital and Rendezvous Navigation.

#### Minimum Key Rendezvous (MINK)

This writeup contains those routines associated exclusively with the minimum key rendezvous (automatic rendezvous) logic, including the program sequencing logic, the logic for W matrix automatic reinitialization, and the special alignment program computations.

#### Noun Definitions (NNDF)

This writeup contains the computations and constants directly associated with the decoding of the various nouns used with the display system. Also included in this writeup is the information reflected in the noun tables incorporated in the program, and a list of the program internal references to individual nouns.

### Optics Computations (OPTC)

This writeup contains the computations associated with the optics system, including those for monitoring optics status, changing the position of the optics, performing optics marks and processing them, and computing optics pointing information.

### Orbital Integration (ORBI)

This writeup contains the orbital integration and related control (interface and driver) functions that are included in the program for precision (Encke) integration, and the interface and driver logic for conic integration (the Kepler routine is in Conic Routines). Also included are the routines to transfer between powered flight and free flight (and between free flight and powered flight, i.e. Average-G) navigation.

### Orbital and Rendezvous Navigation (ORVN)

This writeup contains the computations associated with orbital navigation and rendezvous navigation (except R22 which is in Measurement Incorporation, as are some W matrix routines). Included is the control logic associated with the generation of displays and initialization of the W matrix (except for the Minimum Key Rendezvous material on automatic W-matrix initialization).

### Prelaunch Alignment (PREL)

This writeup contains the computations performed prior to launch to erect the platform and, by gyrocompassing, align it to the proper orientation for launch. Also included is the logic for processing a change to the required launch azimuth (initiated by a verb 78). The computations shown are provided independently of those for IMU calibration (in Testing Routines), even though some duplication of the resulting information is necessary, for convenience in description.

### Rendezvous Computations (REND)

This writeup contains the calculations associated with the performance of the pre-thrust and thrusting guidance of the rendezvous programs (the Lambert routine itself, however, is in Conic Routines). The program control for these pre-thrust and thrusting programs is in the Burn Control writeup.

### Steering Computations (STER)

This writeup contains the guidance computations associated with the SPS burn program (P40), some of which are also used with the RCS burn program (P41). This includes the cross-product steering routine, computation of preferred IMU alignment and burn attitude, the thrust monitoring computations (logic based on monitor outputs is in Burn Control), etc. The velocity-required logic for Lambert guidance is in Rendezvous Computations.



### Telemetry (TELE)

This writeup contains the computations performed when a telemetry interrupt is received, and the logic associated with the generation of the "erasable memory dump" downlist (verb 74). Also included is a brief summary of the information which is telemetered.

### Testing Routines (TEST)

This writeup contains the computations performed for checking the computer erasable and fixed memory (including the memory sum check): these are the only computer self-check computations in the program. Also included are the computations associated with the prelaunch calibration of the IMU that are provided in fixed memory (limited mainly to drift testing). These latter computations are shown independently of those for prelaunch alignment, even though some duplication of the resulting information is necessary, for convenience in description. In addition, information from an earlier program (Sundisk) was used to supply an indication of expected values of erasable memory constants and program steps to be used with the drift testing.

### Uplink Processing (UPLK)

This writeup contains the computations performed in response to receipt of the special-purpose uplink/DSKY verbs (70-73), and the logic associated with the incorporation of a state vector update.

### Verb Definitions (VBDF)

This writeup contains the computations associated with the decoding of most of the special-purpose verbs (except those in DPIR and UPLK). Only the initial computations performed after the verb is decoded are shown in this writeup (such as the setting of a control cell and the establishing of the job), with subsequent calculations given in another writeup (such as Display Computations).

## Program Computation Flow

Because of the nature of the Apollo Guidance Computer hardware (and software) design, there is no unique "flow" of computations that takes place. Instead, various calculations can proceed virtually independently of one another at different cycle rates. For some purposes, however, that do not require detailed simulation or understanding of the computer calculations, the simplification of a straight-line flow can be made. The following paragraphs outline briefly the sequence of computations which takes place with such a simplification, and are intended to orient the user to some of the contents of this document. A review of the segment on "Summary of Individual Writeups" will also prove useful.

### Prelaunch

When power is first applied to the computer, one of the early functions which is performed is the setting of the computer's initial conditions in erasable memory. This is accomplished by means of an input of V36E, which causes the computations starting at "SLAP1", page GENP-8, to be entered. Following this, the various erasable memory presets required for the flight (such as constants stored in erasable memory for flexibility) can be performed either by DSKY or Uplink means. An uplink interrupt causes "UPRUPT", page DATA-1, to be entered; a main panel DSKY key depression causes "KEYRUPT1", on the same page, to be entered; and a navigation panel DSKY key depression causes "MARKRUPT", page OPTC-9, to be entered (where checks are first made for an optics mark or mark reject). All three data inputs eventually cause "CHARIN", page DATA-2, to be established to process the input information.

Following this, the IMU can be made ready for flight, making use of computations starting at "IMUMON", page IMUC-3. Next, an input of V37 E 01E will start program 01 (P01), on page PREL-1 ("GTSCPSS"). This causes the initial conditions for the prelaunch erection and gyro compassing routines to be set. In addition, the IMU CDU's are zeroed, coarse aligned to the desired IMU angles for the flight, and then switched to fine align. At the end of this sequence, some more initial conditions are set and the program number on the DSKY changed to 02, whereupon "SLEEPIE" (page PREL-3) is entered.

"SLEEPIE" is the standard end point for the prelaunch gyro compassing routine (vertical erection is performed for 640 seconds at the start of P02, and for 320 seconds after an azimuth change input). It controls a half-second waitlist call of "ALLOOP" to read and reset accelerometers and establish "ALFLT" (page PREL-4). Every five seconds, "EARTHRE\*" is entered to correct for earth rotation effects and initiate pulse torquing of the gyros (page PREL-6).

While in PO2, changes to the azimuth towards which the stable member is being aligned (i.e. the "launch azimuth") may be achieved by V 78 E, which causes "AZMTHCG1", page PREL-6, to be established. A new launch azimuth may then be loaded. After completion of the routine (either by a PRO or V34E), a flag is set that is checked every 5 seconds in "ALFLT". If any change in azimuth is detected, the Z-axis gyro is pulse torqued as specified by the angle change, and then the gyro compassing program is essentially reinitialized (by transfer to "ESTIMS", page PREL-2), although the vertical erection interval will be only 320 seconds instead of the 640 seconds at the start of PO2.

Also while PO2 is running, PO3 can be initiated (the gyro-compassing computations continue) in order to perform an optical verification of azimuth. PO3 can be entered by V 65 E, which causes "GCOMPVER", page PREL-7, to be established. After completion of the optics marks and displays, PO2 again becomes displayed (by entrance to "GCOMP5", page PREL-9).

### Boost

At the beginning (in "ALFLT") and end (in "SLEEP1E") of each half-second gyro-compassing cycle in PO2 or PO3, a check is made for liftoff by performing "CHKCOMED", page PREL-3. If the liftoff input channel bit indicates liftoff, or if a V 75 E (backup liftoff, performing "LFTFLGON", page VBDF-12) has been received, then the prelaunch cycling is halted and "P11", page BOOS-1, is entered.

"P11" updates ephemeris time to have an origin of liftoff time (and zeros the computer clock), changes the program number display to 11, gets rid of the backup liftoff verb (or other DSKY input) if it is on the DSKY, initializes position and velocity based on pad location data, and computes an initial value for [REFSMMAT], the reference-to-stable-member matrix. It also performs some initialization functions for the computation of position and velocity every two seconds, which is a process starting at "READACCS", page GENP-2. The value of liftoff time may be "updated" by using verb 70 (in P27, as described on page UPLK-9), if desired, after P11 has been terminated and PO0 entered.

Every two seconds during the boost phase, and whenever else Average-G computations are running, "READACCS" is entered. This routine causes accelerometer data to be sampled (and accelerometers reset) via "PIPASR" (page IMUC-1). This task also causes "SERVICER" to be established (page GENP-4), which checks for excessive (saturated) accelerometer output, compensates accelerometer data for scale factor and bias errors via "1/PIPA" (page IMUC-1: gyro compensation during powered flight is also handled there), updates position and velocity using "CALCRVG" (page GENP-4), and then transfers to "VHHDOT", page BOOS-2. "VHHDOT" computes quantities for display purposes, and



causes them to be displayed by noun 62 (see page NNDF-27) unless some other display has been specified by the crew. This two-second cycle continues until manual specification of another program is provided (by V37 means) by the crew.

Shortly after "P11" was first entered, the present CDU angles are sampled and used as the "desired" CDU angles for driving a display (in roll, pitch, yaw body coordinates) on the FDAI attitude error needles. Starting a desired length of time after liftoff deduced (the time interval is specified by an erasable memory constant), "ATERJOB", page BOOS-3, begins using nominal roll rate information and a stored pitch polynomial to compute the desired CDU angles. These computations are halted a desired length of time (again specified by an erasable memory constant) after they were started. They are performed with a low "priority", meaning that they will be done only if no other computation (such as Average-G) must be done. In addition, a delay of  $\frac{1}{4}$  second is imposed between the end of one pass through these equations and the start of the next pass. Backup automatic steering of the Saturn using the output of the polynomial (with errors considered zero at initiation) can be achieved by setting appropriate spacecraft switches.

Manual inputs for Saturn control can be accomplished by first keying in V 48 E and loading DAPDATR1 with 3xxxx (page DPIR-2) to specify the Saturn DAP configuration. By next keying in V 46 E, "STABLISH", page DPIR-1, is entered, which in turn causes "SATSTKON", page BOOS-5, to be entered. This in turn causes "REDOSAT" to be entered, which finally causes "SATSTICK" (page BOOS-6) to be entered every 0.1 second to generate outputs based on RHC (rotational hand controller) settings. When "SATSTKON" is entered, flagword bits are set which cause "ATERJOB" to halt.

### Coast

When it is decided that Average-G computations should be terminated at the end of Saturn thrusting, this can be accomplished by V37 E OOE, which will cause "V37", page GENP-15, to be entered. Since Average-G will be running, "ISITPOO", page GENP-18, will set a flagword bit (bit 1 of FLAGWRD1) to 0, causing "READACCS" to set the exit from the Average-G equations to transfer to "AVGEND" (page GENP-3). There, the state vector from the last Average-G cycle is loaded into the "permanent" state vector for free flight of the CSM (in "AVETOMID", page ORBI-4), and then, after some other settings, "CANV37", page GENP-19, is entered, in order to continue processing of the program whose number was specified by V37 input.

The "CANV37" computations cause a master initialization routine, "INITSUB" (page GENP-17), to be performed to set several flagword bits and do other functions. Then, since program 00 was specified ("POO"), a restart group is set so that "STATINT1" will be established. Restart groups are a software design feature included mainly for the purpose of providing a capability to resume computations in the event of a transient difficulty such as a power glitch, but they also can be used to achieve proper program performance control under normal conditions. Computations similar to those performed when the computer



was initialized at turn-on are then performed (via entrance to "GOPROG2", page GENP-12), and these include establishing "STATINT1".

"STATINT1", page ORBI-1, is the job intended to keep the CSM state vector from becoming too "old", while minimizing the buildup of errors due to the computation. It is entered every 10 minutes (more precisely, it is entered 10 minutes after its previous pass has been completed). This looping can be halted (or looping for an excessive integration time within the orbital integration package) and POO selected by an input of V96E (which sets a flagword bit checked at the start of "STATINT1" and also in "TESTLOOP", page ORBI-10). If "STATINT1" detects that the bit is set, it is reset and no additional ten-minute waitlist call is made (hence integration remains suspended until do e.g. V37 E OOE again).

If the flagword bit (QUITFLAG, bit 5 of FLAGWRD9) indicating a V96E is not set, then further checks are made before deciding to perform an integration. These are done by setting some flagword bits in "STATINT1" that are analyzed in "TESTLOOP": these preclude starting an integration of the CSM state vector unless it is at least four "time steps" (a variable time depending on distance of vehicle from primary body) delayed from the present time. If integration is started, it will continue until the state vector is within one time step of the "present" time (measured at the start of "STATINT1"), but no integration for a fraction of a time step is done. After the CSM state vector has been integrated (if necessary), then the OWS state vector is integrated to the same time as the CSM state vector, so as to keep the two state vectors "together". While the integration of the two state vectors is going on, bit 1 of FLAGWRD2 is set to inhibit processing of a V37 input that would interrupt the integration after the CSM state vector had been integrated, thus leaving the two state vectors out of synchronism with one another. Unless a V96E is performed, the ten-minute checks continue indefinitely, until some other program is specified.

### Uplink

Four verbs (70 - 73) are provided for loading the computer erasable memory with parameters from the ground via the uplink (the same verbs could be used for input via the DSKY, of course). These verbs, which are accepted only if the present program is POO, PO2, or P20 in a non-rendezvous (i.e. tracking) mode, cause P27 to be displayed via computations starting on page UPLK-1. The function of each of the verbs is described starting on page UPLK-9.

Processing of an OWS or CSM state vector update is handled by verb 71 (see page UPLK-10), with the first word in the sequence (after those associated with P27 performance control) serving as an identification of which state vector is involved. The



computations performed to incorporate the state vector are in "INTWAKEU", page UPLK-6. Since the erasable memory general purpose update verbs (71 and 72) include specification of absolute erasable memory cells, considerable flexibility in the uplink information is possible. Aside from some constraints on erasable memory bank changes (verb 71 only), no software lockouts are included, so that internal computer program control cells can be modified (such as flagwords). To change particular flagword bits (for addresses above  $30_8$ ), however, it would be more satisfactory to make use of noun 07 (see page NNDF-22). Channels (of address  $30_8$  and below) can also use the noun, or else noun 10.

#### External Delta V Burn (SPS)

Preparations for an External Delta V burn (for which a velocity increment and ignition time are specified by inputs to the computer) can begin by the performance of an uplink load of the target parameters (and, if desired, an update of the state vector). Performance of the External Delta V pre-thrust program (P30) is then started by input of V37 E 30 E, causing "P30", page BURN-1, to be entered. There, after setting two program control flagword bits (associated with P20), a display of ignition time is produced. This display is generated in the program by loading the accumulator with the proper bit pattern for verb 06 and noun 33 (bits 14-8 contain the verb and bits 7-1 the noun), and then transferring to display interface routine "VNFLASH". This routine (page DINT-6) sets a flagbit indicating that it was entered (for use in processing the response), retains information on the verb and noun desired, and eventually progresses to "MAKEPLAY" (page DINT-7), where the actual display generation functions are initiated. Return is not made to the computations on BURN-1 until a suitable response to the display is provided.

The "MAKEPLAY" computations cause internal communication cells to be set for generation of the information required by the verb-noun pattern: verb 06 means display in decimal and noun 33 (page NNDF-23) references  $T_{ig}$ , the ignition time. After the display is produced ("HMSOUT", page NNDF-3, is entered due to noun table information, causing "DSPDECWD", page DATA-25, to load DSPTAB cells via "DSPIN", page DATA-27, for subsequent loading, once every 40 ms, in "DSPOUTSB", page GENP-7, into hardware output channel 10), the job is suspended ("put to sleep") in "ENDIDLE", page DINT-16, awaiting a crew response.

In general, responses to flashing displays are implemented in the program as a return to one of three addresses depending on the nature of the response. These are referred to as "terminate" (a crew input of V34E), "proceed" (a crew input of V33E or depression of the PRO DSKY button), or "otherwise". The "otherwise" return can be caused by a variety of DSKY inputs, although in general only one of them would be appropriate to the display presented: these include completion of a data entry, input of a recycle verb (V32E), or inputs of verbs 50, 51, 53, 97, or 99. It should be recognized that the "otherwise" term is a document convention (in this document), and does not signify, for example, some sort of continuous program looping until a proceed/terminate response is given.

The various DSKY responses all exit to "RECALTST" (page DATA-33), where the job previously put to sleep is awakened. It starts executing at "IDLERET1", page DINT-16, where a check is made that if a data load verb was received, then the accompanying noun was what was requested originally: if not, the original display is regenerated. Following this, logic in "ENDRET", for the "VNFLASH" entrance (which set the VNFLAG bit), determines on DINT-18 where the return should go. A terminate transfers directly to "GOTOPOOH" (without returning to the caller), a proceed transfers to calling address +1, and an "otherwise" to calling address -1 (to pick up verb/noun pattern again). If the VNFLAG is not set, as true for e.g. "GOFLASH" (such as from "P77", BURN-31), then returns for terminate, proceed, and otherwise go to calling address +1, +2, and +3 respectively. For the "VNFLASH", since the terminate response is handled on DINT-18 rather than by the calling routine, the action for terminate is indicated in parentheses. The "GOTOPOOH" routine, page GENP-15, causes a request for a new program to be generated ("V 37 E" is effectively produced by the software).

If the response was a "proceed", then the program goes on to the next display (noun 81), as indicated on page BURN-1 by "proceed". The noun 81 display process closely parallels that for noun 33: noun 81 (page NNDF-29) causes a display of the velocity increment in local vertical coordinates. It should be noted that, by convention, this information is in the DELVLVC erasable memory cells: no coordinate transformations are done by the DSKY output routines. The scale of the display is XXXX.X fps, as implemented by constant  $K_{\text{sfot}}$ , page NNDF-17, with a suitable scaling shift. The terminate and "otherwise" responses cause a similar effect as for the first display, while "proceed" again causes computations to be started up on page BURN-1.

Following a proceed response to the second display, the XDELVFLG bit is set 1, indicating that an External Delta V burn has been targeted. After this setting has been accomplished, a P20 control bit (used to control updates) is reset and then the CSM state vector integrated to the input (noun 33) ignition time via integration driver routine "CSMPREC", page ORBI-2, entered with desired time in  $T_{\text{decl}}$  and exiting with position vector in  $R_{\text{att}}$  and velocity vector in  $V_{\text{att}}$ . This information is used to retain the state vector at ignition for use by "S40.1" (page STER-1), and also to compute the apocenter and pericenter altitudes expected after the burn for display by noun 42 (page NNDF-24). Following the response to this display, a display of N45 (number of marks, time until ignition, and middle gimbal angle) is generated: the middle gimbal angle is set to  $-0.02^\circ$  (by convention) if the orientation of the IMU is not known, and is computed in "GET+MGA" (page DISP-1) if the IMU orientation is known.

The N45 display uses a verb 16 instead of a verb 06: this means a monitoring function, giving an update (with new data if available) due to entrance to "MONDO" (page DATA-23) once a second from logic in "MONREQ" (same page). The middle register of N45 (R2) has data that is updated once a second also, namely  $T_{\text{logo}}$ , computed as  $(T_{\text{now}} - T_{\text{ig}})$  by "CLOKTASK" (page BURN-24). After a PRO response to the N45 display, the "GOTOPOOH" routine, page GENP-15, is entered to request a new program as described previously for V34E responses to the display. Although there is no lockout requiring P30 to be performed prior to the burn program, subsequent computations make use of quantities computed in P30 (before the N42 display), so P30 should not be bypassed.



Assuming the IMU is on and aligned satisfactorily, after P30 the DAP parameters can be loaded via RO3 (Routine O3), which is initiated by V48E (entering "DAPDISP", page DPIR-2). After loading the components of a particular noun, the calculations to incorporate this data in the parameters actually employed in the DAP are made. It should be noted that changes in the input masses of CSM (or LM, if LM is indicated as being attached) will not have an effect on the vehicle mass properties (moments of inertia and total vehicle weight used in estimating acceleration in the P40/P41 programs) unless a specification of the RCS DAP (LM-off or LM-on) is made in RO3. RO3 should be completed before P40 is selected, since some of the parameters loaded (trim estimates as well as vehicle weight) in RO3 are used in the P40 computations. Software has "LM" from Apollo 15 "baseline".

Assuming the SPS burn program (P40) is desired, input of V37 E 40 E will cause the computations at "P40CSM", page BURN-20, to be entered. There, after doing some initialization and checking that the IMU is aligned (via "RO2BOTH", page IMUC-1), routine "S40.1", page STER-1, is entered to rotate the velocity change input data (computed by P30) by half the estimated central angle of travel during the maneuver, and to compute for optional display the velocity change information in local vertical coordinates (via "GET.LVC", page DISP-1).

The "S40.1" routine exits to "S40.2,3" (page STER-2), which loads cells for the initial attitude required for the burn, as well as computes and loads a "preferred IMU orientation" (which makes gimbal angles read zero at ignition, and hence includes the effect of the pitch and yaw trims entered in Routine O3). This routine exits to "P40SXTY", page BURN-20, where a bit is set indicating that a preferred IMU orientation is available (for use in P52/P54 to generate the initial option code display). Then, after specifying that the maneuver should be of a "VECPPOINT" (as opposed to 3-axis) type, "R60CSM" (page ATTM-2) is entered to perform the maneuver to the ignition attitude. P52 or P54 could be selected at this point to realign the IMU if desired. If one of these programs is entered (by V37 means), of course, it would be necessary to reselect P40 again.

Completion of the attitude maneuver routine causes a checklist code 0204<sub>g</sub> display to be produced, requesting a decision as to whether an SPS engine gimbal drive test should be done (PRO) or bypassed (ENTR). Response to this flash causes "S40.6" (page DPIR-5) to be entered to disable optics driving (since same computer counter cells used to control position of SPS engine bell) and to set the SPS engine to its trim value in pitch and yaw (as was specified by RO3) after doing, if specified, the gimbal trim test. After completion of this routine, an interface cell is set causing "CLOKTASK" (which was started at the start of P40 for optional display) to establish "CLOCKJOB" which in turn displays non-flashing VO6N40 that is updated once a second (by the "CLOCKJOB" computations) with time from ignition. Then the "MIDTOAVL" computations (page ORBI-5) are entered to integrate the state vector to T<sub>ig</sub> - 29.96 seconds, the time when Average-G cycling is to be started. If the integration is not completed at least 12.5 seconds before the Average-G cycling is to be started, then an alarm is generated (pattern 1703<sub>g</sub>) and ignition time is automatically slipped to 29.96 seconds from the time to which the state vector has been integrated (no crew response is required for this to occur).



Five seconds before Average-G is to be started, i.e. at T-34.96 seconds, the DSKY is blanked (except for the program number) due to "TIGBLNK" (page BURN-22) establishing "P4OBLNKR" (page BURN-22). Five to six seconds later, depending on the phasing of "CLOKTASK" with respect to the "countdown" to ignition, the VO6N40 display is restored again via "TIGAVEG" (page BURN-22), which sets NVWORD1 to the desired verb-noun pattern (causing "CLOKTASK" to establish "CLOCKJOB").

Five seconds before nominal ignition, NVWORD1 is set to - 400<sub>8</sub> in "TIG-5". The next time "CLOCKJOB" is entered (i.e. within a second), this negative non-zero setting will cause "CLOCPLAY" to be entered. This routine sets several display interface routine control bits, including the "BURNREQ" bit. When "NV5ODSP" (page DINT-13) detects these bits, it causes a verb 99 to be placed on the DSKY to write over the original verb 06 used to generate a decimal display. This verb 99 permits an ENTR response to go to the "otherwise" return (via "GOLOADLV", page VBDF-7, per page VBDF-22): this response means that no SPS burn is desired, but post-burn display information should be supplied. A PRO response, on the other hand, means that engine ignition is authorized: this causes "V99P" (page BURN-27) to be entered in order to ignite the engine either promptly (if nominal ignition time has already arrived) or, by setting a flagword bit, at nominal ignition time if it has not yet arrived.

Five seconds after "TIG-5" was entered, namely at the original ignition time (or the time updated as a result of slipping ignition to complete the integration), "TIG-0" is entered (page BURN-23). This routine either enters "IGNITION" (same page) at once, if crew authorization has been received, or merely sets a bit that causes "V99P" to enter "IGNITION" if authorization received after the nominal ignition time. The "IGNITION" routine samples the outer gimbal angle for roll hold in the roll TVC DAP, turns on the SPS engine, if a <1 sec. "short" burn was deduced (from a job established by "TIG-5") sets a waitlist task to turn off the engine, disables the RCS DAP, and then delays 0.4 seconds. At the end of this interval, "TVCDAPON" is caused to be entered in 0.01 seconds to get the TVC DAP running (as discussed below); 1.6 seconds later, namely at 2 seconds from ignition, the ullage jets are turned off (they must be initiated manually, and remain at whatever setting they were left when the RCS DAP was disabled, since the "disabling" does not affect the RCS DAP TIME6 logic (see below)) and, if the burn is not "short" (i.e. predicted to be more than 6 seconds), bit 11(STEERSW) of FLAGWRD2 is set 1 to enable steering and computation of time-to-go. This concludes the engine ignition sequence.

After P40 returned from the R60 attitude maneuver, the exit from the Average-G computations (AVEGEXIT) was set to "S40.8" (page STER-5). When "TIGAVEG" is entered 29.96 seconds before nominal ignition time, transfer is made to "PREREAD" (page GENP-1): this causes free-flight gyro bias compensation to be halted, clears the accelerometers, loads the Average-G state vector cells with information computed previously ("MIDTOAV1" results), and calls "READACCS" in 2 seconds (i.e. at nominal ignition - 27.96 seconds). "READACCS" performs in the same manner as described previously for the Boost phase, but instead of having "SERVICER" transfer to "VHHDOT", it transfers to "S40.8" (because of the AVEGEXIT setting).

"S40.8" (page STER-5) comprises the "guidance equations" for the flight. The first function that is done is the performance of "UPDATEV<sub>g</sub>" (page STER-4): for an External Delta V burn, that routine merely updates  $V_g$  based on the accelerometer output (DELVREF). "S40.8" next checks for the magnitude of the accelerometer output being below a pad-load limit (as it would be expected to be until the engine is turned on). If it is, and if the STEERSW bit is zero, the computations cease (STEERSW is reset in e.g. "INITSUB") until the next two-second cycle. The program remains in this mode of updating  $V_g$  after Average-G until the pad-load accelerometer output limit is exceeded. When this happens, the computer's internal estimate of the weight of the CSM vehicle (CSMMASS) is updated by two seconds' worth of nominal SPS mass flow rate, and, if STEERSW is still zero, the computations again cease. If a short burn is done (predicted duration in "S40.13", page STER-7, below 6 seconds), then STEERSW will remain zero; otherwise, it is set 1 in "IGNITION" 2 seconds after ignition.

After the STEERSW bit gets set, and assuming the accelerometer output is sufficient, CSMMASS gets updated, a TVC DAP quantity used for trim estimation is reset, and, after a check for thrust in the proper direction, a value of  $T_{go}$  is computed. If this value (which includes an erasable memory constant for thrust decay effects) is at least 4 seconds, then a new guidance steering command, OMEGAC, is generated for use by the TVC DAP. When it becomes below 4 seconds, "S40.81", page BURN-27, is entered. There, the guidance steering command is set 0 and the engine cutoff computations are called at the proper time.

If "S40.8" finds a single point below the accelerometer output threshold, with the STEERSW bit set, then thrust fail procedures are indicated (since the failure could be in the inertial sensor rather than the engine, however, no engine cutoff command is generated at this point). After a check for a bit set only for response to a fail condition, the guidance steering command is set 0, a DAP cell used for trim updates is zeroed, NVWORD1 is set -0, and STEERSW is set 0. The STEERSW setting reduces "S40.8" to the  $V_g$  (and possibly CSMMASS) update mode, while the NVWORD1 = -0 setting causes "CLOCKJOB" (page BURN-25) to generate a similar display to that for engine ignition authorization, except that the verb, instead of being 99, is now 97 (and, of course, different program branches are taken). An ENTR to the V97N40 display causes the  $T_{togo}$  display to be set to +59 59 (for reasonable times from launch), the engine to be shut off, the TVC DAP to be shut down, the RCS DAP to be started, a new "short" burn estimate via "S40.13" to be made, and finally, about  $2\frac{1}{2}$  seconds after the ENTR, a V99N40 display to appear (all this is done in "V97E", page BURN-26). The effect of the ENTR, therefore, could be considered analogous to returning to the  $T_{ig}$  -5 point, although of course the RCS DAP attitude hold reference probably would be different from that left by R60. If a PRO response is provided (indicating that the engine is on), then "V97P" is entered: this resets STEERSW back to 1 and sets bit 6 of FLAGWRD1 to 1 for 2 seconds to inhibit the "S40.8" failure action. A V34E response to the V97 causes engine shutdown, followed by startup of the RCS DAP and entrance to "GOTOPOOH". It should be noted that a crew response to the V97 flash is necessary before any action is taken by the software to shut down the engine or disable the TVC DAP (the software does, however, zero guidance steering commands and disable trim updates by the DAP).



The normal engine cutoff routine is "ENGINEOFF" (page BURN-28), which shuts off the engine, waits  $2\frac{1}{2}$  seconds, starts up the RCS DAP (after a time delay to avoid interference with TVC DAP shutdown), and finally establishes "POSTBURN" (page BURN-28) to provide a V16N40 display. After a PRO response to this display, AVEGEXIT is loaded with "CALCN85" (page DISP-2), which likewise enters "UPDATEVG" but which then transforms the velocity-to-be-gained information into RCS control coordinates (rotated  $7\frac{1}{4}$  degrees from body axes) via "S41.1" (page STER-8). This information is used to drive a V16N85 display initiated by "TIGNOW" (page BURN-29). After a response to this display, "GOTOPOOH" is entered (which, via "INITSUBA", restores the RCS DAP deadband to the value last specified by the crew), thus ending the External Delta V sequence.

#### External Delta V Burn (RCS)

The initial preparations for an External Delta V burn using the RCS instead of SPS propulsion system can follow the same procedures as described for the SPS burn, since selection of a propulsion system is not made until the burn program is chosen. Following P30, input of V37 E 41 E will cause the computations at "P41CSM", page BURN-29, to be entered. There, after doing some initialization, "P4OS/F", page BURN-20, is entered in order to take advantage of the similarity between the P40 and P41 burn initialization computations. Following return from the maneuver routine, the P41 sequence starts a display of the velocity-to-be-gained in RCS control coordinates, using V06N85. This display is initialized in "P4OSXTY", and is updated once a second (by "CLOKTASK" establishing "DYNDISP", page BURN-25). Following this, the "MIDTOAV1" routine is entered to request the state vector at 29.96 seconds before ignition, with return options as for the SPS case, and the DSKY is blanked by "TIGBLNK" 5 seconds before Average-G turn-on, again as for the SPS case.

When "TIGAVEG" (page BURN-22) is entered to start Average-G at 29.96 seconds before the nominal ignition time, a check is made to conclude that P41 (as opposed to SPS, P40) is running. If it is, the DSKY display is made V16N85, and "TTG/O" is called at the nominal ignition time. The updating of the N85 information by "DYNDISP" is no longer done, since the updates are now based on the exit of Average-G to "CALCN85" (set in "P4OSXTY" after return from R60) which, as discussed for this computation in the final P40 display, does its own coordinate transformation (as well as entering "UPDATEVG" to update velocity-to-be-gained based on accelerometer outputs). The only functions of "TTG/O" (page BURN-29) are to establish "TIGNOW" (page BURN-29), causing the V16N85 display to flash, and to disable the cycling of "CLOKTASK" by setting a flagword bit to 0. Termination of the display (by a terminate or PRO response) causes the "GOTOPOOH" routine to be entered (as in the P40 case).

Monitoring of thrust applied, with Average-G running, can also be done in P47, which starts at "P47CSM", page BURN-29. There, after checking for IMU alignment ("R02BOTH"), routine "MIDTOAV2", page ORBI-6, is entered. This routine performs computations similar to "MIDTOAV1" (described for the SPS case), except that it uses the "ignition slipped" logic, requesting for each orbital integration time step an integration to 12.5 seconds after the present time, and halting (in general) when the given time step successfully completes the integration. This effectively permits Average-G to be turned on "as soon as practical" after entrance to P47. "TIGNON",

page BURN-30, is called to start Average-G, to set the Average-G exit to "CALCN83", and to generate in "P47BODY" (page BURN-30) a V16N83 display of velocity change. "CALCN83", page DISP-1, updates the velocity change (converting it into RCS control coordinates for the N83 display), and also computes the quantities displayed by N62 (for optional crew observation of them by a manual DSKY input): these quantities are the same as those supplied by the P11 Boost program.

### Rendezvous Targeting

Instead of having a fixed externally provided velocity change to be achieved by a burn, as is the case for P30, a capability is also provided to have the burn program parameters determined internally in order to achieve rendezvous. The various rendezvous programs can each be sequenced manually, or an automatic sequence, reducing the number of crew keystrokes required, can be performed.

If a manual input of V37 E 3x E is made ( $x = 1 - 8$ ) from a "base configuration" of no rendezvous programs running (e.g. from P00), and if the REFSMFLG is set (IMU orientation known), then "V37" (page GENP-15) exits to "REND3OS", page MINK-1. There, an initiation of the P20 program is performed by setting the AUTOSEQ bit = 1, entering "AUTOSET" (page MINK-2), transferring to "AUTO37" to complete the processing normally done for a V37 (page GENP-16), and finally starting at "PROG20", page ORVN-1. There, since the AUTOSEQ bit is 1, displays are suppressed and "NDUTINPT", page ORVN-3, transfers to "AUTOCHK", page MINK-1. There, since AUTOSEQ bit is 1, transfer is made to the address in AUTPOINT, which "AUTOSET" had retained as the return address to "REND3OS". Hence "AUTO37" is again entered, this time with the desired program equal to TEMPMM, i.e. in range of 81-88 (due to the start of "REND3OS").

"AUTO37" again performs the standard V37 processing, this time with the program number of 81-88. The table on page GENP-39 shows that in this case transfer is made to these program starting points ("Pxy"), all of which are in the Minimum Key Rendezvous writeup. If P31 originally keyed in, for example, then "P81", page MINK-3, is entered. There, the first thing that is done is performance of "MINKDISP", page MINK-1, which first writes over the "81" of the program number with "31". Then a checklist code display is generated (checklist 00017<sub>8</sub>), requesting a decision as to whether the "automatic" or "manual" sequence is desired. An ENTR means the manual sequence, causing AUTOSEQ bit to be set 0 and return to "P81". There, via "AUTOSET", the P31 computations (page BURN-2) are entered. The final display response there (exit from "VN1645", page BURN-15), transfers to the second line of "GOTOPOOH", page GENP-15. There, a performance of "AUTOCHK", MINK-1, will reveal that the AUTOSEQ bit is zero and hence the V37 display is flashed in the same fashion as described previously for e.g. P30 termination.

The other rendezvous targeting programs can use a similar "manual" mode of operation. P31, P32, P34, and P38 use the External Delta V interface with the burn programs, while P33, P35, and P36 use another interface. P37 is oriented towards a display presentation, and does not interface with a burn program.

This other interface is employed when it is desired to arrive at a certain point at a certain time, and is referred to as "Lambert targeting". These Lambert targeting programs reset a flagword bit (XDELVFLG) to 0 to indicate to P40/P41 that a Lambert burn, as contrasted with an External Delta V burn indicated by having the bit = 1, has been targeted. As previously mentioned, P33, P35, and P36 make use of this Lambert capability.

The major Lambert computation package is entered via "INITVEL", page REND-7, which includes various scaling and other initialization functions before entering "LAMBERT" (page CONC-5). If a Lambert burn is specified, "UPDATEVG" (page STER-4) causes an accumulation of velocity increment information to be maintained until the Lambert computation is complete, and then updates the velocity-to-be-gained vector: as for the External Delta V case, "S40.8" continues to be performed for steering and T<sub>go</sub>. If "UPDATEVG" discovers that a Lambert solution has been generated (checking a cell that is also set negative in "P40SXTY" for P40/P41), then "S40.9" (page STER-6) is established. This job enters "INITVEL" to do the Lambert calculations, and then, after correcting for earth oblateness gravity effects on the target information, it sets the checking cell NBRCYCLS negative to indicate to "UPDATEVG" on its next pass that a new solution is available. In addition to this (comparatively slight) change in the burn calculations, "S40.1" must use a different procedure to calculate the required thrust direction for the burn: this is done in "S40.1B", page STER-1.

If, instead of an ENTR to the checklist 00017<sub>8</sub> display, a PRO response had been given (again considering the P31 example), then "MINKDISP" would have entered "STARTAUT", where after some initializations a return (via AUTPOINT) to "P81" would have been done. There, via "AUTOSET", the usual V37 program-change logic would, as in the ENTR case, have led to "P31" with eventual exit to the second line of "GOTOPOOH". This time, however, the bit AUTOSEQ would still be 1, so that "AUTOCHK", page MINK-1, instead of returning to "GOTOPOOH" for the V37 display would have transferred to the address specified by AUTPOINT (i.e. the address loaded by "AUTOSET", the 4th line of "P81" on page MINK-3). Here, transfer to "BURNHOW", MINK-2, is done to determine the burn program to be selected. If the required velocity change is below ten fps, then P41 is selected; otherwise, P40 is selected (in either case, AUTPOINT now contains the return address from "BURNHOW").



Regardless of which burn program is chosen, "TIGNOW", page BURN-29, exits (for the nominal sequence) to the second line of "GOTOPOOH", where again an entrance to "AUTOCHK" is made, since AUTOSEQ bit = 1. Transfer is made to the contents of AUTPOINT, which by now has the address of the fifth line of "P81". This in turn causes a transfer to the second line of "P82" (page MINK-3), where an analogous sequence is started, this time using P32. The subsequent sequencing of the programs, ending with P48 (at the end of "P87"), can be seen from the other material in the MINK writeup. The P88 program is not part of the normal sequence of MINKEY programs, but instead is entered (by P38 DSKY selection and suitable DSKY responses subsequently) to do targeting, P52, a burn program, P20 maneuver, and P52 again.

The P8x programs are not intended for direct DSKY entrance, but instead represent a convenient way to make use of the V37 processing logic in the software: they can be considered as the "driver" for the various pre-thrust and thrust programs, with the pointer as to where the present sequence has progressed being contained in the AUTPOINT cell.

### IMU Alignment

The four basic programs used for IMU alignment are P51 through P54. P50 and P55 are also defined, for the interface with the OWS (determine relative coordinates and star angles respectively). There is also a capability to use sensor information measured by OWS sensors in a backup mode; the discussion below is concerned with the "normal" operation of the alignment programs P51 - P54 independently of the OWS configuration.

IMU orientation determination is accomplished (for use of normal optics) by P51, which starts at "P51", page INFA-1. After a check that the IMU is on (if it is not, alarm 0210<sub>8</sub> is generated and "GOTOPOOH" is entered), an option is available to coarse align to 0° for all 3 IMU gimbal angles; otherwise, routine "R53" (page OPTC-15) is entered twice to obtain sighting information to a pair of manually selected bodies (one of 37 stars stored in fixed memory, the sun, or the earth (positions likewise computed by CMC)): if a code of 00 for the "celestial body" is entered, then a unit vector is requested by N88 input (see "PLANET", page INFA-8). Following sighting of the two bodies and specification of their identification, routine "R54" (page INFA-13) is entered to display the difference between the measured angle between them and the "theoretical" angle. After a response to this display (indicating acceptance), the reference-to-stable-member matrix ( $[REFSMMA]$ ) deduced from the sightings is computed and stored, a bit set (REFSMFLG) indicating that a known IMU alignment (valid  $[REFSMMA]$ ) is considered to exist, and "GOTOPOOH" is entered. If backup optics is used, V37 E 53 E is keyed in, which causes "P51" to be entered too, but which causes "P51B" (page INFA-2) to perform "R56" (page OPTC-17) to obtain the sighting data instead of "R53". Aside from this difference, the computations associated with P51 and P53 are the same.

Once the IMU has been aligned (the REFSMFLG set), then P52 (starting at "PROG52", page INFA-3), for use of normal optics, can be used to perform realignment. Three options are available for this program: the "preferred orientation" (computed by P40 or P41, or uplinked: the PFRATFLG bit is used only to generate the initial

option code display); the "nominal orientation" (local vertical at a specified time); and the "[REFSMMAT] orientation" (to get rid of the effects of uncompensated IMU drift since the previous alignment). The first two can do coarse alignment (if any angle change exceeds  $1^{\circ}$ ) to the new angles, and then do fine alignment to the angles. All options then perform automatic star selection ("P52C", page INFA-5) if desired and then enter "R51", page INFA-11, to complete the processing of observation data. Instead of doing IMU coarse alignment, it is also possible to change the IMU orientation solely by gyro pulse torquing, in "GYCRS", page INFA-14.

"R51" begins by displaying the star code selected for the first star: this may be changed, or some other body loaded, if desired. "PLANET" is then entered to compute the unit vector to this body, and then "R52" (OPTC-12) is entered. This routine performs automatic optics positioning (if optics in proper mode), and requests marks via "R53" when optics first switched out of computer control. Shaft stop avoidance logic is included in "OPTTEST", page OPTC-1. After two bodies have been sighted and the "R54" display approved, gyro torquing angles are displayed in "R55" (page INFA-14). If these are approved, the IMU gyros are pulse torqued so as to make the information now in [REFSMMAT] (which was updated as necessary when the coarse align check of "CAL53A", page INFA-10, was done) indeed reflect the orientation of the IMU in inertial space. If the backup optics is used, V37 E 54 E is keyed in, which causes "PROG52" to be entered too, but which causes "R56" (as in P53) rather than "R52" to be selected. Aside from this difference (no automatic optics pointing is provided since R52 is not entered), the computations associated with P52 and P54 are the same.

### Tracking and Rendezvous Navigation

Considerable elaborateness is included in the program in order to allow tracking and, if desired, rendezvous navigation computations to be carried out while other computations are running. These computations are conducted under the auspices of program P20, which starts at "PROG20", page ORVN-1. The automatic entrance to this program for Rendezvous Targeting purposes (AUTOSEQ bit = 1) has already been discussed. A manual selection (from e.g. P00) of V37 E 20 E, of course, also causes "PROG20" to be entered. In this circumstance, the first display provided requests a choice to be made of the desired tracking option: #0 is OWS with "VECPOINT" (axis) pointing; #1 is celestial body with "VECPOINT"; #2 is rotation about a body vector; #4 is OWS with three-axis control; and #5 is celestial body with three-axis control. Following this, an N78 display (in "DOV6N78", page ORVN-1) for specification of the body axis of interest is provided, followed by an N79 display of the desired tracking deadband and, for #2, the maneuver rotation rate. This is the final display in the initialization sequence for options #0 or #4 (the "rendezvous tracking" options).

Options #1, #2, and #5 are considered to be "universal tracking", as opposed to "rendezvous", options, and result in the setting of the UTFLAG (bit 9 of FLAGWRD8). Operation in this mode of the program involves only special spacecraft attitude control logic, and allows some special program performance features (such as entrance to P27) that are excluded from the rendezvous mode. Hence it is important to be explicit in identifying which mode of operation of P20 is taking place if the performance of the CMC software in some circumstance is of interest.



If option #2 was specified, the next display is of VO6N34 (in "P20OPT"), to indicate the time when the maneuver is to be started. This is the final display in this option before entrance to "NDUTINPT", page ORVN-3. If option #1 or #5, however, was specified then "V1N70DSP", ORVN-3, is entered so that the desired body can be specified via the VO1N70 display (and, for the "planet" option of 00, the VO6N88 display). Exit is then made to "NDUTINPT", where the UTFLAG is found 1 from "P20OPT". As a result, "STATINT1" is established and then exit to "PIKUP20", page ORVN-4, is made. The "STATINT1" routine, for periodic state vector updating, is the same routine that is established by the V37 logic if POO is selected, and has been previously described. The "PIKUP20" routine, also entered due to restart group settings e.g. after Average-G termination, after checking flags exits (since UTFLAG = 1) to "CALLR6X".

"CALLR6X", page ATTM-22, determines whether the option selected was 1/5 or 2. If #1/5, exit is to "R66CSM", page ATTM-23, which enters "R61CSM", page ATTM-15, every half second. There, the required attitude to point the specified vehicle axis at the specified point is determined ("UTAREAL", page ATTM-20, plus, for 3-axis maneuver, "UTOPT45" on ATTM-21). This loop is continued until interrupted (e.g. by resetting the TRACKFLG bit). If, on the other hand, option #2 was chosen, then "CALLR6X" delays until the N34-specified time is reached, and then enters "R67", page ATTM-23. There, the desired rate is specified and then the terminal part of the standard R60 automatic maneuver routine ("NEWANGL", page ATTM-11) is entered. The program stays in the one-second maneuvering loop until some other action occurs (the "predicted time of maneuver completion" is set to a large number), or an update of some matrix elements is necessary (about once every 34 minutes).

For the rendezvous options, after the response to the VO6N79 display exit is to "NDUTINPT", with, however, the UTFLAG = 0. As a result, both vehicle state vectors are integrated to the present time, the "R22" measurement data incorporation routine is established, and "PIKUP20" is entered. Here, "R61CSM" (ATTM-15) is entered, followed by entrance to the "R52" optics pointing routine which causes "R61CSM" to be initiated periodically to maintain the proper vehicle attitude.

The "R61CSM" routine, page ATTM-15, enters "CRS61.1", page ATTM-16, to do the necessary computations. If it is concluded that a maneuver is necessary (error exceeds  $10^0$  using DAP deadband center and required angles for option 4/5 or pointing angle information for option 0/1), then an exit to "STKTEST", page ATTM-18, is made. There, logic is entered to cause an estimate of the maneuver time to be made by the R60 computations, and then the relative positions of the CSM and OWS are recomputed for about 20 seconds after the end of this time interval (via "RCYC1R61", page ATTM-19), so as to minimize the effects of the changes in relative geometry during the maneuver. If this second computation indicates that a maneuver is still required (and is appropriate), then "R61CSM" will cause "R60CSM" to be entered to carry out the maneuver. At the end of the maneuver ("ENDMANUV"), if both AUTOSEQ and TRACKFLG bits = 1, an exit from the R60 logic is made. Special restart provisions are made on ATTM-3 ("CHKLINUS"/"RELINUS") to handle the case of e.g. target program selection after P20 initiated (although the automatic rendezvous logic should make this an unlikely procedure).

The optics pointing computations, "R52", start on OPTC-12. For the OWS tracking case, "R52H" periodically enters "R61CSM" and then "R52D" to update the optics pointing angles. Due to the setting of the R21MARK bit in "PIKUP20", optics marks can be taken at any time, causing entrance to "MARKRUPT", page OPTC-9, where, from "MARKDIF", optics marks are stored in an interface buffer, in "PUTMARK", for use by R22.

The measurement incorporation portion of P20 is "R22", which is established by "NDUTINPT" and which starts on page MEAS-4. Once entered, this job runs with priority 26<sub>8</sub>, and therefore takes precedence over the R52 loop that runs with priority 14<sub>8</sub> (due to "PIKUP20"). It should also be noted that the restart group settings (e.g. 2.7 and 1.11) for both R22 and the "PIKUP20" area are only priority 10<sub>8</sub>, whereas V37 processing establishes jobs with priority 13<sub>8</sub> ("V37XEQ", page GENP-22). As a result, even though for most program changes the TRACKFLG is reset in "ROO", if tracking is still appropriate for the new program (such as P30 or P21), the new program will be initiated, will set the TRACKFLG = 1, and will pause at its first display before the effect of the P20 restart groups is implemented. Hence when "PIKUP20", for example, checks TRACKFLG it would be 1 if the P20 computations are allowed to run in the background of the newly selected program. After some initial conditions, R22 enters "WAITONE", page MEAS-13, causing a 4 second delay in R22 performance. Then, if R61 is not doing a maneuver and updates are allowed, return is to "REND1", page MEAS-4, where the interface optics buffer "PUTMARK" loads is checked. If it contains mark data, it is sampled for use and the R22CAFLG bit is set. This bit is employed in "MKREJECT", OPTC-11, to determine if optics mark processing is being carried out: if set, then the REJCTFLG bit, checked at the end of "REND7", is set. Hence if it is desired to reject an optics mark, it is necessary to do so before the R22 computations have processed it (assuming the mark was not so erroneous that the N49 excessive update display is triggered). VHF ranging marks are also obtained periodically in the R22 loop by "REND3", page MEAS-5, if authorized (V87E).

If computations at the end of "REND7" (page MEAS-9) conclude that either the position or the velocity state vector update is "excessive" (bigger than tolerances stored in erasable memory), then "RENDISP", page MEAS-12, is entered. This routine establishes "RENDISP2" with priority 27<sub>8</sub> (same page) to generate a priority display using VO6N49 and awaits crew response. Thanks to the "WAITONE" logic (for optics and for radar except when waiting for radar sample to accumulate), it is known that the R60 priority display is not on the DSKY, and to avoid having it be initiated "RENDISP" performs a short loop checking a communication cell with "RENDISP2" to determine when a reply is received (and of what type). R22 and R52 cycling can be halted by V56E (causing "TRACKTRM", page VBDF-8, to be entered) or by selection of POO. Cycling is suspended (but can be started up again) if the TRACKFLG = 0 (e.g. entrance to a burn program). Backup marks can be initiated by keying V54E, causing "R23CSM", page OPTC-11, to be established. If the AUTOSEQ bit is set, "REND4" (MEAS-8) exits to "AUTOW" (MINK-5) to check for automatic W matrix initialization.



## Other Navigation Capability

Another navigation program is P21 (which starts on page ORVN-5). This program merely gives the latitude, longitude, and altitude at a specific time (incremented automatically by 10 minutes if desired). Other quantities are also computed for optional display. A similar program, P29 (page ORVN-11), may be used to derive the time (after an input time) when the spacecraft will arrive at an input value of longitude.

A capability also exists to process data from the VHF range measurement system in order to display values of either current or "optimized" range and range rate, as well as under some conditions an angle. The basic measurement provided by the VHF system is range, of course, with a least increment of 0.01 nmi: the filter process generates a suitable value for range rate from these raw measurements. The same hardware is used for the raw data for the display computation as is used for the rendezvous computations, but the processing that is carried out is different for the two systems. Separate weighting matrices are employed, for example, and a given range sample is only used for one of the two purposes. The rendezvous (R22) computations are based on updates to the state vectors maintained in the computer; those for the VHF display are based solely on the VHF range measurements. Hence the range and range rate display material from the VHF processing here is different from that which would be produced by R31 or R34 ("R31CALL", page DISP-10): these routines start with the base state vector in the computer and then extrapolate the results (in free flight) using conic integration, whereas the VHF processing display does not use these state vectors at all (except for angle computations).

There are three separate methods whereby the VHF processing computations for display purposes can be initiated. All three perform the computations of "R27GO" (page MEAS-14). One method is by a V76E ("V76CALL", page VBDF-12), which causes flag bits to be set enabling entrance to "R27GO" from "WAITONE" (page MEAS-13). As discussed before, "WAITONE" is the idling point for the R22 measurement-incorporation computations of P20 (option 0 or 4). Unless R22 is running (although the UPDATFLG could be reset, of course), therefore, the V76E method is ineffective. A second method is by separate selection of program P25 ("P25CSM", page ORVN-7): contrary to P20, this program does not require that IMU data be available, and it does provide automatic (N76 and N77) displays of the results of the "R27GO" computations (the use of R22 requires manual crew initiation of V16N76E/V16N77E to observe the results). Since the IMU is not required, no angle computations are performed for P25. The third way is by selection of P48 ("P48CSM", page BURN-30). This program too provides automatic displays, and in addition provides compensation of the filter results for accelerometer-derived velocity changes. P48 is similar to P47 (already discussed), except that "CALCN83", instead of computing N62 parameters for display, establishes a lower priority job ("R27JOBA", page DISP-2) to do the R27 computations provided the previous evaluation has been completed.



## Entry

Input of V37 E 61 E will permit "P61" (page ENTP-1) to be entered. Average-G is turned on via "S61.1" (page ENTP-4) which causes "MIDTOAV2" to be entered (the same routine as used by P47) in order to get the Average-G computations started (generally in 12.5 seconds from the beginning of the last time step). After this is done, the IMU alignment is checked in "S61.1A" (page ENTP-5), and then a N61 display of target parameters is supplied. Response to this display causes "NEWNRVN", page ENTP-1, to be entered to sample the state vector from the same two-second Average-G cycle (if the sampling is unsuccessful, the process is repeated). Routine "S61.2", page DISP-13, is entered to compute a number of quantities for display purposes, including Entry Monitor System (EMS) information. The N63 display (which includes time from EMS interface altitude) is updated every 2 seconds in "SERVICER" (page GENP-4), and is also corrected for the computation delay since sampling of state vectors. In addition, a V32E response to the display will cause the display parameters to be recomputed, and the N60 display to reappear. A PRO response causes "P62" to be entered.

"P62", page ENTP-2, can either be entered from "NEWNRVN" or can be selected directly by V37 means (see page GENP-39). It performs "S61.1" again (if from "NEWNRVN", Average-G is already running), starts the entry DAP going ("READGYMB", page DPEN-1), and then starts a flash of checklist code 00041<sub>8</sub> to request separation. A PRO response to this display causes "CM/DAPON" to be entered to activate entry DAP computations (page DPEN-1), and then "P62.1", page ENTP-3, to be entered. Here, after another N61 display of target information, a display of entry angles (via N22) as computed by "P62.3", page ENTP-4, is provided (unless the trim angle already is less than 45°, in which case P63 is entered immediately). Logic in "EXDAP", page DPEN-11, causes "WAKEP62" (page ENTP-4) otherwise to establish "P63".

"P63", page ENTP-4, changes the program number display to P63, sets the verb/noun display to VO6N64 (for use in "ENDEXIT", page ENRY-16), and causes "STARTENT", page ENRY-2, to be entered. This routine sets some initial conditions for the entry program and then causes the computations to enter "SCALEPOP" (page ENRY-3) and then "INITROLL" (page ENRY-5). The "INITROLL" computations monitor the status of the bit indicating 0.05 g (the bit is set/reset in "TARGETNG", entered from "SCALEPOP"), and when it first becomes one changes the program number display to 64 (and the verb-noun to 0674). If the velocity at this point is less than 27,000 fps ( $K_{vfn1}$ ), the exit address from "SCALEPOP" is set to "KEP2" (page ENRY-12): this causes exit to P67 (via "PREFINAL", page ENRY-12) when the drag exceeds 6.5 fps<sup>2</sup>. If velocity is above 27,000 fps, then a more complete set of entry computations can be done (the setting of INRLSW keeps the 27,000 fps check from being performed more than once).

"PREDICT3", page ENRY-13, is the "final phase" of the entry computations, and is entered until the velocity is below 1000 fps, when "P67.1" (page ENRY-16) is entered instead to display V16N67 (computed by "P67.2", same page) until a response is received, when the "GOTOPOOH" routine is entered to stop the computations.



## Digital Autopilots

Five distinct digital autopilots are included in the computer program: the Saturn DAP (mentioned in the Boost phase discussion, and used only to generate outputs based on hand-controller deflections); the RCS DAP (used for free-flight CSM control when undocked from the OWS); the TVC DAP (used for SPS thrusting, and including a roll control section using RCS jets); the Entry DAP; and the docked DAP (used for free-flight control when the CSM is docked to the OWS, and employing some of the same computations as the RCS undocked DAP, but a completely separate jet selection strategy). A capability also exists to "turn off" all the DAPs. The docked DAP can be initiated by performing routine RO4 (V44E, causing "DKDISP", page DPIR-3, to be entered) followed by V45E (entering "DKDAPON", same page). The entry DAP is initiated as part of the entry program sequencing, and the TVC DAP as part of the P40 burn-program sequencing. The Saturn, RCS, or no DAP initiation is accomplished by performing routine RO3 (V48E, causing "DAPDISP", page DPIR-2, to be entered) followed by V46E (causing entrance to "STABLISH", page DPIR-1).

RCS DAP operation is started by verb 46, as mentioned, which causes "STABLISH" to perform "RCS DAPON" (page DPRC-1). This routine sets T5LOC (the address to which transfer is made when computer clock TIME5 overflows) to "RCSATT", and causes TIME5 to overflow in 0.6 seconds. When the clock overflows, "RCSATT", page DPRC-1, is entered. There, provided that G&N control of the spacecraft is set, the DAP is initialized (via "REDAP", page DPRC-4). This includes initialization of the Kalman filter used for determination of the spacecraft attitude rate: while this initialization is taking place (about 1.1 seconds to obtain the necessary ten first-differences of CDU angles), no jets are fired, not even those which might be commanded by manual inputs (such as the translational controller). The complete DAP "computation cycle" is 0.1 seconds, and involves the sequence shown on page DPRC-36. Every other pass through the DAP (or every 200 ms), the attitude error needles on the FDAI are updated (using computations performed in "KMATRIX", page DPRC-3) based either on internal DAP-generated error signals or on the difference between the "desired" attitude and the present CDU angles. Selection of the type of display, and its optional initialization to present attitude, is made by verbs 60-63. Automatic operation for the performance of attitude maneuvers, also used for performance of rotational commands (in "T5PHASE2", page DPRC-7, logic), is controlled by the quantity HOLDFLAG, described on page DPIR-15. The selection of the jets to be used, and the checks for translation inputs, is made starting at "JETSLECT", page DPRC-15, and includes compensation for jet failures that have been reported via RO3 (see pages DPIR-29 and DPIR-30). Actual control of the jet on-time is accomplished by "T6START", page DPRC-25.

Transition from the RCS DAP to the TVC DAP is handled by the program, and cannot be accomplished manually. As mentioned in the discussion of the SPS burn program, the generation of translation-control inputs to the computer for ullage generation must be done by the crew. When "IGNITION" (page BURN-23) is entered to turn on the SPS engine, it halts the performance of the RCS DAP (by setting T5LOC to an idling location). 0.4 seconds later, T5LOC is set to "TVCDAPON", with an associated delay of 0.01 seconds. Consequently, 0.41 seconds after ignition "TVCDAPON", page DPTV-1, will be entered. This routine sets a number of DAP cells zero, and causes "TVCINIT1", page DPTV-1, to be entered 0.01 seconds later. There, other non-zero initial



conditions are set, and "MASSPROP", page DPIR-10, is entered to compute mass properties of the vehicle (in "FIXCW", same page) after initializing cells for "FIXCW" that remain undisturbed during the TVC DAP operation (hence, to save execution time, "FIXCW" is entered during the burn to update vehicle mass properties). "TVCINIT1" exits to "TVCINIT4" (page DPTV-3), which calls "TVCEXEC" in 0.5 seconds, or about 0.92 seconds after ignition, and also causes "DAPINIT", page DPTV-3, to be entered a DAP half-cycle later (for LM-off, 20 ms).

"DAPINIT" samples CDU<sub>1</sub> and CDU<sub>2</sub> for use in forming the past values of these quantities<sup>Y</sup> (to obtain in turn rate information), and sets T5LOC to "PITCHDAP" with a corresponding delay of a DAP cycle (for LM-off, 40 ms, bringing the total time since ignition to about 0.48 seconds). "PITCHDAP", page DPTV-6, calls (via T5LOC setting) "YAWDAP" (page DPTV-8) in a DAP half-cycle; "YAWDAP", in turn, causes "PITCHDAP" to be entered in a DAP half-cycle: for LM-off, this causes SPS actuator output information to be updated every 40 ms. Guidance commands start the first entrance to "S40.8" after STEERSW = 1 (setting to 1 done 2 seconds after ignition, if a "long" burn).

"TVCEXEC", page DPTV-4, is entered (due to "TVCINIT4" waitlist setting) for the first time at about 0.92 seconds after ignition. Here, a roll attitude error is computed and the FDAI error display is updated. Every 10 seconds, "FIXCW" is entered to update the vehicle mass-properties related information (see CSMMASS on page DPIR-12 for the updating history of the cell used as the independent variable for these computations). Provision is made for updating the pitch and yaw trim estimation after a suitable time interval has elapsed (for LM-on, no time delay is imposed before the first correction). In addition, every 0.5 second ("TVCEXEC" keeps calling itself at that rate until the TVC DAP is halted), 0.03 seconds after "TVCEXEC", the TVC roll DAP, starting at "ROLLDAP" (page DPTV-11) is entered to compute a jet on-time for roll attitude hold about the outer gimbal angle value sampled in "IGNITION". Contrary to the RCS DAP, no jet failure information is used; instead, quads are alternated separately for both positive and negative torques.

When the engine cutoff command is generated, "SPSOFF", page DPIR-5, is entered to turn off the engine. In addition, new trim values, if appropriate, are also loaded. For the normal cutoff via "ENGINEOFF" (page BURN-28), "RCS DAPON" is entered 2.5 seconds later, causing "RCSATT" to be entered 0.6 seconds later to start the RCS DAP as described previously. If "S40.8" detects a thrust failure, trim updates are halted (and mass updates there while the indicated failure persists), but it is not until a response to the V97 display in "CLOCKJOB" is received that there is a possibility of having the software cut off the engine and TVC DAP (barring some phenomena as a fresh start, of course).

As mentioned with the Entry phase, entrance to P62 starts the performance of the gimbal angle sampling for the entry DAP; after crew response to the request for separation confirmation, "CM/DAPON", page DPEN-1, is entered to disable the RCS DAP (by setting T5LOC to "T5IDLOC", page DPIR-1) and to set some initial conditions and flagword bits to permit "READGYMB", which is performed as a waitlist task every 0.1 second, to continue on to "BODYRATE", page DPEN-3.



There, T5LOC is set to "ATTRATES" (page DPEN-4) with a corresponding delay of 0.01 second. In "ATTRATES", a check is made of the entry guidance bit indicating 0.05 g's. If drag is below this level, "EXDAP", page DPEN-11, is entered; if above it, then pitch and yaw rate damping is performed and channel 5 loaded appropriately.

If "READACCS" (page GENP-2) senses that the entry DAP is active, it saves some angles computed by "READGYMB" and "ATTRATES", and sets a waitlist call for "SETJTAG" (page DPIR-1) to set a quantity positive non-zero. This setting, which occurs once every 2 seconds, causes "EXDAPIN" (page DPEN-5) to continue with computations leading to a strategy for firing roll jets. The appropriate pattern is loaded into channel 6 in "JETCALL2" (page DPEN-10), and waitlist calls are set for replacement of the pattern (if necessary). Checks are made in "GETON2", page DPEN-7, that the sum of the waitlist-call times will not exceed 2 seconds. Following this, "CM/FDAIR" (page DPEN-10) is entered to display errors on the FDAI or to save quantities for downlink transmission (if the "SETJTAG" task is not done, then "CM/FDAIR" is entered directly from "EXDAPIN").

Docked DAP operation is started by V45E, which as for the RCS (undocked) DAP causes "RCS DAPON", page DPRC-1, to be performed. A flag bit is employed to define which DAP (docked or undocked) is desired, and this bit used in "REDAP", page DPRC-4, to select the appropriate parameters. As mentioned previously, the docked DAP has a completely separate jet selection strategy, starting at "DKJSELECT", page DPDS-1. Because of the length of the computations, it is undesirable to have them performed with other interrupts (such as telemetry) locked out, but on the other hand it is undesirable to wait until the normal job scheduling algorithm has had a chance to recognize a new job of high priority. Because of this, the "DKJSELECT" computations cause "SUPERJOB", DPDS-1, to be entered promptly but at the job rather than task level so the other interrupts can occur.

### Telemetry

Every 20 ms (nominally, for the high bit rate), a signal is received from the telemetry system which causes "DODOWNTM", page TELE-1, to be entered. Four telemetry lists, plus a "dump" of all cells in erasable memory, can be sent, as described on pages TELE-9 through TELE-31. Most changes to the selection of the downlist that is to be sent are made as part of the V37 sequence (programs with their downlists are given on page GENP-39), although changes also take place elsewhere (such as at the end of "NEWRVN" for the transition from P61 to P62, at the start of "P11", and in "V73UPDAT", page UPLK-1, for the verbs causing P27).



## 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, Revision 2, "Apollo Guidance Program Symbolic Listing Information for Block 2," dated 20 November 1969.  
Referenced here as merely "3420.5-27".

Copies of that document may be obtained by the same means as this document was supplied.

## 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: "true values" greater than 1.0 are accommodated by suitable scaling, as described below. 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 value of a number in the computer register (evaluated using the bit weights shown) has been determined, its "true value" may be determined by multiplying this 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 true 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 for more details.

In many instances, the word length of 15 bits, permitting a number to be expressed to  $1: \pm 16384$ , 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 least 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 general, unless otherwise specified words can be considered to be double precision. The following subscripts are used in the programmed equations to indicate the type of number involved, in cases where this information is needed:

dp: Double precision.

sp: Single precision.

tp: Triple precision.

vn: Verb-noun pattern (for DSKY display): first two digits are the verb (bits 14-8 of the word) and last two digits are the noun (bits 7-1 of the word).

x, y, z: Vector components.

0, 1, 2 (on vector variables): Vector components, different significance than x, y, z. For example, x, y, z might refer to IMU coordinates and 0, 1, 2 to body (roll, pitch, yaw) coordinates. Not used with TS.

2 (on numbers): Binary value.

8 (on numbers): Octal value.

Numbers with no subscript are quoted in decimal.

In addition to the scale factor information, it is necessary to know the units in which quantities are expressed. Times are usually in units of "centi-seconds" (0.01 seconds). Powered flight navigation equations give position in units of meters and velocity in units of meters/centi-second. Angles are usually expressed in units of revolutions (one revolution is 360°). Where not reasonably obvious, units for the quantities are given as part of their definition.

## B Constants

Fixed scalar constants are denoted by  $K_j$ , where  $j$  is a subscript of arbitrary type (lower case letters and/or numbers) selected for its mnemonic usefulness (generally similar to the program notation itself). An additional subscript in capital letters may also appear: this quantity is a variable, and is used for indexing purposes.

Constants stored in erasable memory (so they can be changed without hardware implications) are denoted by  $C_j$ , where  $j$  is again a subscript of arbitrary type. If an initial condition is necessary for some cell, but it is subsequently updated by the program, the quantity is given the designation of a variable.

For the comparatively few constants which are vectors (or which can be treated as vectors, such as the IMU calibration constants), the notation employed is an underline under the  $K$  or  $C$  (i.e.  $\underline{K}_j$  or  $\underline{C}_j$ ) except in a few cases where mnemonic usefulness dictated some other presentation (such as unit  $\underline{X}$  for a unit vector in the  $X$  direction). Matrices are designated by their program notation, enclosed in square brackets (no special notation is used to distinguish between matrices which are constants and those which are stored in erasable memory).

## C Variables

Variables are generally designated by several upper case capital letters, sometimes with subscripts (the single letters  $K$  and  $C$ , however, are 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 ( $LdD$  appears in the listing as  $L/D$ ).
- i: A suitable capital letter is substituted as defined in the related equation information for "i" ("j" also used).
- 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  $+MGA$ ). If two "p" letters are in the symbol, these signify parentheses in the symbol in the listing.

An underline of the final 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 final subscript quantity (an additional subscript level or separated by a comma from the other subscript information) using the same symbol as for the vector, but without the underline:  $TS_x$ , for example, would be the X component of the vector  $TS$ .

A variable sometimes is used as a subscript itself, in which case it performs an indexing or address specification function. The capital letter E with such a subscript means the contents of the (generally erasable memory) cell whose address is given by the contents of the cell with the indicated symbol:  $E_{TS}$ , for example, is identical to VAR if TS contains the address of VAR. This notation, of course, signifies use of indirect addressing.

Matrices are designated by enclosing their identifications in square brackets. The matrix may have its own special symbol (such as  $[REFSMMAT]$ ), or it may be designated by the symbol of its first vector component (the first row of the matrix), but without the underline:  $[X_{sm}]$  is the matrix composed of  $X_{sm}$ ,  $Y_{sm}$ , and  $Z_{sm}$ . Some special notation for the W matrix information, to permit more convenient comparisons with official guidance equation information, is described on page MEAS-32.

Interpretive language operations in the guidance computer (see 3420.5-27) permit convenient manipulations to be performed on  $3 \times 3$  matrices, the elements of which are each stored double precision in sequential erasable memory cells (all elements of first row, then second row, and then third row). Individual elements of a matrix are designated by the matrix symbol followed by a subscript between 0 and 8:  $REFSMMAT_3$ , for example, is the first element in the second row of  $[REFSMMAT]$ ;  $REFSMMAT_3$  would be the vector formed of  $(REFSMMAT_3, REFSMMAT_4, \text{ and } REFSMMAT_5)$ , i.e. the second row of  $[REFSMMAT]$  (which gives the stable member Y axis in reference coordinates). The program notation for  $REFSMMAT_3$  would be "REFSMMAT +6", since elements are stored double precision with the most significant half of  $REFSMMAT_0$  at address "REFSMMAT".

#### 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 symbol would be indicated. In order to improve the clarity of the presentation, many of the program-step symbols are omitted from the equations, and others may not precisely correspond to the program step actually identified by the symbol in question. Computer addresses themselves must occasionally be mentioned: these are in CADR or 2CADR formats (single precision and double precision respectively, as described in 3420.5-27). 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, 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 can be considered to be one greater than the calling address (ignoring special provisions for program-step changes between fixed memory banks), but in some instances such as generation of program alarms, 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 programmed equations.

### Program Control

There are three types of program entities that may be considered to take place within the program. These may be classified as:

- a) A Subroutine, which performs a certain function and then returns control to the program entity which called it (Subroutines, of course, may in turn call other Subroutines).
- b) A Task, which is a short sequence of computations performed based either upon a time criterion or upon some external signal.
- c) A Job, which is a program entity (such as a routine to compute a large attitude maneuver, the processing of a keyboard or uplink input character, or the performance of a Lambert routine during powered flight guidance) of long duration (compared to a Task) which must be done in a definite sequence. For example, during powered flight the accelerometer data must be corrected for biases before the state vector updating computations are performed, and the state vector updated before steering commands are generated for the DAP (digital autopilot), and therefore these functions 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. This time delay has a least increment of one centi-second (0.01 second). Unless interrupts are inhibited, a program interrupt (which can be caused by a signal from the telemetry system, the uplink, either of the two

DSKY's, completion of a radar input, two DAP waitlist-type clocks, the "T4RUPT" clock, or the waitlist clock for the CM) causes termination of the computations for the Job and performance of the Task. Tasks, however, are not subject to interruption by other Tasks, but instead continue to completion. There is also a hardware monitoring function which can cause a "hardware restart" that could, if necessary, interrupt a task: in general, however, such restarts should not be encountered.

Jobs are sequenced with the aid of a priority system (see section VIIB of 3420.5-27), and are performed only if no Tasks must be performed (provided that interrupts are not inhibited). If no "productive" computations are required, then a routine called the "dummy job" is performed. This routine 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" (by temporarily setting its priority negative) until some event takes place: the occurrence of the event can "awaken" the job by setting its priority positive again. Jobs can optionally be assigned a set of working storage cells called a "VAC area" (see section VIIB of 3420.5-27). Addresses within this VAC area are referred to as "relative addresses" (relative to the start of the area assigned to the particular job), or as "push-down list addresses". Address 12D, for example, would be the 12th address from the start of the job's VAC area (the "D" emphasizes that the number is quoted in decimal, since octal is also used in the program).

The following program control terms are employed:

**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.

**Call "XXXX" in yy seconds:** Cause a Task with starting address "XXXX" to be entered into the waitlist system to be performed in yy seconds from the present time (yy has a least increment of one centi-second).

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

**End of job:** Terminate the performance of a Job, and transfer control to the "executive" system to initiate performance of the Job which has the highest priority of those remaining. If no Jobs are to be performed, the "dummy job" routine is entered.



End of task: Terminate performance of the waitlist-initiated Task, and transfer control to a routine which checks for other waitlist Tasks which may be due, causing resumption of previous computations (cf. Resume) if there are no such Tasks.

Establish "XXXX": Enter a Job with starting address "XXXX" in the executive system priority list to be performed when appropriate (the priority of the Job is also indicated at the point where it is established). A Job can optionally be established with or without a working storage (VAC) area, but this option selection generally is omitted in the programmed equations information.

If: Carry out the indicated manipulations provided that the indicated conditions are satisfied. Should the conditions not be satisfied, or after performing the indicated manipulations, continue on in sequence (unless, of course, a manipulation involved transfer of program control). To avoid a proliferation of program tags, the manipulations are indicated by indentations, with if necessary additional indentations for additional conditions. In a few cases, it was necessary to continue the indentation function to the following page, and in such cases (such as the top of page ATTM-17) the most recent "If" controlling the performance of the routine is repeated in parentheses in order to ensure that proper interpretation of the information is made.

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 upon the purpose of the routine. May be a "Proceed to" in program listing.

Proceed to "XXXX": Cause the program step with address "XXXX" to be the next one executed, and continue the performance of the program from that point. If no address is specified (as is true for some Subroutines which have multiple return addresses), "proceed" merely means continue with the next equation in sequence. May be a "Perform" in program listing.

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

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 of job and Establishing the Job again 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).



Resume: Start performing again the computations (in a Job) that were interrupted to perform the Task. Used only for Tasks not initiated by waitlist means, cf. End of task.

Return: Proceed to the Subroutine's calling program, from which transfer to the Subroutine was made via a Perform.

In addition to the above terms, frequent use is made of special program control functions to accommodate crew use of the display system. In general, an action requested by the computer program can have three crew responses: a terminate verb, a proceed command (verb 33 or the PRO button), or a recycle verb (verb 32)/data enter. Uses of the display system are buffered by the routines described in the Display Interface Routines writeup, such as "GOFLASH" (entered e.g. from bottom of ATTM-1). These routines cause the display to be activated and then put the Job to sleep awaiting a response. The three responses are indicated by the terms "terminate", "proceed", and "otherwise", and are shown as the result after crew response. In addition, an immediate return (such as "GOFLASHR" use on ATTM-1) may be available, in which case the "Perform", rather than "Proceed to", program control term is used. This immediate return may be used for restart protection and/or to set bits for blanking one of the display registers (or more). Finally, there are the special "VNFLASH"/"VNFLASHR" interface routines, discussed earlier (page 13), for which the terminate response is a direct exit, rather than via the caller: to indicate this (on e.g. BURN-1), the terminate action, for format consistency, is still shown, but is enclosed in parentheses.

### 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 Mathematical Functions writeup.

- 1) cos X: Cosine of X.
- 2) cos<sub>sp</sub> X: Single precision cosine of X.
- 3) cos<sup>-1</sup> X: Arc cosine of X.
- 4) 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).
- 5) limit: Cause the maximum value of a quantity to be less than the value indicated (unless otherwise specified, the magnitude is limited, with sign information preserved). Frequently done by means of a check for overflow.
- 6) modulo: Form a quantity, for X modulo Y, equivalent to the value of Y times the remainder from (X/Y). For example, 380° modulo one revolution is 20°.

- 7) 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) 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.
- 8) 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 zeros, and counting the number of shifts required.
- 9) set  $X = Y$  and  $Y = X$ : Exchange the contents of "X" and "Y".
- 10)  $\text{sgn}(X)$ : Complement the result if X is negative, and otherwise leave the result alone. Unless otherwise specified, if  $X = 0$  the result is also left alone (i.e. 0 is a "positive" number).
- 11) 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.
- 12) sign agreement: Force the signs of the various parts of a multiple-precision word to be the same.
- 13)  $\sin X$ : Sine of X.
- 14)  $\sin_{\text{sp}} X$ : Single precision sine of X.
- 15)  $\sin^{-1} X$ : Arc sine of X.
- 16)  $\text{unit}\underline{X}$ , or  $\text{unit}(\underline{A} + \underline{B})$ : 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 all vector components, before forming the unit, are less than  $2^{-21}$  in magnitude as stored in the computer register: if a cross product of 2 unit vectors is involved, this is a "true value" of  $2^{-19}$  (about 0.002 mr).
- 17)  $\log_e X$ : Natural logarithm of X.
- 18) Vector manipulations:
  - \* ( $\underline{A} * \underline{B}$ ): Vector cross product.
  - ( $\underline{A} \cdot \underline{B}$ ): Vector dot product.



18) Vector manipulations (cont):

(A B): Form individual components of result from the individual components of A and B (e.g. Y component of result is  $A_y B_y$ )

$|A|$ : Length of A.

$|A|^2$ : Square of length of A.

A = ( $a_1, a_2, a_3$ ): Form a vector A from the given components.

[A] B: Premultiply the vector B by the matrix [A] (interpreter operation MXV). The X component of the answer is equal to the dot product of the first row of [A] and B, etc.

B [A] : Postmultiply the vector B by the matrix [A] (interpreter operation VXM). The X component of the answer is equal to the dot product of the first column of [A] and B, etc. Same as premultiplication by the transpose of [A].

19) Scalar manipulations:

+ : If space before sign, addition operation. If no space before sign, increment to address (single precision unless otherwise specified). MPAC+0 is the most significant part of MPAC<sub>dp</sub>, while MPAC +1 would be the addition of 1 to the contents of MPAC. In a subscript, the second meaning applies.

$|X|$ : Absolute value of X.

B = ( $b_1, b_2$ ): Form a double precision number B, with most significant half  $b_1$  and least significant half  $b_2$ .

$\sqrt{X}$  : Square root of X.

$\cap$  ( $X \cap Y$ ): Form the logical product one bit at a time (a "masking" order). If a given bit in both X and Y is a binary one, then the result is a binary one; otherwise, the result is a binary 0.

- 20) A bit is "set to 1" or "set to 0" by forcing its value to be 1 or 0 respectively.
- 21) The difference between two angles (expressed in twos complement form, see 3420.5-27) can be performed in such a way as to give a ones complement result, noted where significant as "ones complement difference formed".
- 22) Loading of registers of less than full-word capacity (such as EBANK which has only 3 bits) is generally indicated for clarity as if the loading were done by a masking operation.



## Major Variables

The following quantities are used at several points in the program, and therefore have not been included in the list of "Quantities in Computations" at the end of each writeup.

CDU: Single precision twos complement value of the computer hardware CDU (Coupling Data Unit) registers used with the IMU (Inertial Measurement Unit). Scale factor B-1, units revolutions. Other CDU's are used with the optics, but unless so specified reference to "CDU" should be interpreted as the IMU CDU's.

DELV: Value of sampled accelerometer output, scale factor B14, units of accelerometer counts (5.85 cm/sec per count). When loaded during flight, only the most significant half of each component is set to the accelerometer output value, with the least significant half set zero. After compensation (in "1/PIPA"), the least significant half can become non-zero.

DSPTAB: Set of 12 single precision erasable memory cells containing "display table" information. DSPTAB+0 through DSPTAB+10 are used to control the display registers and other numerical displays on the DSKY (Display and Keyboard assembly), with the format given on page DATA-40. DSPTAB+11 is used to drive individual indicators with bit assignments given in the table below: required changes for the values of these indicators are flagged for output by setting bit 15 of DSPTAB+11 to 1.

DSPTM1, DSPTM2: Cells used for communication purposes with the display routines (several nouns are associated with the cells to permit displays of appropriate parameters). DSPTM1 also "NORMTEM1".

E: Notation employed for a quantity selected by the address information given by the subscript on E.  $E_{TS}^{dp}$  would mean a double precision word whose address is given by the contents of cell TS.

FLAGWRDi (i = 0 - 9): Set of ten words whose individual bits are used for program control purposes. FLAGWRD4 is used only with the Display Interface Routines logic, and hence is defined in that writeup, while the others are defined in the table below. The flagbit mnemonics, where applicable, are the same as those used in the Colossus 3 programmed guidance equations document (5520.5-205, 20 April 1971) in order to minimize the updating effort required, although in some cases (such as LMTRG, bit 10 of FLAGWRD1) the mnemonic may not be appropriate to the mission. More detailed discussions of individual bits can be found in suitable downlink functional description documents.

FIGWRD10, FIGWRD11: Words whose individual bits are used for program control purposes. See table below.

I, i, j: Index parameters (not necessarily explicitly contained in the program) used as a convenience in describing the performance features of the equations. Their use and significance are defined (unless obvious) where they are used.

MAX: The maximum capacity of a computer cell (octal 37777<sub>8</sub> for a single precision +MAX and octal 40000<sub>8</sub> for -MAX). Decimal equivalent is dependent on scale factor of quantity involved (if scaled B0, value for single precision is  $1 - 2^{-14}$  and for double precision  $1 - 2^{-28}$ ).

MODREG: Single precision cell containing the value of the current computer program number (or "major mode"), generally displayed also on the DSKY (the quantity is a decimal, not octal, number). A setting to -0 is made if a fresh start is performed, thus inhibiting display of the quantity.

MPAC: Multiple-precision (or multiple-purpose) accumulator, a set of seven cells assigned to each job. Frequently used for temporary storage purposes. MPAC+2 is a communication cell with internal DSKY programs (contains the machine address of the quantity to be manipulated).

R: Position state vector, scale factor B29, units meters. Program notation is "RN".

R<sub>att</sub>, R<sub>att1</sub>: Communication cells with the orbital integration package containing the value of the position vector after completion of the integration (i.e. "at time" specified). R<sub>att</sub> is contained in push-down list location OD and R<sub>att1</sub> in push-down list location 14D. Both are scaled B29 in units of meters. The use of the push-down list allows several jobs to perform orbital integration, with the required sequencing/lockout controlled by "INTSTALL" logic. In the Colossus 3 program, R<sub>att1</sub> could have a different scale factor from R<sub>att</sub> (for moon-centered computations), and hence separate cells are employed.

[REFSMAT]: Reference to stable member matrix, elements with scale factor B1. The reference coordinate system "is defined by the line of intersection of the mean earth equatorial plane and the mean orbit of the earth (the ecliptic) at the nearest beginning of the Besselian year in which the mission takes place. The X-axis is along the ascending node of the ecliptic on the equator (the equinox), the Z-axis is along the mean earth north



pole, and the Y-axis completes the right handed triad." Loaded for certain options of the inflight alignment routines, and initially in "P11". The first row gives the X stable-member axis in reference coordinates, so that pre-multiplication of a vector in reference coordinates by  $[REFSMMAT]$  gives the same vector in IMU coordinates.

S1, S2: Interpretive language registers (push-down list locations 40D and 41D) used as communication cells for single-precision address information.

T<sub>att</sub>: Communication cell with orbital integration package (cf.  $R_{att}$ ) containing the value of time tag associated with the state vector output, in push-down list location 12D with scale factor B28, units centi-seconds. Program notation is "TAT".

T<sub>decl</sub>: Communication cell with orbital integration package (cf.  $R_{att}$ ) containing the value of time to which integration is to be performed. Stored in push-down list location 32D, scale factor B28, units centi-seconds.

T<sub>evt</sub>: Time of major event (e.g. liftoff, engine on, or engine off, whichever took place most recently), scale factor B28, units centi-seconds.

T<sub>ig</sub>: Value of "ignition time" (displayed by N33), scale factor B28, units centi-seconds. Also used by "CLOKTASK" to compute a "time-to-go" as  $T_{now} - T_{ig}$  ( $T_{togo}$ , displayed by several nouns). Hence after ignition is replaced by predicted cutoff time; for an ENTR response to the V97N40 thrust-fail display, the most significant half is set to a negative quantity, causing the "time-to-go" display to be +59 59 (for  $T_{now}$  above about 9 minutes).

T<sub>now</sub>: Present value of the computer "clock" (registers TIME2, TIME1) hardware erasable memory cells, scale factor B28, units centi-seconds. Reset 0 in "P11", but can be updated by verb 55, verb 70, or verb 73.

T<sub>pptm</sub>: Time tag associated with  $R$  and  $V$ , scale factor B28, units centi-seconds. Program notation is "PIPTIME" (since it often is the value of the clock when accelerometers were sampled).

THETAD: Single precision twos complement value of the "desired" IMU CDU values, scale factor B-1, units revolutions. Can be loaded or displayed by noun 22, and is used as one of the display options for attitude error needles with RCS DAP (the cells themselves are not used internally by the DAP's: THETADX is used by the RCS DAP, which is a different parameter than THETAD). Cells also have notation (CPHI, CTHETA, CPSI).

TIME1, TIME2: See T<sub>now</sub>.



TS (and variants, such as  $TS$ ,  $TS_1$ , etc.): Real or dummy temporary storage cells, used for convenience in describing the performance of the equations. When used as a communication cell between routines (such as specification of verb-noun pattern to a display routine) generally would be a computer hardware register, such as the accumulator, in the actual program.

UNITR: Value of unitR, scale factor B1.

V: Velocity state vector, scale factor B7, units meters/centi-second. Program notation is "VN".

$V_{att}$ ,  $V_{att1}$ : Communication cells with the orbital integration package containing the value of the velocity vector after completion of the integration (i.e. "at time" specified).  $V_{att}$  is contained in push-down list location 6D and  $V_{att1}$  in push-down list location 20D. Both are scaled B7 in units of meters/centi-second. See  $R_{att}$ .

unitX, unitY, unitZ: Unit vectors in X, Y, and Z directions respectively, scale factor B1.

X1, X2: Interpretive language index registers (push-down list locations 38D and 39D) sometimes used as address communication cells or as counters (scale factor B14). Can be used to select cells or shift amounts in interpretive language. If used as address communication cells, complementing in loading and subsequent use sometimes omitted for clarity.

$X_{dc}$ ,  $Y_{dc}$ ,  $Z_{dc}$  (or  $[X_{dc}]$ ): Unit vectors, scale factor B1, used to define desired IMU orientation for communication with routines in Coordinate Transformations, or to define navigation base orientation. Notation also  $X_{nb}$  etc. in program.

$X_{sm}$ ,  $Y_{sm}$ ,  $Z_{sm}$  (or  $[X_{sm}]$ ): Unit vectors, scale factor B1, used to define the orientation of the "stable member" (i.e. IMU).

#### Meaning of bits in DSPTAB+11

<u>Bit</u>	<u>Meaning</u>
9	Program alarm.
8	Tracker alarm (optics). Also used for VHF data-good information (set e.g. by "LIGHTON").
6	Gimbal lock (controlled by "GLOCKMON").
4	No attitude.

### Meaning of bits in FLAGWRDO

FLAGWRDO also has notation "STATE". Initialized to 00000<sub>8</sub> by a fresh start; bits 13, 12, 11, 10, and 1 are set 0, with other bits left alone, in "INITSUB". Address is 0074<sub>8</sub>.

<u>Bit</u>	<u>Symbol</u>	<u>Meaning</u>
15		Not assigned.
14	JSWITCH	Bit set to 1 within orbital integration package to indicate that extrapolation of W matrix (rather than state vector) being performed.
13	P55.1FLG	Bit set 1 in "P55" to indicate that option 1 of P55 has been selected (a value of 0, set in "INITSUB", means option 2).
12	P50.1FLG	Bit set 1 in "P50" to indicate that option 1 of P50 has been selected (a value of 0, set in "INITSUB", means one of the other options).
11	ATMFLAG	Bit set 1 when "SXTSM" entered at its beginning (i.e. not from "GETUM" for R22), to indicate that STARCODE digit C should be examined in determining the processing that is done (N70/N71). Set 0 in "INITSUB", and also after being checked in "SXTSM".
10	P50FLAG	Bit set 1 in "OAVECS" (part of P50) to cause the check for alarm 401 <sub>8</sub> in "CALCGA" to be bypassed. Bit reset in "CALCGA" after being checked, and also is reset in "INITSUB".
9	NEEDLFLG	Bit set 1 (by V62E or V63E) or 0 (by V61E) to indicate that FDAI attitude error display should be of total attitude error (if 1) or DAP "following error" if 0. See bit 6 of FLAGWRD9.
8	IMUSE	Bit set 1 to indicate IMU "in use" (an alarm is generated if IMU is turned off with bit 1).
7	RNDVZFLG	Bit set 1 to indicate that P20 in rendezvous mode (options #0 or #4) has been enabled, to control proper selection of restart information when Average-G is stopped, and to serve other functions. P20 may be dormant even if bit is 1.
6	SGTMK	Bit set 1 to indicate that sighting marks have been initiated. Program notation "R53FLAG".
5	NC12FLG	Bit set 0 in "P31" and 1 in "P32", used to control the rendezvous equations common to both programs.

<u>Bit</u>	<u>Symbol</u>	<u>Meaning</u>
4	CYC61FLG	Bit set 1 in "STKTEST" (part of R61) to cause R60 computations to exit to "RCYCLR61" after determining required maneuver time, for use in computing a new CSM/object (e.g. OWS) relative position.
3	FREEFLAG	Bit used for temporary program control purposes, such as a communication cell with "R54" that is left at 0 if a V32E response is received, and otherwise is left at 1.
2	NCINTFLG	Bit used in the rendezvous targeting programs (P31, P32, and P33) to specify the type of orbital integration that is desired (1 for conic, 0 for precision).
1	P29FLAG	Bit set 1 in "P29", used to permit some computations common to P21 to be performed. Used in "HOPALONG" to determine if in first iteration (where bit reset).



### Meaning of bits in FLAGWRD1

Bit 12 of FLAGWRD1 is left alone, and other bits set 0, by a fresh start; bit 6 is set 0, with other bits left alone, in "INITSUB".

<u>Bit</u>	<u>Symbol</u>	<u>Meaning</u>
15	2JETSFLG	Bit set 1 in RO3 to indicate that two-jet X translation is desired (0 means 4 jets), ignoring failure information. Program notation "NJETSFLG".
14	STIKFLAG	Bit set 1 in RCS/docked DAP if RHC commands are provided or if $MGA > 75^\circ$ . Used to bypass automatic maneuvers. Bit reset 0 by V58E and in "ROO".
13	ERADCOMP	Bit set 1 to specify that coordinate computation should compute the required earth radius based on the Fischer model. A value of 0 for the bit means that the launch pad radius (6373338 meters) should be used. Program notation is "ERADFLAG".
12	NODOPO1	Bit set 1 at the start of "P11", to inhibit the selection of PO1 subsequently.
11	RCSBURN	Bit set 1 if an RCS burn (P41) is specified, and set 0 if SPS burn (P40) is specified. Program notation "ENG2FLAG".
10	LMTRG	Bit set 1 to indicate that the target being sighted is the OWS. Program notation "TARG1FLG".
9	NCLPFLG	Bit set 1 to compute NSR altitude difference, and 0 for NCC altitude difference, in rendezvous computations (used to control "EHCOMP").
8	CSMUPDAT	Bit set 1 to specify that CSM state vector is to be updated by navigation measurements (a value of 0 means OWS state vector). Can be set to 1 by V81E and to 0 by V80E. Program notation "VEHUPFLG".
7	UPDATFLG	Bit set 1 to indicate that updating of state vector by (optics or VHF) marks is allowed, and set 0 to indicate that such updating is not allowed.
6	IDLEFAIL	Bit set to 1 for 2 seconds in "V97P", to inhibit the engine fail detection logic of "S40.8" from being triggered after a PRO response to the V97N40 display (for the two-second interval).

<u>Bit</u>	<u>Symbol</u>	<u>Meaning</u>
5	TRACKFLG	Bit set to 1 to indicate that tracking via P20 is allowed (in either the rendezvous or universal tracking modes, see bit 7 of FLAGWRD0 and bit 9 of FLAGWRD8). A setting to 0 means that the P20 tracking should not be running.
4	MARKFLG	Bit set 1 for an optics mark (except in R21), used in the mark reject logic (except R21) to determine whether the mark reject should be accepted or a program alarm generated.
3	ITER1SW	Bit set 1 to indicate that first Lambert iteration is being performed, and then reset 0. Program notation "SLOPESW".
2	GUESSSW	Bit set 1 to indicate to Lambert iteration that no starting value is available; otherwise, it is set 0. Program notation "GUESSW".
1	AVEGFLAG	Bit set 1 when "READACCS" (accelerometer sampling and Average-G computations) is initiated, and reset (in "ISITPOO") to halt computations.

### Meaning of bits in FLAGWRD2

FLAGWRD2 is initialized to 00000<sub>8</sub> by a fresh start; bits 12, 11, 9, and 3 are set 0, with other bits left alone, in "INITSUB".

<u>Bit</u>	<u>Symbol</u>	<u>Meaning</u>
15	DRIFTFLG	Bit set 1 to enable free-flight gyro drift compensation.
14	R21MARK	Bit set 1 to indicate that reception of optics rendezvous primary marks is enabled, used to control the location in which mark information is stored.
13	ITERFLG	Bit used for communication purposes with the P31, P32, and P33 iteration routine "ITER": a setting to 1 means that convergence has not been achieved after the specified number of loop iterations (causing an alarm to be displayed).
12	P21FLAG	Bit set 1 in P21 after a "base vector" has been obtained: subsequent evaluations make use of the base vector instead of the present memory permanent state vector.
11	STEERSW	Bit set 1 to permit P40 cross-product steering and time-to-go computations (for "long" burn) to be performed.
10	SKIPVHF	Bit set 1 in "STARTSB2" as a flag for "VHFREAD" that a restart has taken place. The resulting radar interrupt is ignored.
9	IMPULSW	Bit set 1 in "S40.13" if a burn duration of less than 6 seconds is determined. It is used in "IGNITION" to cause the STEERSW bit setting to be bypassed, and to either set a waitlist call for cutoff or call "SBTASK" (depending on SBFLAG, bit 3 of this word).
8	XDELVFLG	Bit set 1 to specify that External Delta V guidance, as contrasted with "Lambert Aimpoint", has been targeted for P40/P41.
7	FIRSTFLG	Bit set 1 to indicate that a pass through "S40.9" has not yet been accomplished (set 1 at the start of "S40.1", and then reset 0 at the end of "S40.9").
	HAVEELEV	Bit set 1 in P35 to indicate that elevation angle is available, so $T_{tpi}$ should be computed and displayed. A zero value means the opposite. Bit set 0, if 1, on final pass in minimum key rendezvous mode if a revised N37 is entered via the DSKY. Program notation is "ETPIFLAG".



<u>Bit</u>	<u>Symbol</u>	<u>Meaning</u>
6	FINALFLG	Bit set 1 to indicate that final pass through rendezvous targeting computations is to be performed.
5	REVFLAG	Bit set 1 in P33 in order to restrict "GET.LVC" to derivation of matrix. Bit reset in "GET.LVC" after being sensed.
4	PFRATFLG	Bit set 1 to indicate to P52/P54 that a "preferred attitude" has been computed and is available. It is set in P40/P41 for burn attitude, and in the P38 minimum key rendezvous sequence after the plane-change information is available. The bit is used to select the initial VO4NO6 option display. A "preferred attitude" can be uplinked and used with the bit left at 0.
3	SBFLAG	Bit set 1 at start of P40 and in R40 to indicate requirement for accumulation of accelerometer data in "READACCS" for the 1-6 second special short-burn computations. It is reset in "S40.13" if it is determined that no requirement for this accumulation exists, as well as in "INITSUB".
2	CALCMAN2	Bit set 1 to specify that maneuver starting procedure should be done in R60 attitude maneuver package, and then reset.
1	NODOV37	Bit set 1 to cause V37 inputs of other than 00, unless in TVC mode, to be ignored. Set e.g. while periodic (P00 or P20 tracking) integration is done, while the P77 integration is done, when P06 is done, or when entry DAP is started. Program notation is "NODOFLAG".

### Meaning of bits in FLAGWRD3

Bit 13 of FLAGWRD3 is left alone, and other bits set 0, by a fresh start; bits 14 and 9 are set 0, with other bits left alone, in "INITSUB".

<u>Bit</u>	<u>Symbol</u>	<u>Meaning</u>
15	V50N18FL	Bit set 1 by V37 selection of a new program (except P00), and by V58E. It is set 0 upon completion of "R61CSM". If bit is 1 and attitude error is sufficient, then R60 is entered from R61.
14	GLOKFAIL	Bit set 1 if "CALCGA" generates a 401 <sub>g</sub> alarm due to computed middle gimbal angle in excess of 60°. Bit reset in "INITSUB" as well as in prelaunch programs.
13	REFSMFLG	Bit set 1 if a meaningful [REFSMMAT] is considered to be available.
12	500**FLG	Bit set 0 after PRO response to N87 (or "otherwise" to N89) in R04, and set to one after 500 <sub>g</sub> alarm is generated in docked DAP. Used to allow only one 500 <sub>g</sub> alarm per R04 computation.
11	501**FLG	Bit set 0 as described for bit 12, and set to one after 501 <sub>g</sub> alarm is generated in docked DAP.
10	VFLAG	Bit left at 1 by star selection routine if a suitable pair of stars has not been found.
9	POOFLAG	Bit set 1 to flag to orbital integration logic that doing periodic (P00 or P20 tracking) integration, in order to enable the logic checks for this type of integration. Bit reset in "INITSUB".
8	PRECIFLG	Bit set 1 for various orbital integration entrances except from "STATINT1" (periodic P00 or P20 tracking) for CSM. It is used in "TESTLOOP" to identify the special "STATINT1" integration, and bypass any integrations of less than a time step. Hence if bit is 1, integration is done up to the specified time (which may require a fraction of a time step).
7	CULTFLAG	Bit set 1 in star selection routine (P52/P54) if star is occulted by the earth or sun. Also set in "SXTANG1" for trunnion "true" angle outside the 0-90° range.
6	45/46FLG	Bit set 1 after a V45E (docked DAP), and set 0 after a V46E (if RCS DAP specified). Used in "REDAP" to determine which DAP, docked or undocked, should be started.

<u>Bit</u>	<u>Symbol</u>	<u>Meaning</u>
5	STATEFLG	Bit set 1 if the "permanent" state vector of the CSM or OWS is to be updated by the orbital integration package (used and reset in "A-PCHK"). Permanent state vector also updated if bit 1 (WMATINT) of FLAGWRD3 = 1.
4	CONICINT	Bit set 1 if conic integration is to be performed by orbital integration (if 0, Encke integration is done). Program notation "INTYPFLG".
3	CSMINTSW	Bit set 1 if CSM state vector is to be integrated by orbital integration package (if 0, OWS state vector is to be integrated). Can also enter the package at "INTEGRVS" with vector supplied in separate communication cells. Program notation "VINTFLAG".
2		Not assigned.
1	WMATINT	Bit set 1 if W matrix is to be integrated by orbital integration package (done for each time step after orbital integration of state vector). If 0, no W matrix integration is to be done. Program notation "DIMOFFLAG".



For bits in FLAGWRD4, see Display Interface Routines.

Meaning of bits in FLAGWRD5

FLAGWRD5 is initialized to 00000<sub>8</sub> by a fresh start; bits 10 and 7 are set 0, with other bits left alone, in "INITSUB".

<u>Bit</u>	<u>Symbol</u>	<u>Meaning</u>
15	DSKYFLAG	Bit set 1 in "KEYCOM" if a DSKY ( <u>not</u> uplink) input is received. If the bit is 0, as it is after a fresh start, then the drive of DSKY outputs in "T4RUPT" is usually bypassed (except for the remainder of the 0.12-second interval after DSPTAB+11 is transmitted).
14		Not assigned.
13		Not assigned.
12		Not assigned.
11	FSTINCRP	Bit set 1 if the first incorporation of measurement information in R22 is being made from optics data, and set 0 for second. In R22 for VHF, only one incorporation is made. Program notation is "INCORFLG".
10	NEWTFLAG	Bit set 1 in "HOPALONG" (part of P29) if another pass through a portion of the P21 computations is required, used to cause proper point in the iteration loop to be entered after computations in P21 are completed.
9		Not assigned.
8		Not assigned.
7	ENGONFLG	Bit set 1 when the SPS engine-on command is generated, and set 0 when it is removed. Note that it does <u>not</u> indicate that thrust has been detected. Bit used e.g. to restore the engine-on command after a hardware restart.

<u>Bit</u>	<u>Symbol</u>	<u>Meaning</u>
6	3AXISFLG	Bit set 1 if attitude maneuver package ("KALCMANU") provided with three angles to which maneuver is to be done (e.g. options #4 or #5 of P20, or "R62DISP"). If bit reset, maneuver calculated using "VECPPOINT".
5	BKUPLO	Bit set 1 when V75E (backup liftoff) is received. Program notation is "GRRBKFLG".
4		Not assigned.
3	NOSOLNSW	Bit set 1 if a conic routine (e.g. Lambert or time-radius) unable to provide a solution, and set 0 if a solution generated. Program notation is "SOLNSW".
2		Not assigned.
1	RENDWFLG	Bit set 1 if W matrix considered satisfactory for rendezvous navigation (R22 of P20).

### Meaning of bits in FLAGWRD6

FLAGWRD6 is initialized to 00004<sub>8</sub> by a fresh start; bit 13 is set 0, with other bits left alone, in "INITSUB". Also is "CM/FLAGS".

<u>Bit</u>	<u>Symbol</u>	<u>Meaning</u>										
15	DAPBIT1	DAP status bit (see bit 14).										
14	DAPBIT2	DAP status bit, used with bit 15 to indicate the DAP which is activated, and to select the appropriate starting point if a hardware restart										
		<table><tr><th><u>Bits 15-14</u></th><th><u>DAP Activated</u></th></tr><tr><td>00<sub>2</sub></td><td>None (or entry)</td></tr><tr><td>01<sub>2</sub></td><td>RCS/docked DAP</td></tr><tr><td>10<sub>2</sub></td><td>TVC DAP</td></tr><tr><td>11<sub>2</sub></td><td>Saturn DAP</td></tr></table>	<u>Bits 15-14</u>	<u>DAP Activated</u>	00 <sub>2</sub>	None (or entry)	01 <sub>2</sub>	RCS/docked DAP	10 <sub>2</sub>	TVC DAP	11 <sub>2</sub>	Saturn DAP
<u>Bits 15-14</u>	<u>DAP Activated</u>											
00 <sub>2</sub>	None (or entry)											
01 <sub>2</sub>	RCS/docked DAP											
10 <sub>2</sub>	TVC DAP											
11 <sub>2</sub>	Saturn DAP											
13	ENTRYDSP	Bit set 1 to cause display generation in entry program based on verb-noun in ENTRYVN (in "ENDEXIT"). Reset 0 to inhibit this display (e.g. while "P65.1" display is provided).										
	STRULLSW	Bit set 1 in "IGNITION" if steering is to be enabled 2 seconds after SPS ignition, and set 0 if it is not to be (i.e. short burn, for which ullage termination is the only function at that time).										
12	CMDAPARM	Bit set 1 to "arm" the entry DAP (to allow "entry firings and calculations").										
11	GAMDIFSW	Bit set 1 if GAMDOT (see Entry Computations) is to be calculated (as the difference of present and previous angle information).										
10	GONEPAST	Bit set 1 to indicate that lateral control calculations are to be omitted during entry. It is initialized to 1 in "STARTENT" and reset 0 in "INITROLL" after 0.05g sensed; in "PREDICT3" (P67) it is a "latched" version of bit 8 of FLAGWRD7, indicating target overshoot.										
9	RELVELSW	Bit set 1 in entry equations to cause earth-relative velocity to be used.										
8	EGSW	Bit set 1 in entry equations to indicate P67.										



<u>Bit</u>	<u>Symbol</u>	<u>Meaning</u>
7	NOSWITCH	Bit set 1 in "UPCONTRL" (entry) if drag exceeds 140 fps <sup>2</sup> , in order to inhibit a lateral switch in "LIMITL/D".
6	HIND	Bit set 1 in entry equations to specify that "HUNTEST" iteration is to be performed.
5	INRLSW	Bit set 1 in entry equations after 0.05g sensed, to indicate termination of the period when initial roll attitude is to be held.
4	LATSW	Bit set 0 in entry equations to force the entry DAP to "roll over the top" (rather than shortest route).
3	O5GSW	Bit set 1 when drag is over 0.05g, and reset 0 when less than that amount. Program notation ".O5GSW". Bit set to an initial condition of 1 in fresh start to bypass TTE computation in "SERVICER" (reset e.g. in "NEWNRVN" for P61, and also in "P62" before entry DAP started).
2	CMDSTBY	Bit set 1 when entry DAP computations are not to be inhibited (i.e. when entry DAP "activated"). Program notation "CM/DSTBY".
1	GYMDIF	Bit set 1 in entry DAP if CDU differences and body rates are to be computed (as difference of present and previous angles). Program notation "GYMDIFSW".

### Meaning of bits in FLAGWRD7

FLAGWRD7 is initialized to 00000<sub>8</sub> by a fresh start; bits 13, 12, and 11 are set 0, with other bits left alone, in "INITSUB".

<u>Bit</u>	<u>Symbol</u>	<u>Meaning</u>
15	TERMIFLG	Bit set 1 to terminate performance of "R52": set at end of R53 and (unnecessarily) R56.
14	ITSWITCH	Bit set 1 in P35 to force another iteration in the computation of TPI time. Bit also set in P34. Program notation is "ITSWICH".
13	IGNFLAG	Bit set 1 when nominal ignition time reached.
12	ASTNFLAG	Bit set 1 when crew authorization for ignition has been received.
11	TIMRFLAG	Bit set 1 to permit "CLOKTASK" to continue (loops at a one-second rate to update time-to-go, $T_{\text{togo}}$ ).
10	NORMSW	Bit set 1 to specify that <u>UN</u> is provided as an input from "INITVEL" package to Lambert; if 0, Lambert computes its own value (in "GEOM").
9	RVSW	Bit set 1 to inhibit calculation of final state vector by "COMMNOUT" (for e.g. time-theta or time-radius routines).
8	GONEBYTG	Bit set 1 in entry computations if an overshoot of the target is determined to exist. It is updated each computing interval to provide sign information for entry display of target angle. Program notation is "GONEBY".
7		Not assigned.
6	V37FLAG	Bit set 1 if the Average-G package ("READACCS" etc.) is running. If bit is 1, then V37 processing halts to wait for Average-G termination via "AVGEND".
5		Not assigned.

<u>Bit</u>	<u>Symbol</u>	<u>Meaning</u>
4	UPLOCKFL	Bit set 1 to lock out further uplink inputs (except error reset) after a complement (k $\bar{k}$ k) failure detected in uplink information.
3	VERIFLAG	Bit whose value is complemented when confirmation of P27 update (a V33E proceed response to F V21 NO2) is received.
2	LMATTCH	Bit set 1 in RO3 if DSKY input indicates CSM and LM are attached (not otherwise used).
1	TFFSW	Bit set 1 if calculation of perigee time is to be made, and set 0 if time of free fall is to be obtained.



### Meaning of bits in FLAGWRD8

FLAGWRD8 is initialized to 00000<sub>8</sub> by a fresh start; it is left alone in "INITSUB".

<u>Bit</u>	<u>Symbol</u>	<u>Meaning</u>
15		Not assigned.
14		Not assigned.
13	NEWIFLG	Bit set 1 by various entrances to integration package to indicate the initial pass through the integration, and set 0 in "TESTLOOP" after the logic concludes that performing periodic integration of the CSM state vector (in P00 or P20 tracking): effect is to inhibit start of integration of this type unless time tag at least 4 time steps behind present time.
12		Not assigned.
11		Not assigned.
10		Not assigned.
9	UTFLAG	Bit set 1 to indicate that P20 in tracking mode (options #1, #2, or #5) has been enabled, to control proper selection of restart information when Average-G is stopped, and to serve other functions. P20 may be dormant even if bit is 1.
8		Not assigned.
7	INFINFLG	Bit set 1 to indicate that no conic solution can be obtained (set several places in "GETX" Universal Variable Subroutine).
6	ORDERSW	Control bit used in "ITERATOR", expected to remain 0.
5	APSESW	Bit set 1 in time-radius routine if desired radius outside of pericenter-apocenter range, and otherwise set 0.
4	COGAFLAG	Bit set 1 if calling routine finds that COGA computation overflows (in Conic Routines).

<u>Bit</u>	<u>Symbol</u>	<u>Meaning</u>
3	V96ONFLG	Bit set 1 when a V96E (inhibit P00 integration) input received, and reset 0 when P00 integration is done. See bit 5 of FLAGWRD9. Could also be used to inhibit P20 tracking periodic integration, if P20 selected after the V96E.
2	R67FLAG	Bit set 1 in "R67" when the specified (via N34) time for start of rotation in option #2 of P20 has arrived, used to control logic in automatic maneuver routine.
1	360SW	Bit set 1 in "GETX" package if transfer angle near 360°.

### Meaning of bits in FLAGWRD9

FLAGWRD9 is initialized to 00000<sub>8</sub> by a fresh start; bit 15 is set 0, with other bits left alone, in "INITSUB".

<u>Bit</u>	<u>Symbol</u>	<u>Meaning</u>
15	SWTOVER	Bit set 1 in "SWICHOVR" (entered for V46E with LM-on TVC DAP) when the manually initiated change to the "low bandwidth" LM-on TVC DAP is made. Bit also initialized to 0 in "TVCINIT1".
14		Not assigned.
13		Not assigned.
12	MAXDBFLG	Bit set 1 in RO3 if maximum deadband selected, and set 0 if minimum deadband selected. Used in "FIXDB" and "INITSUBA" to restore the RCS DAP deadband to this selected value.
11		Not assigned.
10		Not assigned.
9	VHFRFLAG	Bit set 0 by V88E (or "V37"), and to 1 by V87E. If the bit is 1, R22 attempts to process VHF range data for state vector update purposes periodically.
8	VHFSOURC	Bit set 1 by R22 if the data source is VHF ranging, and set 0 if it is optics. Program notation is "SOURCFLG".
7	R22CAFLG	Bit set 1 while R22 is processing an optics mark, used if a mark reject is received to determine appropriate action.
6	N22ERNDS	Bit set 1 by V62E and 0 by V63E, causing data selected for total attitude FDAI error display in RCS DAP (see bit 9 of FLAGWRD0) to be THETAD (N22) if bit is 1 and CPHIX (N17, loaded with CDU by V60E) if the bit is 0. Program notation "N22ORN17".



<u>Bit</u>	<u>Symbol</u>	<u>Meaning</u>
5	QUITFLAG	Bit set 1 by V96E, to cause exit from the integration loop in "TESTLOOP". If the bit is 1, the periodic (POO and P20 tracking) integration is terminated in "STATINT1" (where bit is reset without another waitlist call for "STATEINT").
4	R31FLAG	Bit set 1 if R31 (V83E or P37) computations are selected, and set 0 if R34 (V85E) is selected.
3	MID1FLAG	Bit set 1 in "MIDTOAV1" to indicate that integration to the specified input time is desired, and reset if entrance to "MIDTOAV2" is made, or if the specified time is less than 12.5 seconds in the future after any time step (as deduced in "MIDTOAV1" or "CKMID2"). A reset value means to integrate to a certain time in the future (generally 12.5 seconds).
2	MIDAVFLG	Bit set 1 in "MIDTOAV2" (at a point after "MIDTOAV1" enters it) to flag the fact that the coasting flight to powered flight logic is desired. Causes "DIFEQ+2" to transfer to "CKMID2" at the completion of a time step. Bit reset in "INTEXT" at completion of integration.
1	AVEMIDSW	Bit set 1 in "AVETOMID" if "permanent" CSM state vector must be integrated for W-matrix purposes, and reset in "INTEXT". It is used to inhibit loading of CSM state vector cells at the end of the integration, since these cells (which otherwise would be used for downlink) contain the final results from the Average-G computations.

### Meaning of bits in FLGWRD10

Bit 11 of FLGWRD10 is left alone, and other bits set 0, by a fresh start. FLGWRD10 is left alone in "INITSUB".

<u>Bit</u>	<u>Symbol</u>	<u>Meaning</u>
15	PCMANFLG	Bit set 1 for minimum key rendezvous operation of P38, causing "PIKUP20" to exit after doing "R61CSM".
14	INTINUSE	Bit set 1 to indicate that the "INTSTALL" orbital integration routine has accepted a request for a use of the integration package. Program notation is "INTFLAG".
13	INTGRAB	Bit set 1 to indicate that a restart group phase change has taken place while the routine using the integration package (which previously set bit 14) is performing the computations. Program notation is "REINTFLAG".
12	REJECTFLG	Bit set 1 if a mark reject is received when optics in primary P20 mode and bit 7 of FLAGWRD9 = 1. Bit checked just before incorporation of optics mark information into the state vector. Can also be set (due to similar logic) by V86E.
11	HDSUPFLG	Bit used in "STARTAUT" to determine the appropriate initial condition for AZIMANGL (if 1, an angle of 0° is used).
10		Not assigned.
9	EXTRANGE	Bit set 1 in "CRS61.1" after a range magnitude from "R63" has been loaded into R63RANGE. If bit is 1, then "RANGERD7" corrects the VHF range sensor data for the fact that incoming measurement is made modulo $2^{15}/100$ nmi (327.68 nmi). Bit reset in "ROO".
8	P35FLAG	Bit set 1 in "P35/P75B" during the final computation cycle of P36: it is used in "AUTOW" to control W matrix initialization, and is reset in "REND5C". Bit notation a holdover from Colossus 3 program, where the P36 computations were in P35.
7	AUTOSEQ	Bit set 1 in "REND3OS", and left at that value for a PRO to the 00017 <sub>8</sub> checklist display in "MINKDISP". If 1, indicates the automatic rendezvous mode, also known as minimum key rendezvous, is in effect.

<u>Bit</u>	<u>Symbol</u>	<u>Meaning</u>
6	TCOMPFLG	Bit set in "P52AUTO" (P38 plane change in MINKEY) based on the signs of CDU <sub>y</sub> and y component of N81: set 1 if they have different signs, and 0 if they have same signs. Used to control the polarity of the pulse torquing that is done.
5	MANEUFILG	Bit set 1 in "VN1645" after final PRO response (except in P38), and reset in "REND12" after mark has been incorporated. Bit used in "AUTOW" logic to indicate that the first mark after a burn opportunity is being processed. Can also be set in "STARTAUT".
4	PTV93FLG	Bit set 1 in "AUTOW" if W matrix initialization after the next burn is desired. Can also be set in "STARTAUT".
3	TPIMNFLG	Bit set 1 in "P34/P74C" on the final pass of P35, and at the start of P36. It is reset 0 at the start of P35, and in "V37", and is used in "AUTOW" W matrix initialization logic.
2	FULTKFLG	Bit whose value can be changed by use of the display initiated by V57, and whose value is used to control "AUTOW" logic if TPIMNFLG and MANEUFILG bits both 1. A value of 1 for the bit means that do <u>not</u> have both optics and VHF sensors available.
1	PCFLAG	Bit set 1 at the start of P38, for use in controlling program logic, including minimum key rendezvous P52 sequencing.



### Meaning of bits in FLGWRD11

FLGWRD11 is initialized to 00000<sub>8</sub> by a fresh start, and is left alone in "INITSUB". Bits 11 and 6-1 are reset in "R00".

<u>Bit</u>	<u>Symbol</u>	<u>Meaning</u>
15		Not assigned.
14		Not assigned.
13		Not assigned.
12	R27FLAG	Bit set 1 after response to the N72 display initiated by V76E, and reset by V77E. If bit is 1, R27 is called from "WAITONE" (part of R22). Bit also reset at start of P20.
11	CYCLFLAG	Bit used to indicate if previous initiation of R27 still going on in P48 (set/reset in "R27JOBA", and checked in "CALCN83"). If bit is 0, then "R27JOBA" is established to do another R27 cycle.
10	NOUNFLG	Bit set 1 if the automatic display of R27 results in P25 and P48 should use N77; a value of 0 means that N76 is used.
9	N77FLAG	Bit set 1 to inhibit storage into N77 cells (R1 and R2) at the end of the R27 cycle (in "STOR77"), so that these cells remain static. A value of zero causes N76 information to be loaded into N77 cells.
8	AZIMFLAG	Bit set 1 for options #4 and #5 of P20, used in "R63COM1" to control nature of computation (and in other places as necessary to distinguish between 3-axis and "VECPOINT" solutions).
7	FIXFLAG	Bit set 1 in "WOTNOW" to cause R27 to perform optimization for a given time (i.e. N72 is within 95 seconds of that time). If bit is 0, current time is being used for computation.
6	SNAPFLAG	Bit set 1 in "SNAPPY" when get within 20 seconds of N72 time. If bit is 1, R22 (in "WAITONE") is inhibited from cycling to update state vector.
5	P48FLAG	Bit set 1 at start of "P48CSM", and zero at start of "P47CSM": bit used to control computations common to both programs.

<u>Bit</u>	<u>Symbol</u>	<u>Meaning</u>
4	P25FLAG	Bit set 1 at start of "P25CSM", and checked in R27 computations to determine if that program is running.
3	TDFLAG	Bit set 1 to inhibit update of R3 of N77 (angle information) in R27 use with R22 or P48 (cell not updated anyhow in P25).
2	R27UP2	Bit set 1 in "TAKE2" (part of R27), after completion of second entrance for initialization purposes.
1	R27UP1	Bit set 1 in "TAKE2" (part of R27), after completion of first entrance for initialization purposes.

### Channel Assignments

Listed below are the channel (input/output interface) assignments of interest for this flight. For more complete discussions of the channel bits, see section IIE of 3420.5-27. By convention, channel numbers are given in octal.

<u>Channel</u>	<u>Bits</u>	<u>Function</u>
03	14-1	Most significant 14 bits from a 33-stage binary counter driven by a 102.4 kc signal derived from the computer master oscillator (which keeps running when the computer is placed in standby). Scale factor B23, units centi-seconds.
04	14-1	Next most significant 14 bits of the counter used for channel 03. Scale factor B9, units centi-seconds.
05	8-1(SM)	X-axis RCS jets for Service Module.

	<u>Jet</u>	<u>Quad</u>	<u>Rotation</u>	<u>Translation</u>
8	6	B	- Yaw	+ X
7	7	B	+ Yaw	- X
6	8	D	- Yaw	- X
5	5	D	+ Yaw	+ X
4	2	A	- Pitch	+ X
3	3	A	+ Pitch	- X
2	4	C	- Pitch	- X
1	1	C	+ Pitch	+ X

05	8-1(CM)	Pitch and yaw RCS jets for Command Module (entry).
----	---------	--

	<u>Jet</u>	<u>Rotation</u>
8	6	- Yaw
7	7	+ Yaw
6	8	- Yaw
5	5	+ Yaw
4	2	- Pitch
3	3	+ Pitch
2	4	- Pitch
1	1	+ Pitch



<u>Channel</u>	<u>Bits</u>	<u>Function</u>			
06	8-1(SM)	Y-axis and Z-axis RCS jets for Service Module.			
		<u>Jet</u>	<u>Quad</u>	<u>Rotation</u>	<u>Translation</u>
	8	14	C	- Roll	+ Y
	7	15	C	+ Roll	- Y
	6	16	A	- Roll	- Y
	5	13	A	+ Roll	+ Y
	4	10	D	- Roll	+ Z
	3	11	D	+ Roll	- Z
	2	12	B	- Roll	- Z
	1	9	B	+ Roll	+ Z
06	4-1(CM)	Roll RCS jets for Command Module (entry).			
		<u>Jet</u>	<u>Rotation</u>		
	4	10	- Roll		
	3	11	+ Roll		
	2	12	- Roll		
	1	9	+ Roll		
07	7-5	Bits used to select appropriate portion of fixed memory for FBANK register contents (see 3420.5-27) of 30 <sub>8</sub> and above. Proper setting determined by program service routines, hence generally not shown in this document. Known as F EXT or SUPERBNK.			
10	15-1	Channel used to drive latching relays associated with the display system (bits 15-12 = 0001 <sub>2</sub> - 1100 <sub>2</sub> ). Loaded based on entrance to "T4RUPT". Known as OUTO.			
11	13	SPS Engine-on signal.			
	10	Caution Reset signal (for display system lights).			
	9	Test connector Outbit (set 1 in "READACCS" and 0 in "AVGEND", indicating when Average-G is running).			
	7	Operator Error light.			
	6	Flash of verb and noun lights.			
	5	Key Release light.			
	4	Temperature Caution light.			
	3	Uplink Activity light. Also used for other purposes.			
	2	Computer Activity light.			

<u>Channel</u>	<u>Bits</u>	<u>Function</u>
11(cont)	1	ISS Warning light.
12	15	ISS Turn-on Delay Complete.
	14	S4B Cutoff (set in "SPSOFF", reset in "ROO").
	13	S4B Injection Sequence start (reset in "ROO").
	11	Disengage optics DAC (digital to analog converter). Bit set in "S4O.6" (and in "REDOTVC" for a restart) to avoid driving optics system with TVC commands if optics left in Computer Control mode.
	10	Zero optics (reset in "ROO").
	9	S4B Takeover Enable.
	8	TVC Enable.
	6	Enable IMU CDU error counters.
	5	Zero IMU CDU's.
	4	Coarse Align IMU.
	3	Not used (reset in "ROO").
	2	Enable Optics CDU Error Counters (also used for TVC).
	1	Zero Optics CDU's.
13	15	Enable TIME6 countdown (at 1600 pps rate).
	14	Reset trap 32 (bits 10-1 of channel 32).
	13	Reset trap 31B (bits 12-7 of channel 31).
	12	Reset trap 31A (bits 6-1 of channel 31).
	11	Enable Standby.
	10	Test DSKY lights (used for alarms etc.).
	9-8	Not assigned (reset in "ROO").
	7	Telemetry word order code bit.
	6	Block cell 0045 <sub>8</sub> (uplink inputs etc.).
	5	Connect crosslink input to cell 0045 <sub>8</sub> .

<u>Channel</u>	<u>Bits</u>	<u>Function</u>
13(cont)	4	Bit set to 1 to initiate transmission of VHF data to cell 0046 <sub>8</sub> (if bits 3-1 = 001 <sub>2</sub> ).
	3-1	Bits set to 001 <sub>2</sub> to specify that range information from VHF ranging system is to be provided (see bit 4).
14	15	Generate output pulses from CDUXCMD.
	14	Generate output pulses from CDUYCMD.
	13	Generate output pulses from CDUZCMD.
	12	Generate output pulses from TVCYAW/CDUTCMD.
	11	Generate output pulses from TVCPITCH/CDUSCMD.
	10	Generate output pulses from GYROCMD.
	9	Negative gyro torquing command.
	8-7	No gyro axis if 00 <sub>2</sub> X gyro if 01 <sub>2</sub> Y gyro if 10 <sub>2</sub> Z gyro if 11 <sub>2</sub>
	6	Enable power supply associated with gyro torquing.
15	5-1	Main panel DSKY key code input.
16	7	Optics mark reject.
	6	Optics mark.
	5-1	Navigation panel DSKY key code input.
30	15	Bit sensed 0 if stable member temperature within design limits.
	14	Bit sensed 0 if ISS has been turned on or commanded to be turned on.
	13	Bit sensed 0 if IMU fail indication generated.
	12	Bit sensed 0 if CDU fail indication generated(IMU CDU's).
	11	Bit sensed 0 if IMU cage command generated (by crew).
	10	Bit sensed 0 if CMC given control of Saturn.
	9	Bit sensed as 0 if IMU turned on and operating with no malfunctions.
	7	Bit sensed as 0 if optics CDU fail indication generated.
	5	Bit sensed as 0 if liftoff signal generated by S4B Instrumentation Unit.



<u>Channel</u>	<u>Bits</u>	<u>Function</u>
31	15	Bit sensed as 0 if computer in control of spacecraft ("G&N Autopilot Control").
	14	Bit sensed as 0 if crew selects "Free" mode.
	13	Bit sensed as 0 if crew selects "Hold" mode. If bits 14-13 are 11 <sub>2</sub> , this indicates that "Automatic" mode selected.
	12	Bit sensed as 0 if translation in -Z direction commanded.
	11	Bit sensed as 0 if translation in +Z direction commanded.
	10	Bit sensed as 0 if translation in -Y direction commanded.
	9	Bit sensed as 0 if translation in +Y direction commanded.
	8	Bit sensed as 0 if translation in -X direction commanded.
	7	Bit sensed as 0 if translation in +X direction commanded.
	6	Bit sensed as 0 if rotation commanded in negative roll direction.
	5	Bit sensed as 0 if rotation commanded in positive roll direction.
	4	Bit sensed as 0 if rotation commanded in negative yaw direction.
	3	Bit sensed as 0 if rotation commanded in positive yaw direction.
32	2	Bit sensed as 0 if rotation commanded in negative pitch direction.
	1	Bit sensed as 0 if rotation commanded in positive pitch direction.
	14	Bit sensed as 0 if "proceed key" (formerly labeled Standby) is depressed.
	6	Bit sensed as 0 if negative roll commanded by minimum impulse controller.
	5	Bit sensed as 0 if positive roll commanded by minimum impulse controller.

<u>Channel</u>	<u>Bits</u>	<u>Function</u>
32(cont)	4	Bit sensed as 0 if negative yaw commanded by minimum impulse controller.
	3	Bit sensed as 0 if positive yaw commanded by minimum impulse controller.
	2	Bit sensed as 0 if negative pitch commanded by minimum impulse controller.
	1	Bit sensed as 0 if positive pitch commanded by minimum impulse controller.
33	15	Flip-flop bit sensed as 0 if computer oscillator has stopped.
	14	Flip-flop bit sensed as 0 if a computer warning indication produced.
	13	Flip-flop bit sensed as 0 if an accelerometer fail indication produced.
	12	Flip-flop bit sensed as 0 if a telemetry end-pulse rejected, indicating an excessive downlink rate.
	11	Flip-flop bit sensed as 0 if an uplink input bit is rejected, indicating an excessive uplink rate.
	5	Bit sensed as 0 if computer control of optics is set (by crew switch).
	4	Bit sensed as 0 if zero optics mode is set (by crew switch). If bits 5-4 = 1, manual mode selected.
	2	Bit sensed as 0 if VHF ranging data considered OK.
34	15-1	Output channel used to contain the first word of a pair telemetered periodically.
35	15-1	Output channel used to contain the second word of a pair telemetered periodically.
77	9-1	Bits indicating cause of restart (for telemetry; reset by "writing" into channel).

# Meanings of MODREG Values

<u>Decimal Value</u>	<u>Starting Location</u>	<u>Verb 37</u>	<u>Other Entrances</u>	<u>Title</u>
-0	---		"DOFSTART" "SOMERR2"	Fresh start (display of MODREG is suppressed).
00	---	x	"VERB96" "AUTO37"	CMC Idling.
01	"GTSCPSS"	x		Prelaunch Initialization.
02	"SLEEPIE"		"ESTIMS" "GTSOPTCS" (after P03)	Prelaunch Gyro Compassing.
03	"GCOMPVER"		"CKOPTVB" (V65E)	Prelaunch Optical Verification of Gyrocompassing.
06	"P06"	x		CMC Power Down (power up "POSTAND").
07	---			System Test (start in erasable memory, then to "GEOIMUTT").
11	"P11"		"CHKCOMED"	Earth Orbit Insertion Monitor.
20	"PROG20"	x	"P88" "REND30S"	Universal Tracking.
21	"PROG21"	x		Ground Track Determination.
25	"P25CSM"	x		Contingency VHF Range Rate.
27	"V73UPDAT"		"V7iUPDAT" (i = 0-3)	CMC Update.
29	"P29"	x		Time of Longitude.
30	"P30"	x		External Delta V.
31	"P31"	x	"P81"	NC1 Targeting.
32	"P32"	x	"P82"	NC2 Targeting.
33	"P33"	x	"P83"	NCC Targeting.
34	"P34"	x	"P84"	NSR Targeting.
35	"P35"	x	"P85"	TPI Targeting.
36	"P36"	x	"P86"	TPM Targeting.
37	"P37"	x	"P87"	Rendezvous Final Phase.
38	"P38"	x	"P88"	Plane Change Targeting.



<u>Decimal Value</u>	<u>Starting Location</u>	<u>Verb 37</u>	<u>Other Entrances</u>	<u>Title</u>
40	"P40CSM"	x	"BURNHOW"	SPS Thrusting.
41	"P41CSM"	x	"BURNHOW" "P88"	RCS Thrusting.
47	"P47CSM"	x		Thrust Monitor.
48	"P48CSM"	x	"P87"	Rendezvous Thrust Monitor.
50	"P50"	x		ATM Orientation Determination.
51	"P51"	x		IMU Orientation Determination.
52	"PROG52"	x	"P88"	IMU Realign.
53	"P51"	x		Backup IMU Orientation Determination.
54	"PROG52"	x		Backup IMU Realign.
55	"P55"	x		ATM Star Tracker Gimbal Angle.
61	"P61"	x		Entry Preparation.
62	"P62"	x	"NEWRVN"	Entry - CM/SM Separation and Pre-Entry Maneuver.
63	"P63"		"P62.1" "WAKEP62"	Entry Initialization.
64	"INITROLL" "KEP2" (if V below 27,000 fps at 0.05 g).			Entry - Post 0.05G.
65	"UPCONTRL"		"RANGER"	Entry - Upcontrol.
66	"KEP2"		"UPCONTRL"	Entry - Ballistic.
67	"PREFINAL"		"HUNTEST1" "KEP2" "UPCONTRL"	Entry - Final Phase.
77	"P77"	x		CSM Velocity Vector Update.
81-88	"P81"- "P88"		"REND3OS" (and in sequence)	Driver Programs for Minimum Key Rendezvous.

### Special Program Routines

<u>Routine Number</u>	<u>Starting Location</u>	<u>Title</u>
00	"GOTOPOOH"	Final Automatic Request Terminate.
	"V37"	Manual input of V37E xx E.
01	"ABCLOAD"	Erasable and Channel Modification Routine (V25 NO7 E).
02	"RO2BOTH"	IMU Status Check.
03	"DAPDISP"	CSM DAP Data Load (V48E).
04	"DKDISP"	Docked DAP Data Load (V44E).
07	"REND3OS"	MINKEY Controller.
08	"RANGERD"	VHF Range Read.
21	---	Rendezvous Tracking Sighting Mark (enabled when bit 14 of FLAGWRD2 = 1; no special displays).
22	"R22"	Rendezvous Tracking Data Processing.
23	"GOTOR23"/ "R23CSM"	Rendezvous Backup Sighting Mark(V54E).
27	"R27GO"	VHF Range Rate Mark Processing.
30	"V82PERF"/ "V82CALL"	Orbital Parameters Display(V82E).
31	"V83PERF"/ "R31CALL"	Rendezvous Parameter Display No. 1(V83E/P37).
34	"V85PERF"/ "R31CALL"	Rendezvous Parameter Display No. 2(V85E).
36	"V90PERF"/ "R36"	Rendezvous Out-of-Plane Display (V90E).
40	"CLOCKJOB"	SPS Thrust Fail (NWORD1 = -0).
41	"MIDTOAV1" "MIDTOAV2"	State Vector Integration.
50	"CAL53A"	Coarse Align.
52	"R52"	Automatic Optics Positioning.
53	"R53"	Sighting Mark.

<u>Routine Number</u>	<u>Starting Location</u>	<u>Title</u>
54	"R54"	Sighting Data Display.
55	"R55"	Gyro Torquing.
56	"R56"	Alternate LOS Sighting Mark.
60	"R60CSM"	Attitude Maneuver.
61	"R61CSM"	Tracking Attitude.
62	"CREWMANU"/ "R62DISP"	Crew-defined Maneuver (V49E).
63	"V89PERF"/ "V89CALL"	Rendezvous Final Attitude (V89E).
64	"R64"/ "V64PERF"	Optics Angles Transform (V64E).
66	"R66CSM"	Universal Tracking options #1/#5.
67	"R67"	Automatic Rotation Routine (Universal Tracking option #2).



## Checklist and Option Codes

### Checklist (V50 N25)

<u>Octal Code</u>	<u>Location</u>	<u>Significance</u>
0013	"P52D"	Key in normal or gyro torque coarse align.
0014	"R51K"	Key in fine alignment option.
0015	"P51A" "P52C"	Perform celestial body acquisition.
0016	"MKVB50" "R56"	Key in terminate mark sequence.
0017	"MINKDISP"	Perform MINKEY Rendezvous selection.
0020	"PERF20"	Perform MINKEY plane change gyro pulse torquing.
0041	"P62"	Switch CM/SM separation to up.
0062	"P06"	Key CMC to standby.
0204	"P4OSXTY"	Key in SPS gimbal trim test option.

### Option Selection (VO4 NO6)

<u>Octal Option Identification (in R1)</u>	<u>Octal Option Selection (in R2)</u>	<u>Location</u>	<u>Significance</u>
0001		"P52B"	Specify P52/P54 IMU Orientation.
	0001		Preferred Orientation.
	0002		Nominal Orientation.
	0003		[REFSMMAT] Orientation.
0002		"PROG21"	Specify P21/P29 Vehicle.
	0001		This vehicle (CSM).
	0002		Other vehicle (OWS).

<u>Octal Option Identification (in R1)</u>	<u>Octal Option Selection (in R2)</u>	<u>Location</u>	<u>Significance</u>
0012		"P50"	Specify P50 Option.
	0001		Sun only.
	0002		Sun and star.
	0003		Independent source.
0013		"P55"	Specify P55 Option.
	0001		[REFSMMAT].
	0002		Mark.
0024		"PROG20"	Specify P20 Mode.
	0000		Rendezvous, "VEGPOINT".
	0001		Celestial body, "VEGPOINT".
	0002		Body rotation.
	0004		Rendezvous, 3-axis.
	0005		Celestial body, 3-axis.
<u>Option Selection (VO4 N12)</u>			
0002		"V82CALL"	Specify R30 Vehicle.
	0001		This vehicle (CSM).
	0002		Other vehicle (OWS).
0004		"V57CALL"	Specify FULTKFLG Setting.
	0000		VHF <u>and</u> optics.
	0001		VHF <u>or</u> optics.

## Error Codes

The following error codes are included in the program. The "Use" column indicates the type of error code that is provided:

- B signifies an entrance to the "BAILOUT" routine (to generate a "software restart" for a nominally recoverable problem).
- D signifies that a display of the alarm code (VO5 NO9) is generated by the program after "ALARM" is entered.
- L signifies an entrance to the "ALARM" routine without an automatic NO9 display by program (the PROG light is energized).
- P signifies an entrance to the "POODOO" routine to generate a "software restart". If Average-G is running or EXTVBACT (13-1) is non-zero (e.g. extended verb with displays), has same effect as a "B" type; otherwise, all restart groups are reset, giving entrance to "GOTOPOOH".

Error codes are stored in the FAILREG set of cells, and may be examined by manual input of VO5 NO9 E (see page GENP-31).

<u>Octal Code</u>	<u>Use</u>	<u>Location</u>	<u>Significance</u>
00107	L	"TRKSTAR"	P55 star tracker angles out of limits.
00110	L	"MKREJECT"	Mark reject has been entered but ignored (previous marks, if any, already accepted).
00113	L	"MARKRUPT"	No input information in channel 16 when program interrupt #6 processed.
00114	L	"MARKDIF"	More marks made than desired.
00115	L	"VBCOARK"	V41 N91 E keyed with optics mode switch not at computer position.
00116	L	"OPTMON"	Optics mode switch moved from zeroing position before expiration of minimum time period (about 15 seconds).
00117	L	"VBCOARK"	V41 N91 E keyed but CMC has reserved optics CDU's (from gimbal test in P40 to end of burn). Not expected since V41N91 is P00 only.
00120	L	"OPTTEST"	Optics torque request with optics not zeroed since last fresh start or hardware restart.
00121	L	"MARKDIF"	An IMU CDU angle changed by more than $0.033^{\circ}$ in short interval (e.g. 0.05 seconds) after mark.



<u>Octal Code</u>	<u>Use</u>	<u>Location</u>	<u>Significance</u>
00205	L	"SERVICER"	Maximum accelerometer output (in excess of 6399 pulses in a 2-second period).
00206	L	"IMUZERO"	Request for zeroing of IMU CDU's rejected since in coarse align with gimbal lock.
00207	L	"IMUMON"	Inertial sub-system turn-on request not present for the required 90-second period.
00210	L	"IMUZERO" "P51" "R02BOTH"	IMU not operating.
00211	L	"COARS2"	Excessive angle error (more than 2°) after IMU coarse align.
00212	L	"C33TEST" "PIPFREE"	Accelerometer (PIPA) fail signal with failure inhibit set (to inhibit generation of ISS warning).
00213	L	"TNONTEST"	ISS turn-on request with IMU operate signal not present.
00214	L	"IMUMON"	Program using IMU when IMU turned off.
00217	L	"217ALARM"	Error return from "IMUSTALL" routine.
00220	L	"R02BOTH"	IMU not aligned (bit 13, REFSMFLG, of FLAGWRD3 is zero).
00401	L	"CALCGA" "UTOPT45"	Desired middle gimbal angle exceeds 60°.
00402	L	"PERF20"	Second MINKEY pulse torque request must be honored.
00404	D	"COM52"	Priority alarm. Optics not tracking OWS and target outside of 90° limit.
00405	D	"PICEND"	Acceptable star pair is not available.
00406	L	"GOTOR23"	V54E rejected since P20 rendezvous mode not running and/or bit 5 (TRACKFLG) of FLAGWRD1 is zero.

<u>Octal Code</u>	<u>Use</u>	<u>Location</u>	<u>Significance</u>
00421	L	"DIFEQ+2"	Overflow in orbital integration package (probably due to W-matrix).
00500	L	"P/YWALRM"	Not enough jets in docked DAP for pitch/yaw maneuver.
00501	L	"ROLLALRM"	Not enough jets in docked DAP for roll maneuver.
00600	D	"PHSMCH"	Failure in P31/P32 phase match iteration.
00601	D	"EHCOMP"	Failure in P31/P32 height maneuver iteration.
00602	D	"OLPALRM"	Failure in P31/P32 outer loop iteration.
00603	D	"GETRJ"	Failure in QRDTPI iteration (P31, P32, or P33).
00611	D	"P33/P73B" "P34/P74C"	No $T_{ig}$ for given elevation angle (P34/P35).
00777	L	"SETISSW"	A PIPA fail caused ISS warning.
01102	L	"ERRORS"	CMC self-test error.
01105	L	"C33TEST"	Downlink too fast.
01106	L	"C33TEST"	Uplink too fast.
01107	L	"GOPROG4"	Restart phase table entries disagree (hence fresh start done on assumption that the contents of erasable memory not valid).
01301	L	"ARCCOM"	Arc cosine or arc sine argument too big.
01407	L	"S40.8"	$V_g$ increasing.
01426	D	"S61.1A"	IMU orientation unsatisfactory for entry (program waits 10 seconds, then proceeds).
01427	D	"S61.1A"	IMU orientation reversed for entry, so ball reads $0^\circ$ in roll for lift down. Program waits 10 seconds, then proceeds.
01520	L	"AUTO37" "ISITPOO"	V37 (change major mode) cannot be accepted at this time.
01600	L	"SOMEERRR"	Overflow in prelaunch gyro drift test (probably).

<u>Octal Code</u>	<u>Use</u>	<u>Location</u>	<u>Significance</u>
01601	L	"SOMERR2"	Error return from mode switching or gyro torquing in prelaunch program (P01, P02, or P07).
01703	L	"CKMID2" "MIDTOAV1"	Ignition time slipped since insufficient time available for integration.
03777	L	"SETISSW"	ICDU fail caused ISS warning.
04777	L	"SETISSW"	ICDU and PIPA fails caused ISS warning.
07777	L	"SETISSW"	IMU fail caused ISS warning.
10777	L	"SETISSW"	IMU and PIPA fails caused ISS warning.
13777	L	"SETISSW"	IMU and ICDU fails caused ISS warning.
14777	L	"SETISSW"	IMU, ICDU, and PIPA fails caused ISS warning.
20430	P	"GOBAQUE"	Faulty state vector used for precision integration (e.g. position magnitude too low), causing overflow that rectification would not eliminate.
20607	P	"COMMNOUT" "TIMERAD" "TIMETHET"	No solution from conic routine.
21204	P	"LONGCALL" "WAITLIST"	Task insertion attempted with 0 or negative delay (see Section VIIA of 3420.5-27).
21206	P	"ENDIDLE" "NVSUBUSY"	Display system address buffers full, so another job cannot be accommodated.
21210	P	"IMUSTALL"	Job already waiting in IMU "stall" routine, hence another job cannot be accommodated.
21302	P	Square root (page MATH-4)	Square root argument too negative.
21501	P	"DSPALARM"	Display system alarm from internal ("NVSUB") use (such as an illegal noun).
21502	P	"FLASHSUB" "MAKEPLAY"	Illegal request for a flashing display (one already in system, so second cannot be satisfactorily accommodated).
21521	P	"GTSCPSS"	New program request (for P01) cannot be made after P11 started.



<u>Octal Code</u>	<u>Use</u>	<u>Location</u>	<u>Significance</u>
31104	B	"DELAYJOB"	Too many jobs (more than 4) attempting to use job delay routine.
31201	B	"FINDVAC"	No VAC areas available (see Section VIIB of 3420.5-27).
31202	B	"NOVAC2"	No job register sets available (see Section VIIB of 3420.5-27).
31203	B	"WAIT2" "WTLST5"	New task cannot be inserted successfully in waitlist system queue (see Section VIIA of 3420.5-27).
31211	B	"TESTMARK"	Optics use not allowed with extended verb being performed.
32000	B	"DKJSLECT"	Next docked DAP jet selection computation started before previous one completed.



## Job Priorities

Given below are the priorities with which the various Jobs are executed. A Job of lower priority will be suspended in favor of one of higher priority, except where prohibited by other program logic (such as the "INTSTALL" routine). The sequence with which Jobs of the same priority are performed is somewhat indeterminate (VAC-area Jobs of a given priority take precedence over non-VAC-area ones, see 3420.5-27), but they will all be done before Jobs of lower priority.

<u>Priority</u>	<u>Starting Address</u>	<u>Established By</u>
35 <sub>8</sub>	"NBDONLY"	"SVCT3"
34 <sub>8</sub>	"AMBGUPDT"	"KMATRIX"; "REDAP"
33 <sub>8</sub>	"MAKEPLAY"	"MAKEPLAY" (change)
32 <sub>8</sub>	"JAMTERM" "NORMLIZE"	"NVDSP" "PREREAD"
30 <sub>8</sub>	"CHARIN" "DSPMMJB"  "MONDO" "PROCKEY" "P47BODY" "UPJOB"	"KEYCOM" "GOPROG4"; "NEWMODEX"; "TSTLTS3" "MONREQ" "PROCEEDE" "TIGON" "UPSTORE"
27 <sub>8</sub>	"CLOCKJOB" "DYNDISP" "PHIJOB" "RENDISP2" "R23CSM"	"CLOKTASK" "CLOKTASK" "CTASK" "RENDISP" "R23CSM" (change)
26 <sub>8</sub>	"NEWDELHI" "REDOR22" "R22" "R67RSTRT" "R67START"	"UPDTCALL" "REDOR22" (change) "NDUTINPT"; "R22" (change) "R67RSTRT" (change); "UPDTCALL" "R67"
25 <sub>8</sub>	"ENDEXT"	"TICKTEST"
24 <sub>8</sub>	"ENDEXT" "R53JOB"	"ENTANSWR" "R52C"
22 <sub>8</sub>	"ALFLT" "ALFLT " "CHKCOMED" "MKVBDSPI" "MKVB5X"	"ALLOOP" "ALLOOP " "CHKCOMED" (change, ineffective) "MKREJECT" "MARKDONE"

<u>Priority</u>	<u>Starting Address</u>	<u>Established By</u>
21 <sub>8</sub>	"LASTBIAS" "1/CHECK"	"PREREAD" "1/PIPA"
20 <sub>8</sub>	"GTSCPSS" "POSTAND" "R27JOB" "S40.13" "SERVICER" "TIGNOW"	"GTSCPSS" (change) "PO6" (via restart) "READCYCL" "TIG-5"; "V97E" "READACCS" "TTG/O"
17 <sub>8</sub>	"R27JOBA"	"CALCN83"
16 <sub>8</sub>	"AZMTHCG1" "GCOMPVER" "P67.1" "RANGER" "R21END" "R23CSM"	"CHAZFOGC" "CKOPTVB" "PREDICT3" "RANGER" (change) "R21END" (change) "GOTOR23"
15 <sub>8</sub>	"P47BODY" "PLAYJUM1"	"P47BODY" (change) "NORMRET"
14 <sub>8</sub>	"ENDMANUV" "INITDSP" "PIKUP20" "P4OBLINKR" "RELINUS"	"ENDMANU" (restart logic) "PIKUP20" (change) "TIGBLNK" "RELINUS" (change)
13 <sub>8</sub>	"ENTANSWR" "GTSCPSS" "PROG20" "PROG21" "PROG52" "PO6" "P25CSM" "P29" "P30" "P31" "P32" "P33" "P34" "P35" "P36" "P37" "P38" "P40CSM" "P41CSM" "P47CSM" "P48CSM"	"ENTANSWR" (change) "V37XEQ"



<u>Priority</u>	<u>Starting Address</u>	<u>Established By</u>
13 <sub>8</sub> (cont)	"P50"	"V37XEQ"
	"P51"	"V37XEQ"
	"P55"	"V37XEQ"
	"P61"	"V37XEQ"
	"P62"	"V37XEQ"
	"P63"	"WAKEP62"
	"P65.1"	"RANGER"
	"P77"	"V37XEQ"
	"P81"	"V37XEQ"
	"P82"	"V37XEQ"
	"P83"	"V37XEQ"
	"P84"	"V37XEQ"
	"P85"	"V37XEQ"
	"P86"	"V37XEQ"
	"P87"	"V37XEQ"
	"P88"	"V37XEQ"
	"S61.1A"	"S61.1C"
	"VNRSTART"	"VNRSTART" (change)
12 <sub>8</sub>	"POSTBURN"	"ENGINEOFF"
10 <sub>8</sub>	"ATTACHIT"	"ATTACHED"
	"DKDISP"	"DKDISP" (change)
	"DONOUN46"	"DAPDISP"
	"LMTOCM"	"LMTOCMSV"
	"R62DISP"	"CREWMANU"
	"S40.9"	"SETUP.9"
	"V64PERF"	"R64"
	"V89CALL"	"V89PERF"
07 <sub>8</sub>	"R36"	"V90PERF"
	"VNRSTART"	"VNRSTART" (change)
	"V82GOFF1" (VAC area)	"V82GOFLP"
	"V82GON1" (VAC area)	"V82CALL"
	"V82PERF" (no VAC area)	"V82PERF" (change)
05 <sub>8</sub>	"R31CALL"	"P37"; "V83PERF"; "V85PERF"
	"STATINT1"	"NDUTINPT"; "ROO" (via restart logic); "STATEINT"
	"V67CALL"	"V67"
03 <sub>8</sub>	"V83CALL"	"R31CALL"
01 <sub>8</sub>	"ATERJOB"	"ATERTASK"
Same as user	"ENDRET2"	"ENDRET2" (change)
	"INITDSP"	"INITDSP" (change)
	"MAKEPLAY"	"GODSPRS+1"
	"NV5ODSP"	"NV5ODSP" (change)



## Erasable Memory Prelaunch Load

Listed below are the quantities in erasable memory which should be (or, in a few cases, can be) loaded as part of the prelaunch erasable memory load. The "Address" information gives the erasable memory bank format of the address, followed by the erasable CADR format (if different) needed if the cell is loaded by specification of the machine address. An asterisk to the left of the address means that the quantity would have to be loaded by uplink or DSKY means (other than by use of a noun assigned specifically to that quantity) in the event that re-initialization of the complete erasable memory were necessary in flight (these items exclude such quantities as the state vector etc., for which an uplink capability must exist for the nominal mission).

The "Reference" column gives one routine where the quantity has a significant application (or is updated), followed by the page in this document where the quantity is defined (scale factor, units, etc.).

<u>Address</u> <u>(Octal)</u>	<u>Memory</u> <u>Notation</u>	<u>Document</u> <u>Notation</u>	<u>Reference</u>	<u>Comments</u>
0075	FLAGWRD1	FLAGWRD1	"DOFSTART" pg. 47	Bit 12(NODOPO1) left alone until start P11, when bit set 1.
0077	FLAGWRD3	FLAGWRD3	"DOFSTART" pg. 51	Bit 13(REFSMFLG) left alone by fresh start.
0106	FLGWRD10	FLGWRD10	"DOFSTART" pg. 63	Bit 11(HDSUPFLG) left alone by fresh start.
*0373	C31FLWRD	C31FLWRD	"RCSATT" pg. DPRC-29	Backup for bits 15-13 of channel 31 and bits 5-4 of channel 33.
1016	N26/PRI	N26dPRI	"VBRQEXEC" pg. DATA-38	R1 of N26. Set 0 to lock out V30 and V31.
1122-3	PGNCSALT	C <sub>pgncsat</sub>	"P11" pg. BOOS-7	Altitude of IMU above 6373338 meters (used to compute initial conditions for state vector).
1124-5	PADLONG	C <sub>pdong</sub>	"P11" pg. BOOS-7	Pad longitude used to compute initial conditions for state vector at liftoff.
1333-4	FIXTIME	FIXTIME	"CTASK" pg. ORVN-14	N72. Initialize zero for non-optimization mode of R27.
*1356	CDUCHKWD	C <sub>cduchkwd</sub>	"MARKRUPT" pg. OPTC-19	Delay before checking IMU CDU after an optics mark.
*3,1452 (1452)	PBIASX	C <sub>pipabias</sub> <sub>x</sub>	"1/PIPA" pg. IMUC-20	PIPA <sub>x</sub> bias correction.

NOTE: References to "LM" are from Apollo 15 program, and have been retained for mnemonic usefulness.

<u>Address</u> <u>(Octal)</u>	<u>Memory</u> <u>Notation</u>	<u>Document</u> <u>Notation</u>	<u>Reference</u>	<u>Comments</u>
*3,1453 (1453)	PIPASCFX	C <sub>pipascf<sub>x</sub></sub>	"1/PIPA" pg. IMUC-21	PIPA <sub>x</sub> scale factor correction.
*3,1454 (1454)	PBIASY	C <sub>pipabias<sub>y</sub></sub>	"1/PIPA" pg. IMUC-20	PIPA <sub>y</sub> bias correction.
*3,1455 (1455)	PIPASCFY	C <sub>pipascf<sub>y</sub></sub>	"1/PIPA" pg. IMUC-21	PIPA <sub>y</sub> scale factor correction.
*3,1456 (1456)	PBIASZ	C <sub>pipabias<sub>z</sub></sub>	"1/PIPA" pg. IMUC-20	PIPA <sub>z</sub> bias correction.
*3,1457 (1457)	PIPASCFZ	C <sub>pipascf<sub>z</sub></sub>	"1/PIPA" pg. IMUC-21	PIPA <sub>z</sub> scale factor correction.
*3,1460 (1460)	NBDX	C <sub>nbd,x</sub>	"1/PIPA" pg. IMUC-20	X Gyro bias correction.
*3,1461 (1461)	NBDY	C <sub>nbd,y</sub>	"1/PIPA" pg. IMUC-20	Y Gyro bias correction.
*3,1462 (1462)	NBDZ	C <sub>nbd,z</sub>	"1/PIPA" pg. IMUC-20	Z Gyro bias correction.
*3,1463 (1463)	ADIAX	C <sub>ad,x</sub>	"1/PIPA" pg. IMUC-20	X Gyro input axis drift.
*3,1464 (1464)	ADIAY	C <sub>ad,y</sub>	"1/PIPA" pg. IMUC-20	Y Gyro input axis drift.
*3,1465 (1465)	ADIAZ	C <sub>ad,z</sub>	"1/PIPA" pg. IMUC-20	Z Gyro input axis drift.
*3,1466 (1466)	ADSRAX	C <sub>sr,x</sub>	"1/PIPA" pg. IMUC-21	X Gyro spin axis drift.
*3,1467 (1467)	ADSRAY	C <sub>sr,y</sub>	"1/PIPA" pg. IMUC-21	Y Gyro spin axis drift.
*3,1470 (1470)	ADSRAZ	C <sub>sr,z</sub>	"1/PIPA" pg. IMUC-21	Z Gyro spin axis drift.
*3,1700-2 (1700-2)	TEPHEM	T <sub>eph</sub>	"P11" pg. BOOS-10	Triple precision time between ephemeris origin and clock origin (updated at liftoff).
*3,1741 (1741)	EMDOT	C <sub>emdot</sub>	"S40.8" pg. STER-11	Mass flow rate for SPS engine.



<u>Address</u> (Octal)	<u>Memory</u> <u>Notation</u>	<u>Document</u> <u>Notation</u>	<u>Reference</u>	<u>Comments</u>
*3,1742-3 (1742-3)	RVAR	C <sub>rvar</sub>	"REND7" pg. MEAS-23	Value of (percent error/100) <sup>2</sup> for VHF ranging measurement (R22).
*3,1744-6 (1744-6)	RVARMIN	C <sub>rvarmin</sub>	"REND7" pg. MEAS-23	Triple precision complement of R22 minimum VHF variance (a negative number for coding ease).
*3,1747 (1747)	EIMPLSEC	C <sub>eimplsec</sub>	"S40.13" pg. STER-11	Impulse from first second of SPS thrusting.
*3,1750 (1750)	EFIMPO1	C <sub>efimp01</sub>	"S40.13" pg. STER-11	Slope of minimum impulse curve for SPS 0-1 second.
*3,1751 (1751)	EFIMPL6	C <sub>efimpl6</sub>	"S40.13" pg. STER-11	Slope of minimum impulse curve for SPS 1-6 seconds, i.e. thrust.
*3,1752 (1752)	LADPAD	C <sub>adpad</sub>	"STARTENT" pg. ENRY-18	Nominal L/D.
*3,1753 (1753)	LODPAD	C <sub>odpad</sub>	"STARTENT" pg. ENRY-18	Final phase L/D.
*3,1754 (1754)	DVTHRESH	C <sub>dvthresh</sub>	"S40.8" pg. STER-11	Velocity increment gate for Delta-V monitor.
*3,1755 (1755)	ALTVAR	C <sub>altvar</sub>	"REND-7" pg. MEAS-23	A priori measurement accuracy of back-up optics in R23.
*4,1400 (2000)	WRENDPOS	WRENDPOS	"REND5C" pg. MEAS-33	W matrix initialization for rendezvous. Can be changed by V67 processing, and "VN1645".
*4,1401 (2001)	WRENDVEL	WRENDVEL	"REND5C" pg. MEAS-33	W matrix initialization for rendezvous. Can be changed by V67 processing, and "VN1645".
*4,1402 (2002)	RMAX	C <sub>rmax</sub>	"REND7" pg. MEAS-23	Maximum automatic rendezvous position update.
*4,1403 (2003)	VMAX	C <sub>vmax</sub>	"REND7" pg. MEAS-24	Maximum automatic rendezvous velocity update.
*4,1404-5 (2004-5)	EMSALT	C <sub>emsalt</sub>	"S61.2" pg. DISP-19	Entry monitoring system initialization altitude.
*4,1406-7 (2006-7)	LOSO	C <sub>loso</sub>	"LSPOS" pg. COOR-11	Longitude of sun at beginning of AGC year.

<u>Address</u> <u>(Octal)</u>	<u>Memory</u> <u>Notation</u>	<u>Document</u> <u>Notation</u>	<u>Reference</u>	<u>Comments</u>
*4,1410-1 (2010-1)	CMOD	C <sub>cmod</sub>	"LSPOS" pg. COOR-11	Function of mean eccentricity used to compute sun position.
*4,1412-3 (2012-3)	CARG	C <sub>carg</sub>	"LSPOS" pg. COOR-11	Mean anomaly of earth-moon barycenter at beginning of AGC year.
*4,1414-21 (2014-21)	LMATRIX	C <sub>lmatrix</sub> <sub>0</sub>	"RP-TO-R" pg. COOR-11	First row of precession-nutation matrix.
*4,1422-7 (2022-7)	LMATRIX +6	C <sub>lmatrix</sub> <sub>3</sub>	"RP-TO-R" pg. COOR-11	Second row of precession-nutation matrix.
*4,1430-5 (2030-5)	LMATRIX +12 UNITW	C <sub>unitw</sub>	"CALCGRAV" pg. GENP-29	Earth polar axis unit vector (scale factor B0).
*4,1436-7 (2036-7)	AZO	C <sub>azo</sub>	"EARTHMX" pg. COOR-11	Angle between X-axis of Basic and Earth-fixed coordinate systems at beginning of AGC year.
*4,1440-1 (2040-1)	TCS	C <sub>tcs</sub>	"P33" pg. BURN-34	Time between NCC and NSR maneuvers.
*4,1442-3 (2042-3)	EPS1	C <sub>eps1</sub>	"PHSMCH" pg. REND-26	Loop convergence criterion.
4,1444-5 (2044-5)	DHNCC	DHNCC	"NC2LOOP" pg. REND-27	R2 of N57. Altitude difference at NCC.
*4,1446-7 (2046-7)	EPS2	C <sub>eps2</sub>	"EHCOMP" pg. REND-26	Loop convergence criterion.
4,1450-1 (2050-1)	DELH1	DELH1	"NC12F" pg. REND-27	R3 of N57. Altitude difference at (TPI) NSR.
*4,1452 (2052)	NH1	C <sub>nh1</sub>	"NC12ST" pg. REND-26	Number of revolutions between NC2 and NCC.
*4,1640 (2240)	WRDTIME	C <sub>wrdtime</sub>	"AUTOW" pg. MINK-8	Minimum time between automatic W matrix initializations.
*4,1641 (2241)	MINBLKTM	C <sub>minblktn</sub>	"AUTOW" pg. MINK-8	Check for time since last mark in automatic W matrix initialization.
*4,1642 (2242)	TBEFCOMP	C <sub>tbefcomp</sub>	"AUTOW" pg. MINK-8	Check of time until final target computation in auto W matrix logic.
*4,1643 (2243)	BRNBLKTM	C <sub>brnblktn</sub>	"AUTOW" pg. MINK-8	Interval w/o marks before & after doing a maneuver.

<u>Address (Octal)</u>	<u>Memory Notation</u>	<u>Document Notation</u>	<u>Reference</u>	<u>Comments</u>
*4,1644 (2244)	MAXWTIME	C <sub>maxwtime</sub>	"AUTOW" pg. MINK-8	Maximum desired time between successive W matrix initializations.
*4,1645 (2245)	FINCMPTM	C <sub>fincmptm</sub>	"AUTOW" pg. MINK-8	Time before maneuver of final mark.
*4,1646 (2246)	INTVAR	C <sub>intvar</sub>	"REND7" pg. MEAS-23	Integration variance employed in R22.
*4,1655-62 (2255-62)	NBOA +6	NBOA <sub>3</sub>	"P50" pg. INFA-23	+Y-axis of ATM Sensor in Navigation Base coordinates.
*4,1663-70 (2263-70)	NBOA +12	NBOA <sub>6</sub>	"P50" pg. INFA-23	+Z-axis of ATM Sensor in Navigation Base coordinates.
*4,1734-5 (2334-5)	RTRIDOT	C <sub>rtridot</sub>	"XTRAP" pg. MEAS-23	R27 W-matrix degradation factor.
*4,1773-4 (2373-4)	1/2ALPHA	C <sub>ld2alpha</sub>	"LOCSKIRT" pg. ATTM-25	Reciprocal acceleration for docked DAP used to compute BIASTEMP for R60 maneuver.
5,1400-1 (2400-1)	AZIMUTH	C <sub>azmth</sub>	"GTSC PSS" pg. PREL-10	Azimuth of vehicle Z-axis.
5,1402-3 (2402-3)	LATITUDE	C <sub>atd</sub>	"ESTIMS" pg. PREL-10	Launch pad latitude.
5,1432 (2432)	TAZEL1	TAZ <sub>1</sub>	"GCOMPVER" pg. PREL-13	Azimuth of optics target #1.
5,1433 (2433)	TAZEL1 +1	TEL <sub>1</sub>	"GCOMPVER" pg. PREL-13	Elevation of optics target #1.
5,1434 (2434)	TAZEL1 +2	TAZ <sub>2</sub>	"GCOMPVER" pg. PREL-13	Azimuth of optics target #2.
5,1435 (2435)	TAZEL1 +3	TEL <sub>2</sub>	"GCOMPVER" pg. PREL-13	Elevation of optics target #2.
5,1633-4 (2633-4)	LAUNCHAZ	LAUNCHAZ	"GTSC PSS" pg. PREL-12	Desired launch azimuth. Can be updated using V78E. Should be loaded with same bit pattern as would be produced by V78E, to avoid re-initializing of gyro compassing unnecessarily.

<u>Address</u> <u>(Octal)</u>	<u>Memory</u> <u>Notation</u>	<u>Document</u> <u>Notation</u>	<u>Reference</u>	<u>Comments</u>
*6,1400 (3000)	ETDECAY	C <sub>tdecay</sub>	"S40.8" pg. STER-12	Value of thrust decay (cell subtracted from time-to-go).
*6,1401 (3001)	EKPRIME	C <sub>ekprime</sub> <sub>0</sub>	"TVCINIT1" pg. DPTV-16	LM-off cross-product steering gain factor (for TVC DAP input).
*6,1402 (3002)	EKPRIME +1	C <sub>ekprime</sub> <sub>1</sub>	"TVCINIT1" pg. DPTV-16	LM-on high bandwidth cross-product steering gain factor.
*6,1403 (3003)	EKTLX/I	C <sub>ektlxdi</sub> <sub>0</sub>	"TVCINIT1" pg. DPTV-16	LM-off TVC DAP nominal gain.
*6,1404 (3004)	EKTLX/I +1	C <sub>ektlxdi</sub> <sub>1</sub>	"TVCINIT1" pg. DPTV-16	LM-on high bandwidth TVC DAP nominal gain.
*6,1405 (3005)	EKTLX/I +2	C <sub>ektlxdi</sub> <sub>2</sub>	"SWICHOVR" pg. DPTV-16	LM-on low bandwidth TVC DAP nominal gain.
*6,1406 (3006)	EREPPFRAC	C <sub>ereppfrac</sub> <sub>0</sub>	"TVCINIT1" pg. DPTV-17	LM-off TVC DAP repetitive c.g. correction fraction.
*6,1407 (3007)	EREPPFRAC +1	C <sub>ereppfrac</sub> <sub>1</sub>	"TVCINIT1" pg. DPTV-17	LM-on high b.w. TVC DAP repetitive c.g. correction fraction.
6,1410 (3010)	PACTOFF	PACTOFF	"TVCINIT1" pg. DPTV-24	TVC DAP pitch trim. Loaded by N48.
6,1411 (3011)	YACTOFF	YACTOFF	"TVCINIT1" pg. DPTV-27	TVC DAP yaw trim. Loaded by N48.
*6,1412 (3012)	HBN10	C <sub>hbn10</sub> <sub>0</sub>	"TVCINIT1" pg. DPTV-17	LM-on high bandwidth filter.
*6,1413 (3013)	HBN11/2	C <sub>hbn10</sub> <sub>1</sub>	"TVCINIT1" pg. DPTV-17	LM-on high bandwidth filter.
*6,1414 (3014)	HBN12	C <sub>hbn10</sub> <sub>2</sub>	"TVCINIT1" pg. DPTV-17	LM-on high bandwidth filter.
*6,1415 (3015)	HBD11/2	C <sub>hbn10</sub> <sub>3</sub>	"TVCINIT1" pg. DPTV-17	LM-on high bandwidth filter.
*6,1416 (3016)	HBD12	C <sub>hbn10</sub> <sub>4</sub>	"TVCINIT1" pg. DPTV-17	LM-on high bandwidth filter.
*6,1417 (3017)	HBN20	C <sub>hbn10</sub> <sub>5</sub>	"TVCINIT1" pg. DPTV-17	LM-on high bandwidth filter.



<u>Address</u> (Octal)	<u>Memory</u> <u>Notation</u>	<u>Document</u> <u>Notation</u>	<u>Reference</u>	<u>Comments</u>
*6,1420 (3020)	HBN21/2	C <sub>hbn10</sub> <sub>6</sub>	"TVCINIT1" pg. DPTV-17	LM-on high bandwidth filter.
*6,1421 (3021)	HBN22	C <sub>hbn10</sub> <sub>7</sub>	"TVCINIT1" pg. DPTV-17	LM-on high bandwidth filter.
*6,1422 (3022)	HBD21/2	C <sub>hbn10</sub> <sub>8</sub>	"TVCINIT1" pg. DPTV-17	LM-on high bandwidth filter.
*6,1423 (3023)	HBD22	C <sub>hbn10</sub> <sub>9</sub>	"TVCINIT1" pg. DPTV-17	LM-on high bandwidth filter.
*6,1424 (3024)	HBN30	C <sub>hbn10</sub> <sub>10</sub>	"TVCINIT1" pg. DPTV-17	LM-on high bandwidth filter.
*6,1425 (3025)	HBN31/2	C <sub>hbn10</sub> <sub>11</sub>	"TVCINIT1" pg. DPTV-17	LM-on high bandwidth filter.
*6,1426 (3026)	HBN32	C <sub>hbn10</sub> <sub>12</sub>	"TVCINIT1" pg. DPTV-17	LM-on high bandwidth filter.
*6,1427 (3027)	HBD31/2	C <sub>hbn10</sub> <sub>13</sub>	"TVCINIT1" pg. DPTV-17	LM-on high bandwidth filter.
*6,1430 (3030)	HBD32	C <sub>hbn10</sub> <sub>14</sub>	"TVCINIT1" pg. DPTV-17	LM-on high bandwidth filter.
*6,1431 (3031)	SLOPE2	C <sub>slope2</sub> <sub>0</sub>	"JLOOP" pg. DPRC-29	Undocked DAP phase plane parameter.
*6,1432 (3032)	SLOPE2 +1	C <sub>slope2</sub> <sub>1</sub>	"JLOOP" pg. DPRC-29	Docked DAP phase plane parameter.
*6,1433 (3033)	WLH/SLOP	C <sub>wlhdslp</sub> <sub>0</sub>	"JLOOP" pg. DPRC-29	Undocked DAP phase plane parameter.
*6,1434 (3034)	WLH/SLOP +1	C <sub>wlhdslp</sub> <sub>1</sub>	"JLOOP" pg. DPRC-29	Docked DAP phase plane parameter.
*6,1435 (3035)	WL-H/SLP	C <sub>wlmhds1p</sub> <sub>0</sub>	"JLOOP" pg. DPRC-29	Undocked DAP phase plane parameter.
*6,1436 (3036)	WL-H/SLP +1	C <sub>wlmhds1p</sub> <sub>1</sub>	"JLOOP" pg. DPRC-29	Docked DAP phase plane parameter.
*6,1437 (3037)	ECP	C <sub>ecp</sub> <sub>0</sub>	"NO62PT" pg. DPDS-19	Pitch rotation to trans- lation factor for docked DAP.

<u>Address</u> (Octal)	<u>Memory</u> <u>Notation</u>	<u>Document</u> <u>Notation</u>	<u>Reference</u>	<u>Comments</u>
*6,1440 (3040)	ECYW	C <sub>ecp1</sub>	"NO62PT" pg. DPDS-19	Yaw rotation to translation factor for docked DAP.
*6,1441 (3041)	ALPHAP	C <sub>alphap0</sub>	"TRCOMP" pg. DPDS-19	Pitch translation factor effect on roll for docked DAP.
*6,1442 (3042)	ALPHAYW	C <sub>alphap1</sub>	"TRCOMP" pg. DPDS-19	Yaw translation factor effect on roll for docked DAP.
*6,1443 (3043)	KMJDCKD	C <sub>kmjdckd</sub>	"REDAP" pg. DPRC-29	Average roll acceleration for docked DAP.
*6,1444 (3044)	KMJLDCKD	C <sub>kmjldckd</sub>	"REDAP" pg. DPRC-29	Average pitch acceleration for docked DAP.
*6,1445 (3045)	KMJ2DCKD	C <sub>kmj2dckd</sub>	"REDAP" pg. DPRC-29	Average yaw acceleration for docked DAP.
*6,1446 (3046)	J/MDCKD	C <sub>jdmcdckd</sub>	"REDAP" pg. DPRC-28	Reciprocal of average roll couple acceleration (docked DAP).
*6,1447 (3047)	J/MLDCKD	C <sub>jdmldckd</sub>	"REDAP" pg. DPRC-28	Reciprocal of average pitch couple acceleration (docked DAP).
*6,1450 (3050)	J/M2DCKD	C <sub>jdm2dckd</sub>	"REDAP" pg. DPRC-28	Reciprocal of average yaw couple acceleration (docked DAP).
*6,1455 (3055)	CLOKDELT	C <sub>clokdelt</sub>	"P4OS/F" pg. BURN-34	Phasing of "CLOKTASK" with respect to T <sub>ig</sub> .
*6,1470 (3070)	DAPDATR3	DAPDATR3	"P/YWCOMP" pg. DPDS-19	R1 of N87. Docked DAP control word.
*6,1471 (3071)	CH5FAIL	CH5FAIL	"COUPLE" pg. DPDS-19	R2 of N87. Docked DAP channel 5 jet inhibit mask.
*6,1472 (3072)	CH6FAIL	CH6FAIL	"FORCE" pg. DPDS-19	R3 of N87. Docked DAP channel 6 jet inhibit mask.
*6,1473 (3073)	DKRATE	DKRATE	"T5PHASE2" pg. DPRC-30	R1 of N89. Docked DAP maneuver rate.
*6,1474 (3074)	DKDB	DKDB	"REDAP" pg. DPRC-30	R2 of N89. Docked DAP deadband.
*6,1514 (3114)	DAPDATR1	DAPDATR1	"S41.2" pg. DPIR-13	Undocked DAP control word. It is loaded by R1 of N46.

<u>Address (Octal)</u>	<u>Memory Notation</u>	<u>Document Notation</u>	<u>Reference</u>	<u>Comments</u>
6,1515 (3115)	DAPDTR2	DAPDTR2	"S41.2" pg. DPIR-14	Undocked DAP control word. It is loaded by R2 of N46.
6,1521 (3121)	LEMMASS	LEMMASS	"FIXCW" pg. DPIR-22	LM mass. Loaded by R2 of N47.
6,1522 (3122)	CSMMASS	CSMMASS	"FIXCW" pg. DPIR-12	CSM mass. Loaded by R1 of N47, and updated (double precision) in "S40.8".
6,1712 (3312)	POLYNUM	C <sub>ponum</sub>	"ATERJOB" pg. BOOS-8	Booster pitch polynomial degree information.
6,1713-30 (3313-30)	POLYNUM +1	C <sub>pyc,i</sub>	"ATERJOB" pg. BOOS-8	Booster pitch polynomial (stored double precision, with lowest-powered term in 3313-4, etc.).
6,1731-2 (3331-2)	SATRLRT	SATRLRT	"ATERJOB" pg. BOOS-9	Booster roll rate (sign can be changed in "P11").
6,1733 (3333)	RPSTART	C <sub>rpstart</sub>	"ATERJOB" pg. BOOS-8	Time from CMC-deduced liftoff when roll and pitch drive of error needles should start.
6,1734 (3334)	POLYSTOP	C <sub>polystop</sub>	"ATERJOB" pg. BOOS-7	Value of complement of (stop time minus C <sub>rpstart</sub> ), where stop time is when performance of pitch polynomial evaluation should halt (quantity a negative number in memory).
*6,1741 (3341)	SATRATE	C <sub>satrt<sub>0</sub></sub>	"SATSTICK" pg. BOOS-8	Number of pulses (B12) loaded into error counter for no RHC deflection in Saturn DAP (e.g. 0).
*6,1742 (3342)	SATRATE +1	C <sub>satrt<sub>1</sub></sub>	"SATSTICK" pg. BOOS-8	Number of pulses loaded into error counter for positive RHC deflection in Saturn DAP.
*6,1743 (3343)	SATRATE +2	C <sub>satrt<sub>2</sub></sub>	"SATSTICK" pg. BOOS-8	Number of pulses loaded into error counter (with sign) for negative RHC deflection in Saturn DAP.
*6,1744 (3344)	SATRATE +3	C <sub>satrt<sub>3</sub></sub>	"SATSTICK" pg. BOOS-8	Number of pulses loaded into error counter for simultaneous positive and negative RHC input in Saturn DAP (e.g. 0).

<u>Address</u> (Octal)	<u>Memory</u> <u>Notation</u>	<u>Document</u> <u>Notation</u>	<u>Reference</u>	<u>Comments</u>
6,1751 (3351)	SATSCALE	C satscale	"NOPOLYM" pg. BOOS-8	Gain factor for automatic Saturn steering in P11.
*6,1777 (3377)	ALFAPAD	C alfapad	"P62.1" pg. ENTP-6	Nominal entry trim angle (expected to be negative number).
7,1400-1 (3400-1)	LAT(SPL)	LATSPL	"DELRSP" pg. DISP-25	Target latitude for R1 of N50 (R30 quantity in P00/P11). It is loaded by R1 of N61.
7,1402-3 (3402-3)	LNG(SPL)	LNGSPL	"DELRSP" pg. DISP-25	Target longitude for R1 of N50 (R30 quantity in P00/P11). It is loaded by R2 of N61.
*7,1424 (3424)	SBDELT	C sbdelt	"IGNITION" pg. BURN-34	Startup delay for "SBTASK".
7,1640-1 (3640-1)	ELEV	ELEV	"ELCALC" pg. REND-28	R2 of N55. Elevation angle for rendezvous.
7,1642-3 (3642-3)	TTPI	T tpi	"NC12F" pg. REND-36	N37. Ignition time for P35.

In the event that the memory must be initialized in flight, cell 1477 (GCOMPSW) should be checked for a non-negative value: it is unnecessary to set the cell as part of the prelaunch load, since the program sets it to 0 before entering P02 from P01.

The OWS state vector should be satisfactory for use in the periodic P00 integration, as well as for the integration done at the end of Average-G.

See Uplink Processing for other memory addresses which may be of interest for initialization.



## Attitude Maneuvers

R62DISP            Established by "CREWMANU" for a V49E

TS =  $0622_{vn}$

Proceed to "GOFLASH": if terminate, proceed to "ENDEXT"  
                                 if proceed, proceed  
                                 otherwise, proceed to previous line

Set bit 6(3AXISFLG) of FLAGWRD5 = 1

Perform "R6OCSM"

Proceed to "ENDEXT"

V89CALL            Established by "V89PERF" for a V89E. This is R63

Perform "R02BOTH"

Set bit 8(AZIMFLAG) of FIGWRD11 = 0

UTPIT = 0

UTYAW = 0

TS =  $0678_{vn}$

Perform "GOFLASHR": if terminate, proceed to "ENDEXT"  
                                 if proceed, proceed to "V89RECL"  
                                 otherwise, proceed to previous line

TS =  $100_2$  and perform "BLANKET"            (R3BLNK)

End of job

V89RECL

$R63TIME = T_{now} + K_{dplmin}$

Perform "R63"

TS =  $0618_{vn}$

Proceed to "GOFLASH": if terminate, proceed to "ENDEXT"  
                                 if proceed, proceed  
                                 otherwise, proceed to "V89RECL"

Set bit 6(3AXISFLG) of FLAGWRD5 = 0

Perform "R6OCSM"

Proceed to "ENDEXT"

R60CSM (Entered from "P4OSXTY" (P40,P41), "R61CSM", "R62DISP",  
and "V89RECL")

TEMPR60 = Return address

If bit 6(3AXISFLG) of FLAGWRD5 = 0:

Perform "VECPOINT"

THETAD = TS

Proceed to "TOBALL"

TOBALL

TS = 0618<sub>vn</sub>

Perform "GOPERF2R": if terminate, proceed to "R61TEST"  
if proceed, skip next 2 lines  
otherwise, proceed to "ENDMANU1"

Perform "CHKLINUS"

End of job

If bit 6(3AXISFLG) of FLAGWRD5 = 0:

Perform "VECPOINT"

THETAD = TS

TS = channel 31

If bits 14-13 of C31FLWRD  $\neq$  00<sub>2</sub>:

TS = C31FLWRD

If bits 15-13 (G&N Autopilot Control complement, Free Mode complement,  
and Hold Mode complement) of TS  $\neq$  011<sub>2</sub>: (11<sub>2</sub> means Automatic)

Proceed to "TOBALL"

TS = 0618<sub>vn</sub>

Perform "GODSPR"

Perform "CHKLINUS"

Proceed to "KALCMAN3" ("ENDMANU" establishes "ENDMANUV" for exit)

ENDMANU1

Proceed to address specified by TEMPR60

ENDMANUV (Established by "ENDMANU")

If bit 7(AUTOSEQ) of FLAGWRD10 = 1:

If bit 5(TRACKFLG) of FLAGWRD1 = 1:

Proceed to "ENDMANU1"

Proceed to "TOBALL"

CHKLINUS

If bit 12(PDSPFLAG) of FLAGWRD4 = 0:

Return

TBASE1 = (Return address -3 for last display, i.e. the TS setting)

Set restart group 1 to phase 7 (1.7, causing "RELINUS" to be established with priority  $10_8$  if a restart)

TS =  $100_8$  and perform "BLANKET" (PRIOREQ)

Return

RELINUS (Entered for restart group 1.7, due to "CHKLINUS")

If bit 5(TRACKFLG) of FLAGWRD1 = 0:

Proceed to "FIXDB"

Set bit 12(PDSPFLAG) of FLAGWRD4 = 1

If bit 9(UTFLAG) of FLAGWRD8 = 0:

Set bit 10(IMTRG) of FLAGWRD1 = 1

OPTIND = +0

Call "CTASK" in 0.01 second

Set bit 14(R21MARK) of FLAGWRD2 = 1

Change priority of present job to  $14_8$

Proceed to address specified by TBASE1 (regenerate last display due to setting of this cell in "CHKLINUS")

R61TEST

If MODREG = 0, proceed to "ENDMANU1"

If bit 12(PDSPFLAG) of FLAGWRD4 = 0, proceed to "GOTOPOOH"

Proceed to "TRACKTRM"

# VECPPOINT

$TS_3 = CDU$  (In push-down address 25D)

$TS = TS_3$

$X2 = "MIS"$

Perform "CDUTODCM"

$TS = \text{unit}(\text{POINTVSM } [MIS])$  (In push-down address 28D)

$TS_1 = TS * SCAXIS$

If  $(TS_1 - K_{\text{minmag}} < 0$  (or if  $\text{unit}TS_1$  overflows):

    If  $TS \cdot SCAXIS \geq 0$ :

$TS = TS_3$

    Return

$TS = \text{unit}(MIS_3 * \text{unit}X) * SCAXIS$

If  $|TS| - K_{\text{minmag}} < 0$  (or if  $\text{unit}TS$  overflows):

$TS = \text{unit}X$

$COF = \text{unit}TS$  (unit operation omitted if  $TS$  set to  $\text{unit}X$ )

$TS = \frac{1}{2}$  (i.e.  $180^\circ$ )

Proceed to "COMPMATX"

$COF = - \text{unit}TS_1$

$TS = \cos^{-1} (SCAXIS \cdot TS)$

Proceed to "COMPMATX"



### COMPMATX

Perform "DELCOMP"

X1 = "MIS"

X2 = "DEL"

Perform "MXM3" (leaves  $[TS] = [MFS]$ )

If  $|TS_3| - K_{\text{singim}} < 0$ , proceed to "FINDGIMB" ( $TS_3$  is 4th element of matrix  $[TS]$ ,  
If  $|SCAXIS_x| - K_{\text{sinvec1}} \geq 0$ , proceed to "FINDGIMB" (sine middle gimbal angle)

$[MIS] = [TS]$

$TS = (MIS_3 \text{ sgn } MIS_3) * SCAXIS$

$COF = - SCAXIS \text{ sgn } TS_x$

$TS = K_{\text{vecang1}}$

If  $|SCAXIS_x| - K_{\text{sinvec2}} < 0$ :

$TS = K_{\text{vecang2}}$

Perform "DELCOMP"

X1 = "MIS"

X2 = "DEL"

Perform "MXM3"

Proceed to "FINDGIMB"

### FINDGIMB

X1 = "TS"

Perform "DCMTOCDU"

$TS = TS$ , converted to twos complement single precision, B-1 revolutions

Return (to routine calling "VECPOINT")

KALCMAN3

(Entered from "STKTEST" and "TOBALL")

BCDU = CDU

X2 = "MIS"

TS = BCDU

Perform "CDUTODCM"

X2 = "MFS"

TS = THETAD

Perform "CDUTODCM"

X1 = "MIS"

Perform "TRANSPOS"

[TMIS] = [TS]

X1 = "TMIS"

X2 = "MFS"

Perform "MXM3"

[MFI] = [TS]

X1 = "MFI"

Perform "TRANSPOS"

[TMFI] = [TS]

COFSKEW = (TMFI<sub>5</sub> - MFI<sub>5</sub>, MFI<sub>2</sub> - TMFI<sub>2</sub>, TMFI<sub>1</sub> - MFI<sub>1</sub>)

CAM =  $\frac{1}{2}$  (MFI<sub>0</sub> + MFI<sub>8</sub> - 1 + MFI<sub>4</sub>)

AM =  $\cos^{-1}$  CAM

If AM - K<sub>mina</sub> < 0:

Inhibit interrupts

HOLDFLAG = -1

CDUXD = THETAD

Proceed to "NOGO"

If  $AM - K_{\max} < 0$ :

(Tag here "CHECKMAX")

$$COF = \text{unitCOFSKEW}$$

Proceed to "LOCSKIRT"

$$[MFISYM] = \frac{1}{2} ([MFI] + [TMFI])$$

TS = 1 - CAM, with magnitude limited  $< 2$

$$COF = \text{unit} \left( \sqrt{\frac{MFISYM_0 - CAM}{\frac{1}{2} TS}}, \sqrt{\frac{MFISYM_4 - CAM}{\frac{1}{2} TS}}, \sqrt{\frac{MFISYM_8 - CAM}{\frac{1}{2} TS}} \right)$$

If  $COF_x \geq COF_y$  and  $COF_x \geq COF_z$ : ( $COF_x$  largest component)

$$COF = COF \text{ sgn COFSKEW}_x \quad (\text{Tag here "METHOD1"})$$

$$COF_y = COF_y \text{ sgn MFISYM}_1$$

$$COF_z = COF_z \text{ sgn MFISYM}_2$$

Proceed to "LOCSKIRT"

If  $COF_y > COF_x$  and  $COF_y \geq COF_z$ : ( $COF_y$  largest component)

$$COF = COF \text{ sgn COFSKEW}_y \quad (\text{Tag here "METHOD2"})$$

$$COF_x = COF_x \text{ sgn MFISYM}_1$$

$$COF_z = COF_z \text{ sgn MFISYM}_5$$

Proceed to "LOCSKIRT"

If  $COF_z > COF_x$  and  $COF_z > COF_y$  (as it will): ( $COF_z$  largest)

$$COF = COF \text{ sgn COFSKEW}_z \quad (\text{Tag here "METHOD3"})$$

$$COF_x = COF_x \text{ sgn MFISYM}_2$$

$$COF_y = COF_y \text{ sgn MFISYM}_5$$

Proceed to "LOCSKIRT"

# TRANSPOS

$$[TS] = \begin{bmatrix} E_{X10} & E_{X13} & E_{X16} \\ E_{X11} & E_{X14} & E_{X17} \\ E_{X12} & E_{X15} & E_{X18} \end{bmatrix}$$

Return

### CDUTODCM

Convert  $TS$  to double precision ones complement angles  $TS_x$ ,  $TS_y$ , and  $TS_z$ . Matrix elements formed starting at address given by the contents of  $X2$ .

$$E_{X20} = \cos TS_y \cos TS_z$$

$$E_{X21} = \sin TS_x \sin TS_y - \cos TS_x \cos TS_y \sin TS_z$$

$$E_{X22} = \cos TS_x \sin TS_y + \sin TS_x \cos TS_y \sin TS_z$$

$$E_{X23} = \sin TS_z$$

$$E_{X24} = \cos TS_x \cos TS_z$$

$$E_{X25} = -\sin TS_x \cos TS_z$$

$$E_{X26} = -\sin TS_y \cos TS_z$$

$$E_{X27} = \sin TS_x \cos TS_y + \cos TS_x \sin TS_y \sin TS_z$$

$$E_{X28} = \cos TS_x \cos TS_y - \sin TS_x \sin TS_y \sin TS_z$$

Return

### DELCOMP

Entered with angle  $TS$ . Matrix elements stored starting at  $DEL$ . All matrix element have magnitudes limited  $<1$ .

$$TS_1 = 1 - \cos TS \quad (\text{magnitude limited } <2)$$

$$DEL_0 = COF_x^2 TS_1 + \cos TS$$

$$DEL_4 = COF_y^2 TS_1 + \cos TS$$

$$DEL_8 = COF_z^2 TS_1 + \cos TS$$

$$DEL_3 = COF_x COF_y TS_1 + COF_z \sin TS$$

$$DEL_1 = COF_x COF_y TS_1 - COF_z \sin TS$$

$$DEL_2 = COF_x COF_z TS_1 + COF_y \sin TS$$

$$DEL_6 = COF_x COF_z TS_1 - COF_y \sin TS$$

$$DEL_7 = COF_y COF_z TS_1 + COF_x \sin TS$$

$$DEL_5 = COF_y COF_z TS_1 - COF_x \sin TS$$

Return



### MXM3

$$TS_1 = [E_{X1}] (E_{X2_0}, E_{X2_3}, E_{X2_6})$$

$$TS_2 = [E_{X1}] (E_{X2_1}, E_{X2_4}, E_{X2_7})$$

$$TS_3 = [E_{X1}] (E_{X2_2}, E_{X2_5}, E_{X2_8})$$

$$[TS] = \begin{bmatrix} TS_{1x} & TS_{2x} & TS_{3x} \\ TS_{1y} & TS_{2y} & TS_{3y} \\ TS_{1z} & TS_{2z} & TS_{3z} \end{bmatrix}$$

Return

### DCMTOCDU

Entered with matrix starting address in X1. Cf. "CDUTODCM"  
for matrix elements.

$$TS_z = \sin^{-1} E_{X1_3}$$

$TS = \cos TS_z$ , with magnitude limited  $\leq 1$

$$TS_y = \sin^{-1} (-E_{X1_6} / TS)$$

If  $E_{X1_0} < 0$ :

$$TS_y = \frac{1}{2} \operatorname{sgn} TS_y - TS_y \quad (\text{the } \frac{1}{2} \text{ corresponds to } 180^\circ)$$

$$TS_x = \sin^{-1} (-E_{X1_5} / TS)$$

If  $E_{X1_4} < 0$ :

$$TS_x = \frac{1}{2} \operatorname{sgn} TS_x - TS_x$$

$TS = (TS_x, TS_y, TS_z)$  (double precision, 80 revolutions)

Return

### LOCSKIRT

If WHICHDAP = 0:

$TS = K_{arate} \text{RATEINDX}$  (Tag here "WCALC")

Perform "DELCOMP"

(If WHICHDAP = 0):

$$\text{BRATE} = [\text{QUADROT}] (K_{\text{arate}}^{\text{RATEINDX}} \text{COF})$$

$$T_{\text{tm}} = K_{\text{angtm}} \text{AM} / K_{\text{arate}}^{\text{RATEINDX}}$$

$$\text{BIASTEMP} = K_{\text{biasc}} \text{BRATE}$$

If WHICHDAP  $\neq$  0:

$$\text{TS} = K_{\text{tufitdk}} \text{DKRATE}$$

Perform "DELCOMP"

$$\text{TS}_1 = K_{\text{tufitdk}} \text{DKRATE}$$

$$\text{BRATE} = [\text{QUADROT}] \text{TS}_1 \text{COF}$$

$$T_{\text{tm}} = K_{\text{angtm}} \text{AM} / \text{TS}_1$$

$$\text{BIASTEMP} = C_{\text{ld2alpha}} (\text{BRATE}_0 | \text{BRATE}_0 | , \text{BRATE}_1 | \text{BRATE}_1 | , \text{BRATE}_2 | \text{BRATE}_2 | )$$

If bit 4(CYC61FLG) of FLAGWRD0 = 1: ("KALCMAN3" entered from "STKTEST")

Proceed to "RCYCLR61"

Set bit 2(CALCMAN2) of FLAGWRD2 = 1 (start of steering generation)

Proceed to "NEWANGL"

#### NOGO

Inhibit interrupts

Perform "STOPRATE"

If bit 2(R67FLAG) of FLAGWRD8 = 0:

Call "ENDMANU" in 0.02 second

End of job

Call "R67" in 1 second

Set restart group 1 to phase 11 (1.11, causing "PIKUP20" to be established with priority 10<sub>8</sub> if a restart)

End of job

NEWDELHI (Established by "UPDTCALL")

If bit 2(R67FLAG) of FLAGWRD8 = 1:

If bit 5(TRACKFLG) of FLAGWRD1 = 0:

Proceed to "FIXDB"

If HOLDFLAG  $\geq$  0:

Inhibit interrupts

CDUXD = CDU

Proceed to "NOGO"

Proceed to "NEWANGL"

NEWANGL (Entered from "LOCSKIRT", "NEWDELHI", and "R67RSTR")

X1 = "MIS"

X2 = "DEL"

Perform "MXM3"

[MIS] = [TS]

X1 = "MIS"

Perform "DCMTOCDU"

NCDU = TS, converted to twos complement single precision, B-1 rev.

If bit 2(CALCMAN2) of FLAGWRD2 = 1:

Set bit 2(CALCMAN2) of FLAGWRD2 = 0 (Tag here "MANUSTAT")

If bit 2(R67FLAG) of FLAGWRD8 = 1:

$T_{tm} = (2^{28} - 1)$  centi-seconds  $(37777_8 \ 37777_8)$

If bit 2(R67FLAG) of FLAGWRD8 = 0:

$T_{tm} = T_{tm} + T_{now} - 100$  centi-seconds

Inhibit interrupts (Tag here "TMDON")

HOLDFLAG = -1

(If bit 2(CALCMAN2) of FLAGWRD2 = 1):

If WHICHDAP = 1:

CH31TEMP = - 25<sub>8</sub>

Skip next 2 lines

If RATEINDX  $\geq$  6: (Note that this done for R67 also,  
regardless of the R67-specified rate)

Set bit 15 of RCSFLAGS = 1 (means high rate)

WBODY = BRATE

BIAS = BIASTEMP (least significant half only)

NEXTIME = TIME1 + 100, modulo 2<sup>14</sup> cs

DELCDU = QUADROT<sub>0</sub> (NCDU - BCDU) (ones complement difference formed)

CDUXD = BCDU<sub>sp</sub> (most significant half of CDUXD loaded; least  
significant half left alone)

BCDU = NCDU

Release interrupts

TS = T<sub>tm</sub> - T<sub>now</sub>

If TS  $>$  0:

Call "UPDTCALL" in (NEXTIME - TIME1) centi-seconds (if  
difference is negative, 16384 is added, i.e. 2<sup>14</sup>)

NEXTIME = NEXTIME + 100, modulo 2<sup>14</sup> cs

End of job

TS = TS + 101, limited  $\geq$  1 centi-second and  $\leq$  100 centi-seconds

Call "MANUSTOP" in TS centi-seconds

End of job

#### UPDTCALL

If bit 2(R67FLAG) of FLAGWRD8 = 1:

If R61CNTR = 0:

Establish "R67RSTRT" (priority 26<sub>8</sub>)

End of task



(If bit 2(R67FLAG) of FLAGWRD8 = 1):

R61CNTR = R61CNTR - 1

Establish "NEWDELHI" (priority 26<sub>8</sub>)

End of task

#### MANUSTOP

DELCDUY = 0

DELCDUZ = 0

WBODY<sub>1</sub> = 0

WBODY<sub>2</sub> = 0

BIAS<sub>1</sub> = 0

BIAS<sub>2</sub> = 0

CDUYD = THETAD<sub>y</sub> (least significant

CDUZD = THETAD<sub>z</sub> half of

CDUXD = THETAD<sub>x</sub> CDUXD left alone)

Perform "STOPRATE"

Proceed to "ENDMANU"

#### ENDMANU

Establish "ENDMANUV" (priority 14<sub>8</sub>)

End of task

#### STOPRATE

DELCDUX = 0

WBODY<sub>0</sub> = 0

BIAS<sub>0</sub> = 0

Set bit 15 of RCSFLAGS = 0 (means no longer have high rate)

```

If  HOLDFLAG < 0:
    HOLDFLAG = |HOLDFLAG| - 1
    THETADX = CDUXD-sp
    If  WHICHDAP = 1:
        CH31TEMP = - 258

DELCDUY = 0
DELCDUZ = 0
WBODY1 = 0
WBODY2 = 0
BIAS1 = 0
BIAS2 = 0
Return

```

#### FIXDB

```

Inhibit interrupts
If bit 12(MAXDBFLG) of FLAGWRD9 = 1:
    Perform "SETMAXDB"
If bit 12(MAXDBFLG) of FLAGWRD9 = 0:
    Perform "SETMINDB"
Set bit 2(R67FLAG) of FLAGWRD8 = 0
Release interrupts
Set restart group 1 to phase 11 (1.11, causing "PIKUP20" to be
    established with priority 108 if a restart)
Inhibit interrupts
Perform "STOPRATE"
End of job (where interrupts released)

```

R61CSM (Entered from "PIKUP20", "R52H", and "R66CSM")

GENRET = Return address

If R61CNTR > 0:

R61CNTR = R61CNTR - 1, limited > +0

Proceed to address specified by GENRET

If R61CNTR < 0:

Proceed to address specified by GENRET

Inhibit interrupts

ADB =  $K_{mindb}$

TS = |DBPTC| (absolute value converts input of e.g.  $350^{\circ}$  to  $\sim 10^{\circ}$ )

If TS  $\neq$  0:

ADB = TS

Release interrupts

Perform "CRS61.1"

Set bit 15(V50N18FL) of FLAGWRD3 = 0

If TS = 0: (no R60 performance)

R61CNTR = 3

Proceed to address specified by GENRET

Set bit 6(3AXISFLG) of FLAGWRD5 = 0 (Tag here "R61C1")

If bit 8(AZIMFLAG) of FLAGWRD11 = 1:

Set bit 6(3AXISFLG) of FLAGWRD5 = 1

R61CNTR = -1

Set bit 12(PDSPFLAG) of FLAGWRD4 = 1

Perform "R60CSM"

Set bit 12(PDSPFLAG) of FLAGWRD4 = 0

TS = 0

Perform "PRIODSP" (clears DSKY of V06N18 if in Minkey mode)

Set restart group 1 to phase 11 (1.11, causing "PIKUP20" to be established with priority  $10_8$  if a restart)

R61CNTR = 0

Proceed to address specified by GENRET

CRS61.1 (Entered only from "R61CSM")

Q611 = Return address

THETTEMP = THETADX

R63TIME =  $T_{\text{now}}$

Perform "R63" (enter here from "RCYCLR61" to refine maneuver by R60)

R63RANGE = R63TIME

Set bit 9(EXTRANGE) of FLGWRD10 = 1

If MODREG  $\neq$  27:

Set bit 3(Uplink Activity) of channel 11 = 0

If HOLDFLAG > 0: (includes e.g. in Free or SCS control)

Set bit 4(CYC61FLG) of FLAGWRD0 = 0

TS = 0

Proceed to address specified by Q611

If bit 8(AZIMFLAG) of FLGWRD11 = 1:

TS = THETTEMP - THETAD (ones complement difference formed)

If any( $|TS_i| - K_{\text{degree}10}$ )  $\geq$  0: (i = x,y,z)

Proceed to "STKTEST"

If bit 8(AZIMFLAG) of FLGWRD11 = 0:

CDUSPOT = (THETTEMP<sub>y</sub>, THETTEMP<sub>z</sub>, THETTEMP<sub>x</sub>) (Tag here "NOAZFLG1")

TS = SCAXIS

Perform "TRG\*NBSM"

If (TS · POINTVSM) -  $K_{\text{costen}}$  < 0:

Proceed to "STKTEST"



(If bit 8(AZIMFLAG) of FLGWRD11 = 0):

$$TS_1 = K_{\text{deltyme}} (TS * \text{POINTVSM})$$

If  $(|TS_1| - K_{\text{maxrate}}) \gg 0$ :

$$TS_1 = K_{\text{maxrate}} \text{unit} TS_1$$

$$DTHETASM = K_{\text{radto rev}} TS_1$$

TS = channel 31 (Tag here "AUTOCK")

If bits 14-13 of C31FLWRD  $\neq 00_2$ :

$$TS = C31FLWRD$$

If bits 15-13 (G&N Autopilot Control complement, Free Mode complement, and Hold Mode complement) of TS  $\neq 011_2$ : ( $11_2$  means Automatic)

Set bit 4(CYC61FLG) of FLAGWRD0 = 0

$$TS = 0$$

Proceed to address specified by Q611

If bit 14(STIKFLAG) of FLAGWRD1 = 1: (Tag here "DAPCK")

Set bit 4(CYC61FLG) of FLAGWRD0 = 0

$$TS = 0$$

Proceed to address specified by Q611

If bit 9(UTFLAG) of FLAGWRD8 = 1: (Tag here "STEP3CK")

If R63TIME = 0: (set 0 in "R63" computations, i.e. "UTAREAL",  
for body code  $\leq 46_8$ )

$$TS = 0 \quad (\text{MPAC})$$

Proceed to "CRS61.2A"

$TS = DCDU - \text{SAVEVEL}$  (Tag here "CRS61.2"; DCDU has OWS velocity,  
or 0 if UTFLAG = 1)

$$TS_1 = K_{\text{rvcsds}} (- \text{unit} TS * \text{SAVEPOS}) \quad (\text{B6})$$

$$TS = [\text{REFSMMAT}] (|TS|/R63TIME) TS_1 \quad (\text{R63TIME has } |LOS| \text{ value})$$

Proceed to "CRS61.2A"

#### CRS61.2A

If bit 8(AZIMFLAG) of FLGWRD11 = 0:

$$TS = TS + DTHETASM \quad (\text{BO})$$

$$TS_2 = TS \quad (TS_2 \text{ in } 20D)$$

$$DTHETASM = K_{tnth} TS_2$$

Perform "CDUTRIG"

Perform "SMCDURES"

$$TS_3 = DCDU, \text{ rescaled to B-1 revolutions (in 14D)}$$

Perform "CDUTRIG"

$$TS = TS_2$$

Perform "\*SMNB\*"

$$TS = K_{pt8} [MBDYTCTL] TS \quad (\text{B-3 rev/decisecond, in MPAC})$$

Inhibit interrupts

If bit 8(AZIMFLAG) of FLGWRD11 = 1:

$$TS_3 = TS_3 + K_{ld200} (THETAD - THETTEMP) \quad (\text{ones comp. difference formed})$$

$$CDUXD_{sp} = THETADX \quad (\text{Tag here "THTDXYZ"})$$

$$WBODY = TS \quad (\text{from MPAC})$$

$$DELCDU = TS_3 \quad (\text{from 14D})$$

If HOLDFLAG = 0:

If WHICHDAP = 1:

$$CH31TEMP = -25_8$$

$$HOLDFLAG = -1$$

Release interrupts

Set bit 4(CYC61FLG) of FLAGWRDO = 0

$$TS = 0$$

Proceed to address specified by Q611

STKTEST (Entered from "CRS61.1" if R60 maneuver appropriate)

$$TS = \text{bit 4(CYC61FLG) of FLAGWRDO}$$

Complement bit 4(CYC61FLG) of FLAGWRDO

If TS = 0:

Proceed to "KALCMAN3" (exits to "RCYCLR61" since CYC61FLG set, using check in "LOCKIRT")

If bit 15(V5ON18FL) of FLAGWRD3 = 1:

TS = 1

Proceed to address specified by Q611

Set bit 3(Uplink Activity) of channel 11 = 1

Set bit 4(CYC61FLG) of FLAGWRD0 = 0

TS = 0

Proceed to address specified by Q611

RCYCLR61 (Entered from "LOCSKIRT" if CYC61FLG set)

$R63TIME = T_{now} + K_{20p48sec} + T_{tm}$

Proceed to fourth line of "CRS61.1"

R63 (Entered from "CRS61.1" and "V89RECL")

Q6111 = Return address

$T_{decl} = R63TIME$

Perform "CSMCONIC"

$SAVEPOS = R_{att}$

$SAVEVEL = V_{att}$

$POINTVSM = (SAVEVEL * SAVEPOS)$  (B36, for use in "UTOPT45")

If bit 9(UTFLAG) of FLAGWRD8 = 1:

Proceed to "UTAREAL"

$T_{decl} = R63TIME$

Perform "LEMCONIC"

$DCDU = V_{att}$  (here is OWS velocity) (B7 m/cs)

$TS = R_{att} - SAVEPOS$

$SAVEPOS = unitTS$

$R63TIME = |TS|$  (here is LOS magnitude) (B29 meters)

$TS = SAVEPOS$

Proceed to "R63COM1"

# R63COM1

If bit 8(AZIMFLAG) of FLGWRD11 = 1:

Proceed to "UTOPT45" (P20 option 4/5, 3-axis solution)

POINTVSM = [REFSMMAT] TS

SCAXIS = (cos UTyaw cos UTPIT, sin UTyaw cos UTPIT, - sin UTPIT)

Perform "VECPOINT"

THETAD = TS

Proceed to address specified by Q6111

UTAREAL (Entered from "R63" if UTFLAG set, i.e. option 1/5 of P20)

TS = R63TIME

Perform "LSPOS" (leaves unit sun vector in TS<sub>1</sub>, 2D)

TS<sub>2</sub> = -unitSAVEPOS (in 8D, earth vector)

VELdC = K<sub>ldc</sub> SAVEVEL + (TS<sub>1</sub> \* K<sub>eclipol</sub>) (Tag here "SAVESTAR")

DCDU = 0 (velocity cell)

R63TIME = [SAVEPOS] (B29 meters)

SAVEPOS = - unitSAVEPOS

If UTSTARNO = 0:

SAVEVEL = 0

R63TIME = 0 (zero value used in "CRS61.1" to bypass LOS rate)

TS = unit ( unit(K<sub>ldsqrt3</sub> PLANVCUT) + VELdC )

Proceed to "R63COM1"

If (UTSTARNO - 46<sub>8</sub>) < 0: (catalog star)

SAVEVEL = 0

R63TIME = 0

TS = 6 UTSTARNO

TS = unit(K<sub>cat<sub>TS</sub></sub> + VELdC)

Proceed to "R63COM1"



If UTSTARNO = 46<sub>8</sub>: (sun)

R63TIME = 0

TS = unit(TS<sub>1</sub> + VELdC)

Proceed to "R63COM1"

If UTSTARNO ≥ 47<sub>8</sub>: (earth) (includes 77777<sub>8</sub> loading in "VLN7ODSP")

TS = unit(TS<sub>2</sub> + VELdC)

Proceed to "R63COM1"

UTOPT45 (Entered from "R63COM1" if AZIMFLAG = 1, i.e. option 4/5 of P20)

UTSB = TS (line-of-sight information)

UTSBP = (cos UTYAW cos UTPIT, sin UTYAW cos UTPIT, - sin UTPIT)

UTSAP = unit(UTSBP \* unitY) (Note this same as unit of (sin P, 0, cos Y cos P), so error if P = 0 and Y = 90°)

POINTVSM = unit(UTSB \* POINTVSM) (POINTVSM has v \* r from "R63")

UTSA = POINTVSM cos AZIMANGL + unit(UTSB \* POINTVSM) sin AZIMANGL

UTUYP = unit(UTSAP \* UTSBP) (cf. STV' in "AXISGEN")

UTUZP = UTSAP \* UTUYP (cf. STW' in "AXISGEN")

UTUY = unit(UTSA \* UTSB) (cf. STV in "AXISGEN")

UTUZ = UTSA \* UTUY (cf. STW in "AXISGEN")

UTX = UTSAP<sub>x</sub> UTSA + UTUYP<sub>x</sub> UTUY + UTUZP<sub>x</sub> UTUZ

UTY = UTSAP<sub>y</sub> UTSA + UTUYP<sub>y</sub> UTUY + UTUZP<sub>y</sub> UTUZ

UTZ = unit(UTSAP<sub>z</sub> UTSA + UTUYP<sub>z</sub> UTUY + UTUZP<sub>z</sub> UTUZ)

UTY = unitUTY

UTX = unitUTX

UTAMG = unit(UTX \* REFSMMAT<sub>3</sub>) (cf. TS in "CALCGA")

COSTH = UTAMG · UTZ

SINTH = UTAMG · UTY

Perform "ARCTRIG"

UTOGA = THETA

COSTH = (UTAMG \* UTX) \* REFSMMAT<sub>3</sub>

SINTH = REFSMMAT<sub>3</sub> \* UTX

Perform "ARCTRIG"

UTMGA = THETA

If |THETA| - K<sub>p166</sub> > 0:

Perform "ALARM" (pattern 401<sub>8</sub>)

COSTH = REFSMMAT<sub>6</sub> \* UTAMG (Tag here "UTCga1")

SINTH = REFSMMAT<sub>0</sub> \* UTAMG

Perform "ARCTRIG"

UTIGA = THETA

THETAD = (UTOGA, UTIGA, UTMGA), converted to twos complement B-1 rev, sp.

Proceed to address specified by Q6111

CALLR6X (Entered from "PIKUP20" if UTFLAG = 1, i.e. options 1, 2, 5 of P20)

Set bit 14(R21MARK) of FLAGWRD2 = 0

If OPTNTYPE = 0: (original option selection 1 or 5)

Proceed to "R66CSM"

TS = R67TIME - T<sub>now</sub>, with sign agreement forced (Tag "TYPE2", option 2)

If TS<sub>sp</sub> > 0: (TS B28 cs, so this means 163.84 sec or more)

Delay 163.83 seconds (by putting job to sleep via "DELAYJOB")

Proceed to 4th line of "CALLR6X"

If TS ≤ 0:

TS = 1

Call "R67" in TS centi-seconds

End of job

R66CSM (Entered from "CALLR6X")

If bit 5(TRACKFLG) of FLAGWRD1 = 0:

Proceed to "FIXDB"

Perform "R61CSM"

Delay 0.5 second (by putting job to sleep via "DELAYJOB")

Proceed to "R66CSM"

R67 (Called by "CALLR6X" and "NOGO")

Establish "R67START" (priority  $26_8$ )

Set bit 2(R67FLAG) of FLAGWRD8 = 1

End of task

R67START

TS = channel 31

If bits 14-13 of C31FLWRD  $\neq 00_2$ :

TS = C31FLWRD

If bits 15-13 (G&N Autopilot Control complement, Free Mode complement, and Hold Mode complement) of TS  $\neq 011_2$ : ( $11_2$  means Automatic)

Proceed to "NOGO"

If bit 14(STIKFLAG) of FLAGWRD1 = 1:

Inhibit interrupts

CDUXD = CDU

Proceed to "NOGO"

BCDU = CDU

COF = ( $\cos$  UTYAW  $\cos$  UTPIT,  $\sin$  UTYAW  $\cos$  UTPIT,  $-\sin$  UTPIT)

Inhibit interrupts

ADB =  $K_{mindb}$

TS = |DBPTC|

If TS  $\neq$  0:

ADB = TS

Release interrupts

TS =  $K_{\text{tufits}}$  RATEPTC

Perform "DELCOMP"

BRATE = [QUADROT]  $K_{\text{tufits}}$  RATEPTC COF

BIASTEMP = 0

Set restart group 1 to phase 13 (1.13, causing "R67RSTRT" to be established with priority 10<sub>g</sub> if a restart)

Proceed to "R67RSTRT"

R67RSTRT (Also entered from "UPDTCALL" establishing)

If bit 5(TRACKFLG) of FLAGWRD1 = 0:

Proceed to "FIXDB"

R61CNTR = 2048 (used in "UPDTCALL" to cause "R67RSTRT" to be established when = 0)

Change priority of present job to 26<sub>g</sub>

X2 = "MIS"

TS = BCDU

Set bit 2(R67FLAG) of FLAGWRD8 = 1

Perform "CDUTODCM"

Set bit 2(CALCMAN2) of FLAGWRD2 = 1

Proceed to "NEWANGL"

## Quantities in Computations

See also list of major variables and list of routines

ADB: See Digital Autopilot Interface Routines.

AM: Magnitude of required maneuver angle, scale factor B0, units revolutions.  
The  $\cos^{-1}$  routine gives an answer in the range  $0^\circ - 180^\circ$ .

AZIMANGL: See Orbital and Rendezvous Navigation. In documentation is "OMICRON".

BCDU: Set of three single precision cells, scale factor B-1, units revolutions, used to contain the previously desired values of the CDU angles.

BIAS (components  $BIAS_0$ ,  $BIAS_1$ ,  $BIAS_2$ ): See Digital Autopilot RCS Routines.

BIASTEMP: Temporary storage for the values of BIAS to be loaded into those cells in "NEWANGL", scale factor B13, units revolutions. Since only the least significant parts of BIASTEMP are used, this gives a loading for BIAS with the B-1 scaling required.

BRATE: Vector giving maneuver rates about the x, y, and z axes, scale factor B-3, units revolutions/decisecond (i.e. revolutions per 0.1 second). Could also be considered scaled B0 in "units" of  $450^\circ/\text{second}$ . The "axes" are "control" ones (see [QUADROT]).

$C_{ld2\alpha}$ : Erasable memory (double precision) constant, program notation "1/2ALPHA", scale factor B19, units of  $(\text{revolutions/decisecond}^2)^{-1}$ .  
Used in "LOCSKIRT" to compute BIASTEMP for docked (to OWS) maneuvers.

C31FLWRD: See Digital Autopilot RCS Routines.

CAM: Cosine of AM, scale factor B1.

CDUSPOT: See Coordinate Transformations.

CDUXD (components CDUXD, CDUYD, CDUZD): Values of desired CDU angles, scale factor B-1, units revolutions, used in RCS DAP if HOLDFLAG negative. See Digital Autopilot RCS Routines.

CH31TEMP: See Digital Autopilot RCS Routines.

COF: Unit vector, scale factor B1, giving the "eigenvector of rotation when rotating SCAXIS to the desired position." It is computed in "VECPPOINT" as the cross-product of the desired (POINTVSM) orientation in navigation base (i.e. spacecraft) coordinates and the SCAXIS complement, and is recomputed in "KALCMAN3".

COFSKEW: Vector, scale factor B2, in direction of COF (before maneuver magnitude checks, and equal to  $2 \sin AM$  times maneuver vector). Its unit (giving a scaling of B1) is employed if AM is between  $K_{\min}$  and  $K_{\max}$ .



COSTH: See Coordinate Transformations.

DBPTC: See Orbital and Rendezvous Navigation.

DCDU: See Coordinate Transformations. In addition to being an output of "SMCDURES", also used in "R63" to contain OWS velocity vector, scale factor B7, units meters/centi-second (this special meaning indicated by comment as necessary).

[DEL]: Matrix computed in "DELCOMP", whose elements have scale factor B0. The matrix is a transformation representing the effect of a rotation about the input vector COF by the angle TS (i.e. the matrix taking a vector in initial navigation base/spacecraft coordinates to the same vector in final navigation base coordinates).

DELCDU (DELCDUX, DELCDUY, DELCDUZ): Desired CDU change each 0.1 second, scale factor B-1, units revolutions (or revolutions/deci-second).

DKRATE: See Digital Autopilot RCS Routines.

DTHETASM: See Coordinate Transformations. Used in earlier part of "CRS61.2A" to contain a limited correction factor to null pointing error, scale factor B0, units revolutions/second.

GENRET: Single precision cell used to contain return address from "R61CSM".

HOLDFLAG: See Digital Autopilot Interface Routines.

K<sub>ld200</sub>: Constant, program notation "1/200", scale factor B0, single precision. Value is 0.005, corresponding to (1/200), nulling error in 20 seconds.

K<sub>ldc</sub>: See Inflight Alignment.

K<sub>ldsqrt3</sub>: Constant, program notation "1/SQRT3", scale factor B0, value 0.57735021. Value corresponds to  $1/\sqrt{3}$ .

K<sub>20p48sec</sub>: Constant, program notation "20.48SEC", scale factor B28, units centi-seconds. Value is  $1 \times 2^{-17}$ , corresponding to  $20.48 \times 100 \times 2^{-28}$ , where first term is value in seconds, second converts to centi-seconds, and third is scale factor.

K<sub>angtm</sub>: Constant, program notation "ANGLTIME", scale factor B19, used to rescale time from seconds to centi-seconds. Octal value is 00003<sub>8</sub> 04000<sub>8</sub>, corresponding to  $100 \times 2^{-19}$ .

K<sub>arate</sub>: Set of four constants, program notation "ARATE", defined only for I = 0, 2, 4, 6. They give the allowable maneuver rates, scale factor B-4, units revolutions/second. Values are 0.0022222222, 0.0088888889, 0.0222222222, and 0.0888888889: these correspond to rates of 0.05, 0.2, 0.5, and 2.0 degrees/second.

$K_{biasc}$ : Constant, program notation "BIASCALE", scale factor B16, value 0.0002543132. Value corresponds to  $(1/0.6) \times 10 \times 2^{-16}$ , where first term is reciprocal of "slope of decision line" (boundary of dead zone of plot of rate error vs. angle error has slope of  $0.6^\circ/\text{second}/^\circ$ ), second converts BRATE from deci-seconds to seconds, and third is scale factor.

$K_{cat}$ : See Inflight Alignment.

$K_{costen}$ : Constant, program notation "COSTEN", scale factor B2. Used double precision but stored single precision with nominal value of  $0.98481 \times 2^{-2}$ , where first term is  $\cos 10^\circ$  and second is scale factor. Double precision octal is 07702<sub>8</sub> 00034<sub>8</sub>, corresponding to about  $0.984864 \times 2^{-2}$ .

$K_{degree10}$ : Constant, program notation "DEGREE10", scale factor B-1, units revolutions. Used double precision but stored single precision with nominal value of 0.05556, corresponding to  $10^\circ$ . Double precision octal is 01616<sub>8</sub> 37651<sub>8</sub>, corresponding to about  $10.008^\circ$ .

$K_{deltyme}$ : Constant, program notation "DELTYME", scale factor B-2. Used double precision but stored single precision with nominal value of  $0.05 \times 2^2$ , where first term is desired nulling rate (i.e. zero error in 20 seconds, as for  $K_{ld200}$ ) and second is scale factor. Double precision octal is 06315<sub>8</sub> 01616<sub>8</sub>, corresponding to about  $0.050004 \times 2^2$ .

$K_{dplmin}$ : Constant, program notation "DPLMIN", scale factor B28, units centi-seconds. Value is  $60000 \times 2^{-28}$ , corresponding to 60 seconds or one minute.

$K_{eclipol}$ : See Inflight Alignment.

$K_{maxa}$ : Constant, program notation "MAXANG", scale factor B0, units revolutions. Used double precision but stored single precision with nominal value of 0.472222, corresponding to  $170^\circ$ . The octal is 17071<sub>8</sub> 00004<sub>8</sub>, giving about the same result.

$K_{maxrate}$ : Constant, program notation "MAXRATE", scale factor B0, units of radians/second. Value is 0.00174, corresponding approximately to  $0.09969 \times \pi/180$ , where first term is value in degrees/second and second converts to radians.

$K_{mina}$ : Constant, program notation "MINANG", scale factor B0, units revolutions. Used double precision but stored single precision with nominal value of 0.00069375, corresponding to  $0.25^\circ$ . Octal value is 00013<sub>8</sub> 17071<sub>8</sub>, corresponding to about  $0.252^\circ$ .

$K_{mindb}$ : See Digital Autopilot Interface Routines.

$K_{\text{minmag}}$ : Constant, program notation "DPB-14", scale factor B2, value  $2^{-14}$ , corresponding to a true value of  $2^{-12}$ . Since is used with unit vectors after forming cross product, corresponds to an angle between the vectors of about 0.244 mr. Equivalent function performed elsewhere in program without explicit use of constant.

$K_{\text{pl66}}$ : See Coordinate Transformations.

$K_{\text{pt8}}$ : Constant, program notation "POINT8", scale factor B-3, value 0.8. Value corresponds to  $0.1 \times 2^3$ , to convert between units of revolutions/second scaled B0 and revolutions/deci-second scaled B-3 (equivalent to multiplication by  $360^\circ/(45^\circ \times 10)$ ).

$K_{\text{radtorev}}$ : Constant, program notation "RADTOREV", scale factor B0, value 0.15915494, to convert between radians and revolutions. Value corresponds to  $(1/2\pi)$ .

$K_{\text{rvcsds}}$ : Constant, program notation "RVCS/RDS", scale factor B4, value  $15.915494 \times 2^{-4}$ , used to convert velocity from meters/centi-second to meters/second and angles from radians to revolutions. Value corresponds to  $100 \times (1/2\pi) \times 2^{-4}$ , where first term is time conversion, second converts radians, and third is scale factor.

$K_{\text{singim}}$ : Constant, program notation "SINGIMLC", scale factor B1, value 0.4285836003. True value is 0.8571672006, corresponding to  $\sin 59^\circ$  (approximately: the "2" should be "3" in the true value).

$K_{\text{sinvec1}}$ : Constant, program notation "SINVEC1", scale factor B1, value 0.3796356537. True value is 0.7592713074, corresponding to  $\sin 49.4^\circ$ . Program comments indicate this angle is used as a check that "are pointing the thrust axis".

$K_{\text{sinvec2}}$ : Constant, program notation "SINVEC2", scale factor B1, value 0.24621178. True value is 0.49242356, corresponding to  $\sin 29.5^\circ$ . Program comments indicate this angle is used as a check that "are pointing the AOT" (LM hardware), or otherwise "must be pointing the transponder or some vector in the Y or Z plane."

$K_{\text{tnth}}$ : Constant, program notation "TENTH", scale factor B-3, value  $0.1 \times 2^3$ , corresponding to a true value of 0.1 (to convert angle increments from revolutions/second to revolutions/deci-second).

$K_{\text{tufitdk}}$ : Constant, program notation "TUFITDK", scale factor B5, value 0.3125. Value corresponds to  $10 \times 2^{-5}$ , where first term converts from units of rate/deci-second to rate/second and second is scale factor.

$K_{\text{tufits}}$ : Constant, program notation "TUFITS", scale factor B-2, value 0.4. Value corresponds to  $0.1 \times 2^2$ , where first term converts the normal XX.XXX°/sec scaling of RATEPTC to X.XXXX°/sec and second is scale factor.

$K_{\text{vecang1}}$ : Constant, program notation "VECANG1", scale factor B0, units revolutions. Value is 0.1388888889, corresponding to  $50^\circ$ .

$K_{vecang2}$ : Constant, program notation "VECANG2", scale factor B0, units revolutions. Value is .09722222222, corresponding to 35°.

[MBDYTCTL]: Matrix of constants, elements with scale factor B1. True values are:

$$\begin{bmatrix} 1.0 & 0 & 0 \\ 0 & 0.99200495 & -0.12619897 \\ 0 & 0.12619897 & 0.99200495 \end{bmatrix}$$

Values are the same as those used in conjunction with [QUADROT].

[MFI]: Matrix with elements having scale factor B2, "relating final spacecraft axes to initial spacecraft axes."

[MFISYM]: Matrix with elements having scale factor B2, "the symmetric part of [MFI]", computed if AM exceeds  $K_{maxa}$ .

[MFS]: Matrix with elements having scale factor B1, "a direction cosine matrix relating the final spacecraft orientation to stable member axes." If  $\underline{A}$  is a vector in spacecraft coordinates, then the same vector in stable member axes ( $\underline{B}$ ) is given by  $\underline{B} = [\text{MFS}] \underline{A}$ .

[MIS]: Matrix with elements having scale factor B1, "a direction cosine matrix relating the initial spacecraft orientation to stable member axes."

NCDU: Value of next-desired CDU angles, single precision, scale factor B-1, units revolutions.

NEXTIME: Single precision value of time, scale factor B14, units centi-seconds, when "NEWDELHI" is to be entered next. Used to compensate for delays in performing "NEWDELHI" due to jobs of higher priority.

OPTIND: See Optics Computations.

OPTNTYPE: See Orbital and Rendezvous Navigation.

PLANVCUT: See Orbital and Rendezvous Navigation.

POINTVSM: Unit vector, scale factor B1, giving the direction in which SCAXIS is to be pointed. The components of this vector should be in stable member coordinates. Also used in R63 computations for utility storage purposes.

Q611: Single precision cell used to contain return address from "CRS61.1".

Q6111: Single precision cell used to contain return address from "R63".



[QUADROT]: Matrix of constants, elements with scale factor B0. Values are:

$$\begin{bmatrix} 0.1 & 0 & 0 \\ 0 & 0.09920 & -0.01262 \\ 0 & 0.01262 & 0.09920 \end{bmatrix}$$

Matrix serves to convert time units from rates/second to rates/decisecond (a factor of 0.1), and also "accounts for the roll displacement of the reaction jets with respect to navigation base coordinates" by rotating about X by  $-7.25^\circ$ .  $\sin 7.25^\circ = 0.1262$  and  $\cos 7.25^\circ = 0.9920$ . QUADROT<sub>0</sub> used to convert from rev/sec to rev/decisecond.

R61CNTR: See Orbital and Rendezvous Navigation. Also used in option 2 of P20 to control period of entrance to "R67RSTRT" (about once every 34 minutes) from "UPDTCALL" to restore [MIS] .

R63RANGE: See Measurement Incorporation.

R63TIME: Value of time for which R63 computations are carried out, scale factor B28, units centi-seconds. After being used for that purpose, loaded in R63 computations with magnitude of the line-of-sight vector, scale factor B29, units meters. Set zero if option 1/5 of P20 with star code  $\leq 46_8$  (i.e., not earth); otherwise has CSM vector.

R67TIME: See Orbital and Rendezvous Navigation.

RATEINDX: See Digital Autopilot Interface Routines.

RATEPTC: Single precision cell, scale factor B-2, units revolutions, loaded by R1 of N79 (for option 2 of P20). Since the desired display scale is  $X.XXXX^\circ/\text{sec}$  but the noun processing routine considers the input to be  $XX.XXX^\circ$ , another factor of 0.1 is accomplished by  $K_{\text{tufits}}$ . Due to the noun routine, the maximum value cannot exceed about  $8.9999^\circ/\text{sec}$ .

RCSFLAGS: See Digital Autopilot Interface Routines.

SAVEPOS: Cells used in "R63" initially to contain CSM position vector, scale factor B29, units meters. For P20 options 0/4 subsequently loaded with the unit line-of-sight vector, scale factor B1, between the vehicles; for options 1/5 loaded with the negative unit vector.

SAVEVEL: Cells used in "R63" initially to contain CSM velocity vector, scale factor B7, units meters/centi-second. Zeroed for options 1/5 of P20 if star code  $\leq 45_8$ .

SCAXIS: Unit vector, scale factor B1, giving the "axis to be pointed" along the direction indicated by POINTVSM. SCAXIS components should be in spacecraft (i.e. navigation base) coordinates. Also used in R63 computations for utility storage purposes.

SINTH: See Coordinate Transformations.



T<sub>tm</sub>: Time required to perform maneuver of magnitude AM (computed in XTU "LOCSKIRT"), scale factor B28, units centi-seconds. Subsequently changed to be the value of the computer clock when the maneuver should be completed (as part of initial-cycle computations). Set to +MAX to disable the exit if doing option 2 of P20.

TBASE1: Cell used for address storage purposes in "CHKLINUS" to permit restoration of program display after a restart (by "RELINUS"). Cell normally used to contain the waitlist restart information (see 3420.5-27), and is single precision.

TEMPR60: Single precision cell used to contain return address from "R60CSM".

THETADX: See Digital Autopilot RCS Routines.

THETTEMP: Single precision value of THETADX sampled at the start of "CRS61.1", used to retain a consistent sample for subsequent computations, scale factor B-1, units revolutions.

[TMFI]: Transpose of matrix [MFI], scale factor B2.

[TMIS]: Transpose of matrix [MIS], scale factor B1.

UTAMG: Unit vector, scale factor B1, used in the dedicated "CALOGA" routine included in "UTOPT45". Stored in push-down list location 6D.

UTIGA, UTMGA, UTOGA: Values of inner, middle, and outer gimbal angles, scale factor B0, loaded from THETA (see Coordinate Transformations), units revolutions. Stored in 26D, 28D, and 24D respectively.

UTPIT: See Orbital and Rendezvous Navigation. Sometimes called "rho".

UTSA: Information on desired orientation (based on AZIMANGL) about the LOS (UTSB) vector, scale factor B1, computed in "UTOPT45". Stored in 18D, with UTUX also used in program as notation.

UTSAP: Information on "primed coordinate system" orientation of UTSA, scale factor B1, computed in "UTOPT45". Stored in 0D, with UTUXP also used in program as notation.

UTSB: Information on desired pointing (LOS) vector, scale factor B1, loaded in "UTOPT45". Stored in 24D.

UTSBP: Information on "primed coordinate system" orientation of UTSB (i.e. spacecraft axis to be pointed in LOS direction), scale factor B1, loaded in "UTOPT45". Stored in 6D.

UTSTARNO: See Orbital and Rendezvous Navigation.

UTUY, UTUY<sub>P</sub>, UTUZ, UTUZ<sub>P</sub>: Vectors, scale factor B1, used in "UTOPT45" computations within the dedicated "AXISGEN" routine.

UTX, UTY, UTZ: Output of the dedicated "AXISGEN" routine in "UTOPT45", formed before "exit" as unit vectors, scale factor B1, for use in the dedicated "CALCGA" routine.

UTYAW: See Orbital and Rendezvous Navigation. Sometimes called "gamma".

VELdC: See Inflight Alignment.

WBODYY (components WBODY<sub>0</sub>, WBODY<sub>1</sub>, WBODY<sub>2</sub>): See Digital Autopilot RCS Routines.

WHICHDAP: See Digital Autopilot RCS Routines.

## Boost Computations

P11 (Entered from "CHKCOMED" after liftoff deduced)

Set  $T_{\text{evt}} = T_{\text{now}}$  and  $T_{\text{now}} = 0$  (with special provisions for restarts)

$T_{\text{eph}} = T_{\text{eph}} + T_{\text{evt}}$  (special restart provisions)

Inhibit interrupts

Perform "PREREAD1"

Release interrupts

Call "ATERTASK" in 0.5 second

TS = 11 and perform "NEWMODEX"

Set bit 12(NODOP01) of FLAGWRD1 = 1

DNLSTCOD = 3

Perform "CLEANDSP" (gets rid of backup liftoff, V75)

$\text{ERCOMP} = \text{ERCOMP} + \left[ X_{\text{sm}} \right] \text{THETAN}$

TS =  $T_{\text{evt}}$

Perform "EARTHRR\*" (starting at second line)

(OGC, IGC, MGC) = CDU, rescaled to scale factor 80 revolutions, double precision

ALT =  $C_{\text{pgncsat}}$

LONG =  $C_{\text{pdong}}$

LAT =  $C_{\text{atd}}$

Set bit 13(ERADCOMP) of FLAGWRD1 = 0 (means pad radius altitude base)

TS = 0

Perform "LALOTORV"

$R_{\text{nl}} = \text{ALPHA}$

$TS_2 = 0$  (time argument)

$ALPHAV = \text{unit}(\cos \text{LONG} \cos \text{LAT}, \sin \text{LONG} \cos \text{LAT}, \sin \text{LAT})$

Perform "LALOTORV" (starting at 3rd line)

$Z_r = - \text{unit}ALPHAV$

$V_{nl} = K_{merthrt} (R_{nl} * C_{unitw})$

$TS_1 = \text{unit}(Z_r * C_{unitw})$  (east)

$TS_2 = \text{unit}(TS_1 * Z_r)$  (- south)

$X_r = \text{unit}(TS_1 \sin \text{LAUNHAZ} + TS_2 \cos \text{LAUNHAZ})$

$Y_r = - \text{unit}(X_r * Z_r)$

$$\begin{bmatrix} X_r \\ Y_r \\ Z_r \end{bmatrix} \quad \text{(cells also used for temporary storage during computation)}$$

$SATRLRT = SATRLRT \operatorname{sgn}(\frac{1}{2} - \text{LAUNHAZ} + C_{azmth})$

Set bit 13(REFSMFLG) of FLAGWRD3 = 1

AVEGEXIT = "VHHDOT"

$ldPIPADT = K_{2secp}$

$TS = 200 - \text{TIME1}$ , limited  $\gg 1$

Call "READACCS" in TS centi-seconds (i.e. 2 seconds after liftoff deduced)

Proceed to "NORMLIZE"

#### VHHDOT

$VMAGI = |V|$

$HDOT = \text{unit}R \cdot V$

$ALTI = |R| - K_{rpad}$

$TS = 0662_{vn}$

Proceed to "REGODSP"





$$\underline{Y}_{dc} = \text{unit}(-\sin \text{ANGROLL} \cos \text{NMPH}, -\cos \text{ANGROLL}, -\sin \text{ANGROLL} \sin \text{NMPH})$$

$$\underline{Z}_{dc} = \text{unit}(-\cos \text{ANGROLL} \cos \text{NMPH}, \sin \text{ANGROLL}, -\cos \text{ANGROLL} \sin \text{NMPH})$$

Perform "CALCGTA"

Proceed to "NOPOLYM"

#### NOPOLYM

Perform "CDUTRIG"

$\underline{TS}_1 = (\text{IGC}, \text{MGC}, \text{OGC})$ , converted to single precision twos comp., B-1 rev.

$\underline{TS}_2 = \text{CDUSPOT}$ , converted to single precision twos comp., B-1 rev.

$\underline{TS} = \underline{TS}_1 - \underline{TS}_2$  (ones comp. difference formed) (y,z,x for 0,1,2)

$$\underline{AK}_0 = \frac{1}{4} (\underline{TS}_2 + \underline{TS}_0 \text{ SINCDUZ})$$

$$\underline{AK}_1 = \underline{TS}_0 \text{ COSCDUX COSCDUZ} + \underline{TS}_1 \text{ SINCDUX}$$

$$\underline{AK}_2 = -\underline{TS}_0 \text{ SINCDUX COSCDUZ} + \underline{TS}_1 \text{ COSCDUX}$$

If  $\text{SATSW} > 0$ : (no takeover in effect)

Perform "NEEDLER"

Delay 0.25 seconds (by putting job to sleep via "DELAYJOB")

Proceed to "ATERJOB" (program comments indicate loop rate of about 0.56 second)

If  $\text{SATSW} = 0$ : (first takeover cycle)

$$\underline{\text{BIASAK}} = -\underline{\text{AK}}$$

$$\text{SATSW} = -1$$

$$\underline{\text{AK}} = \underline{\text{BIASAK}} + \underline{\text{AK}} \quad (\text{tag here "STEERSAT"})$$

$$\underline{\text{AK}} = C_{\text{satscale}} \underline{\text{AK}}$$

Perform "NEEDLER"

Delay 0.25 seconds (by putting job to sleep via "DELAYJOB")

Proceed to "ATERJOB"

ATRESET            Entered from "ATERJOB" for a change in bit 10 of channel 30

Set bit 3 of RCSFLAGS = 1            (causes "NEEDLER" re-initialization)

Perform "NEEDLER"

Delay 0.06 seconds (by putting job to sleep via "DELAYJOB")

TS = SATSW

If TS > 0:            (control just given)

Set bit 9(S4B Takeover Enable) of channel 12 = 1

SATSW = 0            (first takeover cycle, causing BIASAK loading)

If TS ≤ 0:            (control just removed)

SATSW = 1

Set bit 9(S4B Takeover Enable) of channel 12 = 0

Perform "NEEDLER"

Delay 0.25 seconds (by putting job to sleep via "DELAYJOB")

Proceed to "ATERJOB"

SATSTKON            Entered from "STABLISH"

T5LOC = "REDOSAT"

Set TIME5 to cause program interrupt #2 in 0.01 seconds

Set bits 15-14 (DAPBIT1, DAPBIT2) of FLAGWRD6 = 11<sub>2</sub>

Perform "ZEROJET"

Proceed to "PINBRNCH"

REDOSAT

Set bit 3 of RCSFLAGS = 1

Perform "NEEDLER"

Set bit 9 (S4B Takeover Enable) of channel 12 = 1 (Reset e.g. by  
"STABLISH")

T5LOC = "SATSTICK"

Set TIME5 to cause program interrupt #2 in 0.10 seconds

Resume

# SATSTICK

T5LOC = "SATSTICK"

Set TIME5 to cause program interrupt #2 in 0.10 seconds

TS = bits 6-1 of (- channel 31)

Perform "STICKCHK"

$AK_0 = C_{\text{satrt}_{\text{RMANNDX}}}$

$AK_1 = C_{\text{satrt}_{\text{PMANNDX}}}$

$AK_2 = C_{\text{satrt}_{\text{YMANNDX}}}$

Perform "NEEDLER"

Resume

## Quantities in Computations

See also list of major variables and list of routines

ldPIPADT: See IMU Computations.

$AK$  ( $AK_0$ ,  $AK_1$ ,  $AK_2$ ): See Digital Autopilot Interface Routines.

ALPHAV: See Coordinate Transformations.

ALT: See Coordinate Transformations.

ALTI: Value of vehicle altitude above  $K_{rpad}$ , scale factor B29, units meters. It is displayed in R3 of N62.

ANGROLL: Value of roll angle remaining to be performed, scale factor B0, units revolutions.

AVEGEXIT: See General Program Control.

BIASAK: Value of the complement of  $AK$ , same scaling, saved on the pass through "ATERJOB" following the one which senses a logic 0 to 1 transition (binary 1 to 0) of bit 10 of channel 30. It effectively permits the errors to be "nulled" at this point, and is applied until Saturn control is removed. If get a restart, or if switch subsequently set to CMC position, a new set of BIASAK values is determined.

$C_{atd}$ : See Prelaunch Alignment.

$C_{azmth}$ : See Prelaunch Alignment.

$C_{pdong}$ : Erasable memory constant, program notation "PADLONG", scale factor B0, units revolutions, giving longitude of pad.

$C_{pgncsat}$ : Erasable memory constant, program notation "PGNCSALT", scale factor B29, units meters, giving "altitude of IMU" (measured from  $K_{erad}$ , see Coordinate Transformations). The magnitude of the initial-condition position vector is the sum of  $K_{erad}$  and  $C_{pgncsat}$ .

$C_{polystop}$ : Single precision erasable memory constant, program notation "POLYSTOP", scale factor B14, units centi-seconds, giving minus (time from liftoff minus  $C_{rpstart}$ ) when performance of pitch polynomial should be stopped. The pitch polynomial will be halted, regardless of the numbers in the erasable memory constants, when time from liftoff reaches 163.84 seconds.

$C_{ponum}$ : Single precision erasable memory constant, program notation  
 "POLYNUM", scale factor B14, giving "polynomial degree -1" of the polynomial used to compute POLYANS. Erasable memory cell allocation is such that  $C_{ponum}$  should not exceed 5 (giving a final term of  $C_{pyc,6}^{TS_1^6}$ ).

$C_{pyc,i}$  ( $i = 0 - C_{ponum} + 1$ ): Erasable memory coefficients used to compute POLYANS, with highest order coefficient stored in ("POLYLOC" +3), next highest in ("POLYLOC" +1), etc. Scale factor of  $C_{pyc,0}$  is B5 revolutions; scale factor of  $C_{pyc,1}$  is B-9 revolutions/centi-second; scale factor of  $i$ th is  $B(5 - 14i)$ , with units of revolutions/(centi-second)<sup>1</sup>. For  $C_{ponum} = 5$ , the first term,  $C_{pyc,0}$ , is in ("POLYNUM" +1).

$C_{rpstart}$ : Single precision erasable memory constant, program notation  
 "RPSTART", scale factor B14, units centi-seconds, giving the time (from CMC-deduced liftoff) when the evaluation of the roll and pitch computations should be started for generation of attitude error displays (as opposed to driving the displays from the difference between sampled attitude and present attitude).

$C_{satrt_i}$  ( $i = 0 - 3$ ): Set of single precision erasable memory constants, program notation ( $i = 0$ ) "SATRATE", scale factor B12, units output pulses. Cells used in "SATSTICK" to specify the number of pulses generated by "NEEDLER" routine for deflections of the RHC in roll, pitch, and yaw (a positive input via  $\underline{AK}$  gives a negative output). Values for  $i = 0$  and  $i = 3$  should be zero; for  $i = 1$ , the value for a positive maneuver is supplied; and for  $i = 2$ , the value for a negative maneuver is supplied. A saturated error counter would be produced for a setting of  $384 \times 4 = 1536$  (the factor of 4 is due to the value of  $K_{dacsc}$  in "NEEDLES") least increments.

$C_{satscale}$ : Single precision erasable memory constant, program notation  
 "SATSCALE", scale factor B2, used as the gain on the  $\underline{AK}$  outputs for Saturn steering based on polynomial output. A value of 1.00 for the constant would cause a saturated error counter for  $67.50^\circ$  of roll error and  $16.875^\circ$  of pitch/yaw error (the saturated error counter corresponds to 384 pulses). The "1.00" value is effectively the one employed to drive the needles if no Saturn steering is done. For RHC control,  $C_{satrt}$  is used to specify the output information.

$C_{unitw}$ : See General Program Control.

CDUSPOT: See Coordinate Transformations.

COSCDUX, COSCDUY, COSCDUZ: See Coordinate Transformations.

DNLSTCOD: See Telemetry.



ERCOMP: See Prelaunch Alignment.

HDOT: Value of altitude rate for display purposes, scale factor B7, units meters/centi-second.

IGC: See Coordinate Transformations. Used to retain sampled value of  $CDU_y$  near deduced liftoff for attitude error display during the first  $C_{rpstart}$  centi-seconds after liftoff deduced.

$K_{2secp}$ : See IMU Computations.

$K_{merthrt}$ : Constant, program notation "--ERTHRAT", scale factor B-18, units radians/centi-second. Value is  $-7.292115138E-7 \times 2^{18}$ , corresponding to an earth rotation period of about 86164.099 seconds. The octal value gives a decimal equivalent of about  $-7.2921150717E-7$ .

$K_{rpad}$ : See Burn Control.

LAT, LONG: See Coordinate Transformations.

LAUNCHAZ: See Prelaunch Alignment.

MGC: See Coordinate Transformations. Used to retain sampled value of  $CDU_z$  near liftoff (see IGC).

NMPH: Value of nominal pitch history, scale factor B0, units revolutions, obtained from POLYANS.

OGC: See Coordinate Transformations. Used to retain sampled value of  $CDU_x$  near liftoff (see IGC).

PMANNDX: See Digital Autopilot Interface Routines.

POLYANS: Output of polynomial used for determination of nominal pitch history, scale factor B5, units revolutions.

$R_{n1}$ : See General Program Control.

RCSFLAGS: See Digital Autopilot Interface Routines.

RMANNDX: See Digital Autopilot Interface Routines.

SATRLRT: Quantity used to specify nominal Saturn roll rate, scale factor B0, units revolutions/centi-second. It is set as part of the pre-launch erasable load, and subsequently has sign changed in the latter part of "P11" depending on the deduced polarity of the required Saturn roll. Roll rate should not exceed about 2.2 degrees/second (i.e.,  $2^{-14}$  rev/centi-second), or malfunction of the check in "ATERJOB" for zeroing of ANGROLL could result. If it does not, check there will set ANGROLL to 0 when the value reaches about  $(0.05/n)$  degrees beyond the nominal roll amount, where "n" is the Saturn roll rate in degrees/second.

SATSW: Single precision cell, scale factor B14, used to control the logic associated with automatic steering of Saturn (based on polynomial outputs). An initial condition of +1 is set in "ATERTASK" (entered also if a restart), meaning display only. A value of +0 is set in "ATRESET" just after control takeover (i.e., present value of SATSW = 1 and of channel 30 bit 10 = 0). In "NOPOLYM", the value of 0 causes BIASAK to be initialized, and the value of SATSW set to -1 (indicating a subsequent pass through Saturn automatic steering). When the value of channel 30 bit 10 is sensed as 1 (a logic 0, meaning no control), and SATSW is  $\leq 0$ , then "ATRESET" entered again, this time causing a reset of SATSW to +1, the configuration for attitude error display only.

SINCDUX, SINCDUY, SINCDUZ: See Coordinate Transformations.

$T_{eph}$ : Triple precision value of ephemeris time, program notation "TEPHEM", scale factor B42, units centi-seconds. Cell must be initialized as part of prelaunch erasable memory load to the elapsed time between ephemeris origin and the pre-launch origin setting for the computer clock. When liftoff is deduced,  $T_{evt}$  is added to  $T_{eph}$ .  $T_{eph}$  is changed by V70.

T5LOC: See Digital Autopilot Interface Routines.

THETAN: See Prelaunch Alignment.

TIME5: See Digital Autopilot Interface Routines.

$V_{n1}$ : See General Program Control.

VMAGI: Value of magnitude of vehicle inertial velocity, scale factor B7, units meters/centi-second. It is displayed by several nouns.

$\underline{X}_r$ : Unit vector, scale factor B1, forming first row of [REFSMMAT] computed after liftoff deduced, and defined as "horizontal vector at launch azimuth".

$\underline{Y}_r$ : Unit vector, scale factor B1, forming second row of [REFSMMAT], and formed as  $\underline{Z}_r * \underline{X}_r$ .

YMANNDX: See Digital Autopilot Interface Routines.

$\underline{Z}_r$ : Unit vector, scale factor B1, forming third row of [REFSMMAT], and defined as "local vertical".

# Burn Control

P30

Set bit 7(UPDATFLG) of FLAGWRD1 = 1

Set bit 5(TRACKFLG) of FLAGWRD1 = 1

TS = 0633<sub>vn</sub>

Proceed to "VNFLASH": (if terminate, proceed to "GOTOPOOH")  
if proceed, proceed  
otherwise, proceed to previous line

TS = 0681<sub>vn</sub>

Proceed to "VNFLASH": (if terminate, proceed to "GOTOPOOH")  
if proceed, proceed  
otherwise, proceed to previous line

Set bit 8(XDELVFLG) of FLAGWRD2 = 1

Set bit 7(UPDATFLG) of FLAGWRD1 = 0

T<sub>decl</sub> = T<sub>ig</sub>

Perform "CSMPREC"

V<sub>tig</sub> = V<sub>att</sub>

R<sub>tig</sub> = R<sub>att</sub>

R<sub>act3</sub> = R<sub>att</sub>

UNRM = unit(R<sub>act3</sub> \* V<sub>tig</sub>)

TS<sub>2</sub> = - UNRM

TS<sub>3</sub> = - unitR<sub>act3</sub>

TS<sub>1</sub> = TS<sub>3</sub> \* UNRM

DELVSIN = DELVLC  $\begin{bmatrix} TS_1 \\ TS_2 \\ TS_3 \end{bmatrix}$  (DELVLC notation also "DELVSLV")

VGDISP = |DELVSIN|

$$VVEC = V_{tig} + DELVSIN$$

$$RVEC = R_{tig}$$

$$XXXALT = K_{rpad}$$

Perform "APSIDES"

$$HPER = (TS_{rp} - XXXALT), \text{ limited } \leq K_{maxnm}$$

$$HAPO = (TS_{ra} - XXXALT), \text{ limited } \leq K_{maxnm}$$

$$TS = 0642_{vn} \quad (\text{Tag here "PARAM30"})$$

Proceed to "VNFLASH": (if terminate, proceed to "GOTOPOOH")  
                           if proceed, proceed  
                           otherwise, proceed to previous line

If bit 13(REFSMFLG) of FLAGWRD3 = 1:

$$TS = DELVSIN$$

Perform "GET+MGA"

If bit 13(REFSMFLG) of FLAGWRD3 = 0:

$$pMGA = - K_{marsdp}$$

Set bit 11(TIMRFLAG) of FLAGWRD7 = 1

$$NVWORD1 = +0$$

Call "CLOKTASK" in 0.01 seconds

$$TS = 1645_{vn}$$

Proceed to "VNFLASH": (if terminate, proceed to "GOTOPOOH")  
                           if proceed, proceed  
                           otherwise, proceed to previous line

Set bit 11(TIMRFLAG) of FLAGWRD7 = 0

Proceed to "GOTOPOOH"

P31

Set bits 7(UPDATFLG) and 5(TRACKFLG) of FLAGWRD1 = 1

Set bit 1(PCFLAG) of FLAGWRD10 = 0

Set bit 5(NC12FLG) of FLAGWRD0 = 0

TS = 0695<sub>vn</sub>

Proceed to "VNFLASH": (if terminate, proceed to "GOTOPOOH")  
if proceed, proceed  
otherwise, proceed to previous line

TS = 0657<sub>vn</sub>

Proceed to "VNFLASH": (if terminate, proceed to "GOTOPOOH")  
if proceed, proceed  
otherwise, proceed to previous line

TS = 0637<sub>vn</sub> (Tag here "GETTPI")

Proceed to "VNFLASH": (if terminate, proceed to "GOTOPOOH")  
if proceed, proceed  
otherwise, proceed to previous line

TS = NC1TIG

If bit 5(NC12FLG) of FLAGWRD0 = 1: (should not be)

TS = NC2TIG

Set bit 6(FINALFLG) of FLAGWRD2 = 0

TIMEC = TS

T<sub>ig</sub> = TS

NC = HALFREVS (modulo 32 revolutions)

Perform "VN1645"

Proceed to "NC12ST" (exits to "NC12DH")

NC12DH Entered from "NC12OUT"

If bit 6(FINALFLG) of FLAGWRD2 = 0:

Set bit 7(UPDATFLG) of FLAGWRD1 = 1

TS = 0684<sub>vn</sub>

Proceed to "GOFLASH": if terminate, proceed to "GOTOPOOH"  
if proceed, proceed  
otherwise, proceed to previous line

If bit 5(NC12FLG) of FLAGWRD0 = 1: (i.e. P32 rather than P31)

DHNCC = DHDSP

Proceed to "NC12H"



## NC12H

TS = 0681<sub>vn</sub>

Proceed to "VNFLASH": (if terminate, proceed to "GOTOPOOH")  
if proceed, proceed  
otherwise, proceed to previous line

TS<sub>2</sub> = - UP<sub>1</sub>

TS<sub>3</sub> = - unitR<sub>act2</sub>

TS<sub>1</sub> = TS<sub>3</sub> \* UP<sub>1</sub>

Set bit 8(XDELVFLG) of FLAGWRD2 = 1

DELVSIN = DELVLC  $\begin{bmatrix} TS_1 \\ TS_2 \\ TS_3 \end{bmatrix}$

DELVEET<sub>3</sub> = DELVSIN (same cell)

Perform "VN1645"

Proceed to "NC12ST"

DISPALM Entered from "PHSMCH" (600<sub>g</sub>), "EHCOMP" (601<sub>g</sub>), and "GETRJ" (603<sub>g</sub>)

If bit 6(FINALFLG) of FLAGWRD2 = 0:

Set bit 7(UPDATFLG) of FLAGWRD1 = 1

Perform "VARALARM" (TS set before enter "DISPALM")

TS = 0509<sub>vn</sub>

Proceed to "GOFLASH": if terminate, proceed to "GOTOPOOH"  
if proceed, skip next 3 lines  
otherwise, proceed

If MODREG = 33, proceed to "P33"

If MODREG = 32, proceed to "P32"

Proceed to "P31"

DELVLC = 0 (here for PRO to alarm)

If MODREG = 33:

Set bit 8(XDELVFLG) of FLAGWRD2 = 0

Proceed to "N81DISP"

Proceed to "NC12H"

OLPALRM

Entered for alarm 602<sub>g</sub> from "NC12G"

TS = 602<sub>g</sub>

Perform "VARALARM"

TS = 0509<sub>vn</sub>

Proceed to "GOFLASH": if terminate, proceed to "GOTOPOOH"  
if proceed, proceed to "NC12OUT"  
otherwise, proceed

If MODREG = 33, proceed to "P33"

If MODREG = 32, proceed to "P32"

Proceed to "P31"

P32

Set bits 7(UPDATFLG) and 5(TRACKFLG) of FLAGWRD1 = 1

Set bit 1(PCFLAG) of FLAGWRD10 = 0

Set bit 5(NC12FLG) of FLAGWRD0 = 1

TS = 0628<sub>vn</sub>

Proceed to "VNFLASH": (if terminate, proceed to "GOTOPOOH")  
if proceed, proceed  
otherwise, proceed to previous line

TS = 0657<sub>vn</sub>

Perform "VNFLASHR": (if terminate, proceed to "GOTOPOOH")  
if proceed, skip next 2 lines  
otherwise, proceed to previous line

TS = 001<sub>2</sub> and perform "BLANKET" (R1BLINK)

End of job

TS = 0637<sub>vn</sub> (Tag here "GETTPI")

Proceed to "VNFLASH": (if terminate, proceed to "GOTOPOOH")  
if proceed, proceed  
otherwise, proceed to previous line

TS = NC1TIG

If bit 5(NC12FLG) of FLAGWRD0 = 1: (should be)

TS = NC2TIG

Set bit 6(FINALFLG) of FLAGWRD2 = 0

TIMEC = TS

$T_{ig} = TS$

NC = HALFREVS (modulo 32 revolutions) (written over in "NC12ST")

Perform "VN1645"

Proceed to "NC12ST" (exits to "NC12DH", page BURN-3)

### P33

Set bits 7(UPDATFLG) and 5(TRACKFLG) of FLAGWRD1 = 1

Set bit 1(PCFLAG) of FLAGWRD10 = 0

$TS = 0611_{vn}$

Proceed to "VNFLASH": (if terminate, proceed to "GOTOPOOH")  
if proceed, proceed  
otherwise, proceed to previous line

$T_{cdh} = T_{csi} + C_{tcs}$  ( $T_{cdh}$  notation also TNSR;  $T_{csi}$  also TNCC)

$TS = 0613_{vn}$

Proceed to "VNFLASH": (if terminate, proceed to "GOTOPOOH")  
if proceed, proceed  
otherwise, proceed to previous line

$TS = 0637_{vn}$

Proceed to "VNFLASH": (if terminate, proceed to "GOTOPOOH")  
if proceed, proceed  
otherwise, proceed to previous line

Set bit 6(FINALFLG) of FLAGWRD2 = 0

$T_{ig} = T_{csi}$

Proceed to "DISP45A" (enters "VN1645", then computations; exits to "NCCDMP7+17")

### NCCDMP7+17

If bit 6(FINALFLG) of FLAGWRD2 = 0:

Set bit 7(UPDATFLG) of FLAGWRD1 = 1

$TS = 0682_{vn}$

Proceed to "VNFLASH": (if terminate, proceed to "GOTOPOOH")  
 if proceed, proceed  
 otherwise, proceed to previous line

$\underline{TS} = \underline{DELVEET3}$

$\underline{V}_{init} = \underline{V}_{act3}$

$\underline{R}_{init} = \underline{R}_{pass3}$  ( $\underline{R}_{pass3}$  has active vehicle vector from "DISP45A")

Perform "GET.LVC"

Proceed to "N81DISP"

#### N81DISP

$TS = 0681_{vn}$

Proceed to "VNFLASH": (if terminate, proceed to "GOTOPOOH")  
 if proceed, proceed  
 otherwise, proceed to previous line

Set bit 5(REVFLAG) of FLAGWRD2 = 1 (causes "GET.LVC" to derive matrix only)

Perform "GET.LVC"

$\underline{DELVSIN} = \underline{DELVLVC} \begin{bmatrix} \underline{TS}_1 \\ \underline{TS}_2 \\ \underline{TS}_3 \end{bmatrix}$

$\underline{DELVEET3} = \underline{DELVSIN}$  (same cell)

$\underline{TS} = \underline{V}_{init} + \underline{DELVSIN}$

Perform "INTSTALL"

$\underline{VCV} = \underline{TS}$

$\underline{RCV} = \underline{R}_{init}$

$\underline{T}_{et} = \underline{T}_{csi}$  ( $\underline{T}_{csi}$  notation also TNCC)

Set bit 4(CONICINT) of FLAGWRD3 = 1

$\underline{T}_{pass4} = \underline{T}_{cdh}$  ( $\underline{T}_{cdh}$  notation also TNSR)

$\underline{T}_{decl} = \underline{T}_{pass4}$

Perform "INTEGRVS"

$\underline{R}_{targ} = \underline{R}_{att}$

Proceed to "DISP45A"

P34

Set bits 7(UPDATFLG) and 5(TRACKFLG) of FLAGWRD1 = 1

Set bit 1(PCFLAG) of FLAGWRD10 = 0

TS = 0613<sub>vn</sub>

Proceed to "VNFLASH": (if terminate, proceed to "GOTOPOOH")  
if proceed, proceed  
otherwise, proceed to previous line

Set bit 6(FINALFLG) of FLAGWRD2 = 0

$T_{tpi0} = T_{tpi}$

$T_{ig} = T_{cdh}$  ( $T_{cdh}$  notation also TNSR)

Perform "VN1645"

Proceed to "P33/P73B"

P33/P73B (part of P34)

TS =  $T_{ig}$

Perform "ADVANCE" (page BURN-11)

Perform "CDHMVR"

DELVEET<sub>1</sub> = DELVEET<sub>2</sub>

Perform "INTSTALL"

Set bit 4(CONICINT) of FLAGWRD3 = 0

Set bit 4(CONICINT) of FLAGWRD3 = 1 (superseding previous line)

$T_{decl} = T_{tpi}$

$T_{et} = T_{cdh}$

$RCV = R_{act2}$

$VCV = V_{act3}$

Perform "INTEGRVS"

$R_{act3} = R_{att}$

$V_{act3} = V_{att}$



Perform "INTSTALL"

Set bit 4(CONICINT) of FLAGWRD3 = 0

Set bit 4(CONICINT) of FLAGWRD3 = 1 (superseding previous line)

$T_{decl} = T_{tpi}$

$T_{et} = T_{cdh}$

$R_{CV} = R_{pass2}$

$V_{CV} = V_{pass2}$

Perform "INTEGRVS"

$R_{pass3} = R_{att}$

$V_{pass3} = V_{att}$

Set bit 14(ITSWITCH) of FLAGWRD7 = 1

NOMTPI = 0

Perform "S33/34.1"

If TS  $\neq$  0: (non-zero for error return from "S33/34.1")

Perform "ALARM" (pattern 0611<sub>g</sub>)

TS = 0509<sub>vn</sub>

Proceed to "GOFLASH": if terminate, proceed to "GOTOPOOH"  
if proceed, proceed  
otherwise, proceed to "P34"

NOMTPI = 0

If bit 6(FINALFLG) of FLAGWRD2 = 0: (Tag here "P33/P73C")

Set bit 7(UPDATFLG) of FLAGWRD1 = 1

$T_{tpi} = \text{NOMTPI} + T_{tpi}$

$TS = T_{tpi} - T_{cdh}$  ( $T_{cdh}$  notation also TNSR)

$TS = TS - K_{60min}$

If TS  $\geq$  0, proceed to previous line

$T1TOT2 = TS + K_{60min}$  (note that if negative, no modulo action)

$$TS = T_{tpi} - T_{tpi0}$$

$$TS = |TS| - K_{60minh}$$

If  $TS \gg 0$ , proceed to previous line

$$T2TOT3 = (TS + K_{60minh}) \operatorname{sgn}(T_{tpi} - T_{tpi0})$$

$$TS = 0675_{vn}$$

Proceed to "VNFLASH": (if terminate, proceed to "GOTOPOOH")  
if proceed, proceed  
otherwise, proceed to previous line

$$TS_2 = -UP_1 \quad (\text{Tag here "SKIP75"})$$

$$TS_3 = -\operatorname{unit} R_{act2}$$

$$TS_1 = TS_3 * UP_1$$

$$DELVLVC = \begin{bmatrix} TS_1 \\ TS_2 \\ TS_3 \end{bmatrix} \quad DELVEET_1$$

$$(YCSM, YDOTC, YDOTL) = - (AUTOY, CMYDOT, LMYDOT)$$

$$DELVLVC_y = CMYDOT$$

If bit 1(PCFLAG) of FLGWRD10 = 1: (should be 0)

$$\text{If } |DELVLVC_y| - K_{ld10fps} < 0:$$

$$DELVLVC = 0$$

$$TS = 0681_{vn}$$

Proceed to "VNFLASH": (if terminate, proceed to "GOTOPOOH")  
if proceed, proceed  
otherwise, proceed to previous line

$$DELVSIN = DELVLVC \begin{bmatrix} TS_1 \\ TS_2 \\ TS_3 \end{bmatrix} \quad (\text{matrix recomputed for restart reasons, but elements should be same})$$

$$DELVEET_2 = DELVSIN \quad (\text{same cell})$$

Perform "VN1645"

Proceed to "P33/P73B"

# ADVANCE

$T_{decl} = TS$  (TS set to required time before enter)

Perform "PRECSET" (which loads  $UP_1$ )

Set bit 8(XDELVFLG) of FLAGWRD2 = 1

$V_{pass2} = V_{pass3}$

$R_{pass2} = R_{pass3}$

$R_{tig} = R_{act3}$

$R_{act2} = |R_{tig}| \text{ unit } (R_{tig} - (R_{tig} \cdot UP_1) UP_1)$

$V_{tig} = V_{act3}$

$V_{act2} = |V_{tig}| \text{ unit } (V_{tig} - (V_{tig} \cdot UP_1) UP_1)$

Return

## P35

Set bits 7(UPDATFLG) and 5(TRACKFLG) of FLAGWRD1 = 1

Set bit 1(PCFLAG) of FLAGWRD10 = 0

Set bit 3(TPIMNFLG) of FLAGWRD10 = 0

$NN1 = 0$

$TS = 0637_{vn}$

Proceed to "VNFLASH": (if terminate, proceed to "GOTOPOOH")  
if proceed, proceed  
otherwise, proceed to previous line

$TS = 0655_{vn}$

Proceed to "VNFLASH": (if terminate, proceed to "GOTOPOOH")  
if proceed, proceed  
otherwise, proceed to previous line

Set bit 7(HAVEELEV) of FLAGWRD2 = 0

$T_{tpi0} = T_{tpi}$

$T_{ig} = T_{tpi}$

If ELEV  $\neq$  0:

Set bit 7(HAVEELEV) of FLAGWRD2 = 1

Set bit 6(FINALFLG) of FLAGWRD2 = 0

Perform "VN1645"

Proceed to "P34/P74C"

P34/P74C (part of P35)

Set bit 14(ITSWITCH) of FLAGWRD7 = 1 (means iterate "S33/34.1" below)

If bit 7(HAVEELEV) of FLAGWRD2 = 0:

Set bit 14(ITSWITCH) of FLAGWRD7 = 0

NOMTPI = 0

$T_{decl} = T_{tpi} + \text{NOMTPI}$  (Tag here "INTLOOP", entered from below)

Perform "PRECSET"

Perform "S33/34.1"

If TS  $\neq$  0: (non-zero for error return from "S33/34.1")

Perform "ALARM" (pattern 0611<sub>g</sub>)

TS = 0509<sub>vn</sub>

Proceed to "GOFLASH": if terminate, proceed to "GOTOPOOH"  
if proceed, proceed  
otherwise, proceed to 2nd previous line  
(regenerate alarm again)

Proceed to "P35"

If bit 14(ITSWITCH) of FLAGWRD7 = 1: (Tag here "SWCHCLR")

Set bit 14(ITSWITCH) of FLAGWRD7 = 0

Proceed to 5th line of "P34/P74C" ( $T_{decl}$  loading)

If bit 7(HAVEELEV) of FLAGWRD2 = 0:

TS = 0655<sub>vn</sub>

Proceed to "VNFLASH": (if terminate, proceed to "GOTOPOOH")  
if proceed, proceed  
otherwise, proceed to previous line

If bit 7(HAVEELEV) of FLAGWRD2 = 1:

$V_{tprime_x} = T_{tpi}$  (used for temporary storage) (Tag here "P34/P74D")

$TS = 0637_{vn}$

Proceed to "VNFLASH": (if terminate, proceed to "GOTOPOOH")  
if proceed, proceed  
otherwise, proceed to previous line

If bit 6(FINALFLG) of FLAGWRD2 = 1:

If bit 7(AUTOSEQ) of FLGWRD10 = 1:

If  $|V_{tprime_x} - T_{tpi}| \neq 0$ : (i.e. new N37 value)

Set bit 7(HAVEELEV) of FLAGWRD2 = 0

ELEV = 0

$T_{ig} = T_{tpi}$

Proceed to "P34/P74C"

$CSTH = \cos K_{centang}$  (Tag here "P34/74E")

$SNTH = \sin K_{centang}$

$RVEC = R_{pass3}$

Set bit 9(RVSW) of FLAGWRD7 = 1 (means new  $R$ ,  $V$  not desired)

$VVEC = V_{pass3}$

Perform "TIMETHET" (leaves time in T)

$INTIME = T_{tpi}$

$T_{pass4} = T_{tpi} + T$

Perform "S34/35.2"

$DELVTPI = |DELVEET_2|$

$DELVTPF = |V_{pass4} - V_{tprime}|$

$VVEC = V_{iprime}$

$RVEC = R_{act3}$



XXXALT =  $K_{\text{rpad}}$

Perform "APSIDES"

POSTTPI =  $(TS_{\text{rp}} - \text{XXXALT})$

$T_{\text{ig}} = T_{\text{tpi}}$

$TS = T_{\text{tpi}} - T_{\text{tpi0}}$

$TS = |TS| - K_{60\text{minh}}$

If  $TS \gg 0$ , proceed to previous line

$T2TOT3 = (TS + K_{60\text{minh}}) \text{sgn}(T_{\text{tpi}} - T_{\text{tpi0}})$

$TS = 0658_{\text{vn}}$

Proceed to "VNFLASH": (if terminate, proceed to "GOTOPOOH")  
if proceed, proceed  
otherwise, proceed to previous line

If bit 6(FINALFIG) of FLAGWRD2 = 0:

Set bit 7(UPDATFIG) of FLAGWRD1 = 1

If bit 6(FINALFIG) of FLAGWRD2 = 1:

Set bit 3(TPIMNFIG) of FLAGWRD10 = 1

$V_{\text{tprime}} = \text{DEVLVC}$  (temporary storage)

$TS = 0681_{\text{vn}}$

Proceed to "VNFLASH": (if terminate, proceed to "GOTOPOOH")  
if proceed, proceed  
otherwise, proceed to previous line

If  $|\text{DEVLVC} - V_{\text{tprime}}| \neq 0$ : (at least one component change by  
 $\gg 2^{-14} \text{ m/cs} \approx 0.02 \text{ fps}$ )

$TS_2 = - \text{UNRM}$

$TS_3 = - \text{unitR}_{\text{act3}}$

$TS_1 = TS_3 * \text{UNRM}$

$\text{DELVEET3} = \text{DEVLVC}$

$\begin{bmatrix} TS_1 \\ TS_2 \\ TS_3 \end{bmatrix}$

Perform "S34/35.3"

Perform "VN1645" (Tag here "NOWRT35")

Proceed to "P34/P74C"

VN1645

SUBEXIT = Return address

pMGA = K<sub>dpm01</sub>

If bit 6(FINALFLG) of FLAGWRD2 = 1:

pMGA = 2 K<sub>dpm01</sub>

If bit 13(REFSMFLG) of FLAGWRD3 = 1:

TS = DELVEET3 (same cell as DELVSIN)

Perform "GET+MGA"

Set bit 11(TIMRFLAG) of FLAGWRD7 = 1

NVWORD1 = +0

Call "CLOKTASK" in 0.01 seconds

Delay 1 second (by putting job to sleep via "DELAYJOB")

TS = 1645<sub>vn</sub>

Proceed to "GOFLASH": if terminate, proceed to "GOTOPOOH"  
if proceed, proceed  
otherwise, proceed to "CLUPDATE"

If bit 6(FINALFLG) of FLAGWRD2 = 0: (Tag here "N45PROC")

Set bit 6(FINALFLG) of FLAGWRD2 = 1

Proceed to "CLUPDATE"

NOMTIG = T<sub>ig</sub> (Tag here "N45ENAJ")

If bit 1(PCFLAG) of FLGWRD10 = 0:

Set bit 5(MANEUFLG) of FLGWRD10 = 1

If bit 7(AUTOSEQ) of FLGWRD10 = 1:

WRENDPOS = K<sub>posvel2</sub>

WRENDVEL = K<sub>posvel3</sub>

Proceed to second line of "GOTOPOOH"

### CLUPDATE

Set bit 11(TIMRFLAG) of FLAGWRD7 = 0

CADRFSLH+2 = "VNRSTART" + 3

Set restart group 4 to phase 1 (4.1, causing "INITDSP" to enter "VNRSTART" if restart)

Set bit 7(UPDATFLG) of FLAGWRD1 = 0

Proceed to address specified by SUBEXIT

### VNRSTART

Change priority of present job to  $07_8$  (allow "PIKUP20" and "REDOR22")

Change priority of present job to  $13_8$

Set bit 7(UPDATFLG) of FLAGWRD1 = 0

Proceed to address specified by SUBEXIT

### P36

Set bits 7(UPDATFLG) and 5(TRACKFLG) of FLAGWRD1 = 1

Set bit 1(PCFLAG) of FLAGWRD10 = 0

Set bit 3(TPIMNFLG) of FLAGWRD10 = 1

Set bit 6(FINALFLG) of FLAGWRD2 = 0

Perform "VN1645"

Proceed to "P35/P75B"

### P35/P75B (part of P36)

$T_{\text{strt}} = T_{\text{now}}$

$T_{\text{ig}} = T_{\text{strt}} + K_{\text{atiginc}}$

$\text{INTIME} = T_{\text{ig}}$

$T_{\text{decl}} = T_{\text{ig}}$

Perform "PRECSET"

$\text{ULOS} = \text{unit}(\underline{R}_{\text{pass3}} - \underline{R}_{\text{act3}})$

$\text{UNRM} = \text{unit}(\underline{R}_{\text{act3}} * \underline{V}_{\text{act3}})$

Perform "S34/35.2"

If bit 6(FINALFLG) of FLAGWRD2 = 0:

Set bit 7(UPDATFLG) of FLAGWRD1 = 1

If bit 6(FINALFLG) of FLAGWRD2 = 1:

Set bit 8(P35FLAG) of FLGWRD10 = 1

$V_{tprime} = DVLOS$  (temporary storage)

TS = 0659<sub>vn</sub>

Proceed to "VNFLASH": (if terminate, proceed to "GOTOPOOH")  
if proceed, proceed  
otherwise, proceed to previous line

If  $|DVLOS - V_{tprime}| \neq 0$ : (at least one component change by  
 $\geq 2^{-14}$  m/cs  $\approx$  0.02 fps)

$TS_1 = ULOS$

$TS_3 = - \text{unit}(ULOS * UNRM)$

$TS_2 = TS_3 * ULOS$

$DELVEET3 = DVLOS \begin{bmatrix} TS_1 \\ TS_2 \\ TS_3 \end{bmatrix}$

Perform "S34/35.3"

Perform "VN1645" (Tag here "NOWRT36")

Proceed to "P35/P75B"

### P37

Set bits 7(UPDATFLG) and 5(TRACKFLG) of FLAGWRD1 = 1

Set bit 1(PCFLAG) of FLGWRD10 = 0

UTPIT = 0

UTYAW = 0

EXTVBACT = 0002<sub>4g</sub> (sets bits 5 and 3 to 1)

Set bit 4(R31FLAG) of FLAGWRD9 = 1

Establish "R31CALL" (priority 05<sub>g</sub>)

End of job

P38

Set bits 7(UPDATFLG) and 5(TRACKFLG) of FLAGWRD1 = 1

Set bit 1(PCFLAG) of FLAGWRD10 = 0

Set bit 1(PCFLAG) of FLAGWRD10 = 1 (superseding previous line)

Set bit 6(FINALFLG) of FLAGWRD2 = 0

LASTTIG =  $T_{ig}$

TS = 0639<sub>vn</sub>

Proceed to "VNFLASH": (if terminate, proceed to "GOTOPOOH")  
if proceed, proceed  
otherwise, proceed to previous line

Set bit 8(XDELVFLG) of FLAGWRD2 = 1

$T_{decl}$  = LASTTIG

Perform "CSMPREC"

Set bit 9(RVSW) of FLAGWRD7 = 1 (means new R, V not desired)

CSTH = 0

SNTH =  $K_{cs359p}$  (i.e. angle of about 90°)

$RVEC = R_{att}$

$VVEC = V_{att}$

NOTE that UPDATFLG not reset 0, so garbled results  
if get an update computation since RVEC cells  
also used for "INCORP2" restart protection  
(program tag "HOLDW").

Perform "TIMETHET"

$T_{ig}$  = LASTTIG + T

TS = 0633<sub>vn</sub>

Proceed to "VNFLASH": (if terminate, proceed to "GOTOPOOH")  
if proceed, proceed  
otherwise, proceed to previous line

Perform "VN1645"

Proceed to "P38RECYC"



### P38RECYC

$\text{DELVLVC} = 0$

$T_{\text{decl}} = T_{\text{ig}}$

Perform "PRECSET"

If bit 6(FINALFLG) of FLAGWRD2 = 0:

Set bit 7(UPDATFLG) of FLAGWRD1 = 1

$V_{\text{tig}} = V_{\text{act3}}$

$R_{\text{tig}} = R_{\text{act3}}$

$(YCSM, YDOTC, YDOTL) = - (AUTOY, CMYDOT, LMYDOT)$

$\text{DELVLVC}_y = \text{CMYDOT}$

If bit 1(PCFLAG) of FLGWRD10 = 1: (should be)

If  $|\text{DELVLVC}_y| - K_{\text{ldlOfps}} < 0$ :

$\text{DELVLVC} = 0$  (may give 1301<sub>g</sub> alarm at N45 display)

$TS = 0681_{\text{vn}}$

Proceed to "VNFLASH": (if terminate, proceed to "GOTOPOOH")  
if proceed, proceed  
otherwise, proceed to previous line

$TS_2 = - \text{UNRM}$

$TS_3 = - \text{unit} R_{\text{act3}}$

$TS_1 = TS_3 * \text{UNRM}$

$\text{DELVSIN} = \text{DELVLVC} \begin{bmatrix} TS_1 \\ TS_2 \\ TS_3 \end{bmatrix}$

$\text{DELVEET3} = \text{DELVSIN}$  (same cell)

Perform "VN1645"

Proceed to "P38RECYC"

#### P4OCSM

Set bit 11(RCSBURN) of FLAGWRD1 = 0

TS = 1

Set bit 3(SBFLAG) of FLAGWRD2 = 1

If bit 8(XDELVFLG) of FLAGWRD2 = 1:

TS = 0

CSTEER = TS

CAPF =  $K_{feng}$

Proceed to "P4OS/F"

P4OS/F            Entered from "P4OCSM" and "P4LCSM"

NOMTIG =  $T_{ig}$

TS =  $T_{ig} - T_{now}$

Set bit 11(TIMRFLAG) of FLAGWRD7 = 1

NVWORD1 = +0

$TS_1 = 100$  (TS/100, fractional part)            (i.e. TS modulo 100 cs)

$TS_2 = TS_1 + C_{clokdelt}$

Call "CLOKTASK" in TS centi-seconds

Perform "RO2BOTH"

Proceed to "S4O.1"            (which exits to "S4O.2,3", in turn to "P4OSXTY")

#### P4OSXTY

Set bit 4(PFRATFLG) of FLAGWRD2 = 1

Delay 1 second (by putting job to sleep via "DELAYJOB")            (allows  
"FIXDB" performance)

Inhibit interrupts

Perform "SETMINDB"

Set bit 6(3AXISFLG) of FLAGWRD5 = 0

Release interrupts

Perform "R6OCSM"

NBRCYCLS = -1 (initialization for "UPDATEVG" for Lambert burn)

If bit 11(RCSBURN) of FLAGWRD1 = 1:

$TS = V_{gtig}$  (Tag here "P41/DSP")

Perform "S41.1"

$V_{gbody} = TS$  (also updated by "DYNDISP" in "CLOKTASK" logic)

AVEGEXIT = "CALCN85"

NVWORD1 = 0685<sub>vn</sub>

Proceed to "P4OS/SV"

VGDISP =  $|V_{gtig}|$

DVTOTAL = 0

AVEGEXIT = "S40.8"

TS = 0204<sub>g</sub>

Proceed to "GOPERF1": if terminate, proceed to "POST41"  
if proceed, MRKRTMP = +1 and proceed (do test)  
otherwise, MRKRTMP = -1 and proceed

CNTR = +0 (used as flag in "S40.6")

Call "S40.6" in 0.01 second

TS = 5 seconds

If MRKRTMP > 0:

TS = 18 seconds

Delay TS seconds (by putting job to sleep via "DELAYJOB")

NVWORD1 = 0640<sub>vn</sub> (causes "CLOCKJOB" execution)

Set restart group 6 to phase 2 (i.e. 6.2, causing "PRE40.6" to be called if a restart)

Proceed to "P4OS/SV"

P4OS/SV

$$T_{\text{decl}} = T_{\text{ig}} - K_{\text{sec29p96}}$$

Perform "MIDTOAV1": if return to calling address +1, skip next line  
if return to calling address +2 (time slipped),  
proceed

$$T_{\text{ig}} = T_{\text{pptml}} + K_{\text{sec29p96}}$$

$$P4OTMP = TS - K_{5\text{secdp}} \quad (\text{TS set by "MIDTOAV1"})$$

Call "TIGBLINK" in P4OTMP seconds

End of job

TIGBLINK          Entered 5 seconds before start of Average-G

Call "TIGAVEG" in 5 seconds

NVWORD1 = +0

Establish "P4OBLINKR" (priority 14<sub>8</sub>)

End of task

P4OBLINKR

Perform "CLEANDSP" (blanks DSKY except for program number)

End of job

TIGAVEG

Set bit 4(PFRATFLG) of FLAGWRD2 = 0

If bit 11(RCSBURN) of FLAGWRD1 = 1:

$$NVWORD1 = 1685_{\text{vn}} \quad (\text{Tag here "P41REDSP"})$$

Call "TTG/O" in 29.96 seconds

Proceed to "PREREAD"

$$NVWORD1 = 0640_{\text{vn}} \quad (P40)$$

Call "TIG-5" in 24.96 seconds (i.e. at exactly 5 seconds before  $T_{\text{ig}}$ )

Proceed to "PREREAD"

### TIG-5

Call "TIG-0" in 5 seconds

NVWORD1 = - 400<sub>g</sub> (causes "NV50DSP" to paste a V99)

Establish "S40.13" (priority 20<sub>g</sub>)

End of task

### TIG-0

Set bit 13(IGNFLAG) of FLAGWRD7 = 1

If bit 12(ASTNFLAG) of FLAGWRD7 = 0, End of task (no authorization received yet)

NVWORD1 = 0640<sub>vn</sub>

Proceed to "IGNITION"

### IGNITION

Entered from "TIG-0" and called by "V99P"

OGAD = CDU<sub>x</sub>

T<sub>evt</sub> = T<sub>now</sub>

Set bit 7(ENGONFLG) of FLAGWRD5 = 1

Set bit 13(SPS Engine On) of channel 11 = 1

If bit 9(IMPULSW) of FLAGWRD2 = 0:

Set bit 13(STRULLSW) of FLAGWRD6 = 1 (enable steering below)

If bit 9(IMPULSW) of FLAGWRD2 = 1:

Set bit 13(STRULLSW) of FLAGWRD6 = 0 (Tag here "IMPLBURN")

T<sub>ig</sub> = T<sub>now</sub> + T<sub>go</sub>

If bit 3(SBFLAG) of FLAGWRD2 = 1:

TS<sub>1</sub> = K<sub>decpl</sub> T<sub>go</sub> + 1 (Bl4 in units of 10 cs, truncated)

TS<sub>2</sub> = 10 TS<sub>1</sub>

TS = TS<sub>2</sub> - C<sub>sbdelt</sub>

If TS ≤ 0:

TS = 20 + 10 (TS/10, fractional part)

Call "SBTASK" in TS centi-seconds



(If bit 9(IMPULSW) of FLAGWRD2 = 1):

    If bit 3(SBFLAG) of FLAGWRD2 = 0:

        Call "ENGINEOFF" in ( $T_{go} + 1$ ) centi-seconds (uses low half loaded by "S40.13")

    Set bit 9(IMPULSW) of FLAGWRD2 = 0

Set bits 15-14(DAPBIT1, DAPBIT2) of FLAGWRD6 =  $00_2$  (Tag here "PREPTVC")

T5LOC = "T5IDLOC" (turn off RCS DAP except for jet timing)

Delay 0.40 seconds (for thrust buildup)

TVCPHASE = -1

TVCEXPHS = +0

Set bits 15-14(DAPBIT1, DAPBIT2) of FLAGWRD6 =  $10_2$

Set restart group 6 to phase 3 (6.3, removing "PRE40.6" protection)

ERRBTMP = - (ERROR<sub>1</sub>, ERROR<sub>2</sub>)

Set TIME5 to cause program interrupt #2 in 0.01 seconds

T5LOC = "TVCDAPON"

Delay 1.6 seconds (i.e. until 2 seconds from ignition)

If bit 13(STRULLSW) of FLAGWRD6 = 1:

    Set bit 11(STEERSW) of FLAGWRD2 = 1

Channel 5 = 0 (turn off ullage jets)

End of task

#### CLOKTASK

If bit 11(TIMRFLAG) of FLAGWRD7 = 0, End of task

$T_{togo} = T_{now} - T_{ig}$

Call "CLOKTASK" in 1 second

If NVWORD1 = +0, End of task

If NWORD1  $\neq$  0685<sub>vn</sub>: (i.e. not P41 before Average-G)

Establish "CLOCKJOB" (priority 27<sub>g</sub>)

End of task

Establish "DYNDISP" (priority 27<sub>g</sub>)

End of task

#### DYNDISP

TS = V<sub>gtig</sub>

Perform "S41.1"

V<sub>gbody</sub> = TS

Proceed to 3rd line of "CLOCKJOB"

#### CLOCKJOB

CDUSPOT = CDU (most significant halves loaded, in y,z,x order)

Perform "QUICTRIG"

Inhibit interrupts (released as e.g. part of End of job)

If NWORD1 = +0, End of job

If NWORD1 > 0:

TS = NWORD1

Proceed to "REGODSP"

If NWORD1 < 0: (e.g. -400<sub>g</sub>, for V99 generation)

TS = 0640<sub>vn</sub>

Proceed to "CLOCPLAY": if terminate, proceed to "V99T"  
if proceed, proceed to "V99P"  
otherwise, proceed to "V99E"

If NWORD1 = -0: (as it will, for V97 generation)

TS = 0640<sub>vn</sub>

Proceed to "CLOCPLAY": if terminate, proceed to "V97T"  
if proceed, proceed to "V97P"  
otherwise, proceed to "V97E"

V97E Entered for ENTR response to V97, engine fail

NVWORD1 = 0640<sub>vn</sub>

T<sub>ig<sub>sp</sub></sub> = -3276.8 seconds (most significant half set to -20, to give  
59 59 display in "CLOCKJOB" for T<sub>now</sub>  $\sim 9$  min)

Perform "SPSOFF"

Delay 2.5 seconds (by putting job to sleep via "DELAYJOB")

Call "PRE40.6" in 0.01 seconds

Inhibit interrupts (done in waitlist routine)

Perform "RCSDAPON"

Release interrupts

Set bit 12(ASTNFLAG) of FLAGWRD7 = 0

Set bit 13(IGNFLAG) of FLAGWRD7 = 1

Set bit 3(SBFLAG) of FLAGWRD2 = 1

NVWORD1 = - 400<sub>g</sub> (gives a V99 flash, cf. "TIG-5")

Establish "S40.13" (priority 20<sub>g</sub>)

End of job

V97P Entered for a PRO response to V97, engine fail

NVWORD1 = 0640<sub>vn</sub>

Set bit 6(IDLEFAIL) of FLAGWRD1 = 1 (keeps "S40.8" fail logic from  
being triggered)

Set bit 11(STEERSW) of FLAGWRD2 = 1

Delay 2 seconds (by putting job to sleep via "DELAYJOB")

Set bit 6(IDLEFAIL) of FLAGWRD1 = 0

End of job

V97T Entered for a V34E response to V97, engine fail

NVWORD1 = +0

Perform "SPSOFF"

Delay 2.5 seconds (by putting job to sleep via "DELAYJOB")

Inhibit interrupts

Perform "RCSDAPON"

Release interrupts

Proceed to "V99T"

V99E Entered for an ENTR response to V99, engine ignition

Inhibit interrupts

Perform "TVCZAP"

Proceed to "P4ORCS"

V99P Entered for a PRO response to V99, engine ignition

Inhibit interrupts (released as part of End of job)

If bit 12(ASTNFLAG) of FLAGWRD7 = 0: (check for restart reasons)

Set bit 12(ASTNFLAG) of FLAGWRD7 = 1

If bit 13(IGNFLAG) of FLAGWRD7 = 1: ("TIG-O" has been done)

Call "IGNITION" in 0.01 second

NVWORD1 = 0640<sub>vn</sub>

End of job

V99T Entered for a V34E response to V99 (and from "V97T")

Inhibit interrupts

Perform "TVCZAP"

Proceed to "POST41"

S40.81 Entered from "S40.8" when time-to-go below 4 seconds

OMEGAC = 0

CNTR = +MAX

Inhibit interrupts

$TS = (T_{ig} - T_{now})$ , with sign agreement forced

$T_{go} + 1 = TS + 1$ , limited  $\gg 1$  centi-second (most significant half of  $T_{go}$  not loaded)

Call "ENGINEOFF" in  $(T_{go} + 1)$  centi-seconds

Set bit 11(STEERSW) of FLAGWRD2 = 0

Release interrupts

Proceed to "SERVEXIT"

#### ENGINEOFF

Perform "SPSOFF"

Delay 2.5 seconds

Perform "SETMAXDB"

Perform "RCSDAPON"

Perform "MASSPROP"

Perform "TVCZAP"

Establish "POSTBURN" (priority 12<sub>g</sub>)

End of task

#### POSTBURN

TS = 1640<sub>vn</sub>

Proceed to "REFLASH": if terminate, proceed to "POST41"  
if proceed, proceed  
otherwise, proceed to previous line

Proceed to "P4ORCS"

#### P4ORCS

AVEGEXIT = "CALCN85"

Delay 2 seconds (by putting job to sleep via "DELAYJOB")

Inhibit interrupts

Perform "SETMINDB"

Release interrupts

Proceed to "TIGNOW"



TIGNOW Entered from "P4ORCS" and established by "TTG/O" for P41

TS = 1685<sub>vn</sub>

Proceed to "REFLASH": if terminate, proceed to "POST41"  
if proceed, proceed  
otherwise, proceed to previous line

AVEGEXIT = "SERVEXIT"

Proceed to second line of "GOTOPOOH"

POST41

AVEGEXIT = "SERVEXIT"

Proceed to "GOTOPOOH"

P41CSM

Set bit 11(RCSBURN) of FLAGWRD1 = 1

CSTEER = 0

TS = K<sub>frcs2</sub>

If bit 15(2JETSFLG) of FLAGWRD1 = 0: (i.e. 4 jets)

TS = 2 TS

CAPF = TS

Proceed to "P4OS/F"

TTG/O Called by "TIGAVEG" for P41

Establish "TIGNOW" (priority 20<sub>g</sub>)

Set bit 11(TIMRFLAG) of FLAGWRD7 = 0

End of task

P47CSM

Set bit 5(P48FLAG) of FLAGWRD11 = 0

Perform "R02BOTH" (Tag here "ASKIMU")

Perform "MIDTOAV2"

P4OTMP = TS+1 (TS set when exit from "MIDTOAV2")

Call "TIGON" in P4OTMP centi-seconds

If bit 5(P48FLAG) of FIGWRD11 = 1:

Call "CTASK" in 0.01 second

End of job

#### P48CSM

Set bit 5(P48FLAG) of FIGWRD11 = 1

Set bit 7(RNDVZFLG) of FLAGWRDO = 0

FIXTIME = 0

Perform "N72STUFF"

Proceed to second line of "P47CSM"

#### TIGON

AVEGEXIT = "CALCN83"

Establish "P47BODY" (priority 30<sub>g</sub>)

Proceed to "PREREAD"

#### P47BODY

DELVIMU = 0

V<sub>gbody</sub> = 0

VSAV = 0

VGDISP = 0 (due to "vector" store order for VSAV)

NC1TIG = 0 (due to "vector" store order for VSAV)

Change priority of present job to 15<sub>g</sub>

If bit 5(P48FLAG) of FIGWRD11 = 0:

TS = 1683<sub>vn</sub>

Proceed to "GOFLASH": if terminate, proceed to "GOTOPOOH"  
if proceed, proceed  
otherwise, proceed to "P47BODY"

Proceed to "GOTOPOOH"

If bit 10(NOUNFLG) of FLGWRD11 = 0: (P48FLAG = 1 if get here)

TS = 1676<sub>vn</sub>

Proceed to "GOFLASH": if terminate, proceed to "GOTOPOOH"  
if proceed, proceed  
otherwise, proceed to "P47BODY"

TS = 1677<sub>vn</sub> (Tag here "SHOWOPT")

Proceed to "GOFLASH": if terminate, proceed to "GOTOPOOH"  
if proceed, proceed  
otherwise, proceed to "P47BODY"

TS = 1683<sub>vn</sub> (Tag here "SHOW83")

Proceed to "GOFLASH": if terminate, proceed to "GOTOPOOH"  
if proceed, proceed  
otherwise, proceed to "P47BODY"

Proceed to "GOTOPOOH"

P77

T<sub>ig</sub> = NONTIG

TS = 0633<sub>vn</sub>

Proceed to "GOFLASH": if terminate, proceed to "ENDP76"  
if proceed, proceed  
otherwise, proceed to previous line

TS = 0681<sub>vn</sub>

Proceed to "GOFLASH": if terminate, proceed to "ENDP76"  
if proceed, proceed  
otherwise, proceed to previous line

Set bit 1(NODOV37) of FLAGWRD2 = 1

T<sub>decl</sub> = T<sub>ig</sub>

Perform "CSMPREC"

TS<sub>3</sub> = - unitR<sub>att</sub>

TS<sub>2</sub> = unit(TS<sub>3</sub> \* V<sub>att</sub>)

TS<sub>1</sub> = unit(TS<sub>2</sub> \* TS<sub>3</sub>)

$$\underline{TS}_4 = \underline{V}_{att} + \underline{DELVLVC} \begin{bmatrix} \underline{TS}_1 \\ \underline{TS}_2 \\ \underline{TS}_3 \end{bmatrix} \quad (\text{in 6D})$$

Perform "INTSTALL"

$$\underline{VCV} = \underline{TS}_4$$

$$\underline{RCV} = \underline{R}_{att}$$

$$\underline{T}_{et} = \underline{T}_{ig}$$

Set bit 4(CONICINT) of FLAGWRD3 = 0

$$\underline{T}_{decl} = \underline{T}_{etcm}$$

Perform "INTEGRVS"

Perform "INTSTALL"

$$\underline{R}_{rect} = \underline{R}_{att1}$$

$$\underline{RCV} = \underline{R}_{rect}$$

$$\underline{T}_{et} = \underline{T}_{att}$$

$$\underline{V}_{rect} = \underline{V}_{att1}$$

$$\underline{VCV} = \underline{V}_{rect}$$

$$\underline{TDELTA V} = 0$$

$$\underline{TNUV} = 0$$

$$\underline{T}_c = 0$$

XKEP = 0 (notation also "XPREV")

Set bit 13(INTGRAB) of FLAGWRD10 = 1

Perform "MOVEACSM"

If bit 1(AVEMIDSW) of FLAGWRD9 = 0:

$$\underline{R} = \underline{RCV} + \underline{TDELTA V}$$

$$\underline{V} = \underline{VCV} + \underline{TNUV}$$

$$\underline{T}_{pptm} = \underline{T}_{et}$$

QPRET = Return address (to line after next)

Perform "INTWAKE" (starting at 3rd from last line, awaken jobs)

Set bit 1(NODOV37) of FLAGWRD2 = 0 (Tag here "OUT")

MRKBUF1 = -1

Proceed to second line of "GOTOPOOH"

ENDP76

MRKBUF1 = -1

Proceed to "GOTOPOOH"



## Quantities in Computations

See also list of major variables and list of routines

AUTOY: Value of out-of-plane position computed in "PRECSET", scale factor B29, units meters. It is computed as  $\underline{R}_{act3} \cdot \underline{UP1}$ .

AVEGEXIT: See General Program Control.

$C_{clockdelt}$ : Single precision erasable memory constant, program notation "CLOCKDELT", scale factor B14, units centi-seconds, used in "P4OS/F" to control the phasing of "CLOKTASK" with respect to nominal ( $T_{ig}$ ) ignition time to minimize the probability of loss of telemetry interrupts.

$C_{sbdelt}$ : Single precision erasable memory constant, program notation "SBDELT", scale factor B14, units centi-seconds, used in "IGNITION" to specify the "startup delay" for "SBTASK" (the lower the value, the less the probability of loss of telemetry interrupts because of lower duty cycle and larger arguments for the square root routine).

$C_{tcs}$ : Erasable memory constant, program notation "TCS" (alternative notation "DT/CC-SR"), scale factor B28, units centi-seconds, giving the time delay between NCC and NSR maneuvers. It is used in "P33" to initialize  $T_{cdh}$  (the NSR time) and in "NCCLOOP" to compute  $T_{as}$ .

CADRF<sub>LSH+2</sub>: See Display Interface Routines.

CAPF: Nominal value of engine thrust (SPS, 2-jet RCS, or 4-jet RCS), scale factor B7, units "M. Newtons", where the unit is equivalent to  $10^4$  Newtons (or the same as expressing force in kilogram-meters/centi-second<sup>2</sup>). Program notation is "F".

CDUSPOT: See Coordinate Transformations.

CMYDOT: Value of CSM out-of-plane velocity computed in "PRECSET", scale factor B7, units meters/centi-second. It is computed as  $\underline{V}_{act3} \cdot \underline{UP1}$ .

CNTR: See Digital Autopilot Interface Routines. A setting of +MAX (in "S40.81" for cutoff) corresponds to a time delay of about 2.27 hours, effectively disabling updates in "TVCEXEC".

CSTEER: See Steering Computations.

CSTH: See Conic Routines.

DELVEET<sub>1</sub>, DELVEET<sub>2</sub>, DELVEET<sub>3</sub>: See Rendezvous Computations.

DELVIMU: See Display Computations.

DELVLVC: Value of velocity increment for display in local vertical coordinates, scale factor B7, units meters/centi-second, displayed by N81 and N86. Program notation also "DELWSLV", and cells can also be used for temporary storage purposes before the display. The z-axis is along  $-r$  at ignition, the y-axis is along  $y * r$ , and the x-axis is along  $y * z$ .

DELVSIN: Value of velocity increment in reference inertial coordinates computed for external Delta-V burns (P30, P31, P32, P34, and P38), scale factor B7, units meters/centi-second. Same cell also has the tag DELVEET3: for compatibility with "VN1645", however, the "loading" of DELVEET3 with DELVSIN is shown where necessary. DELVSIN is used in "S40.1" to compute the required velocity-to-be-gained for the external Delta-V burn.

DELVTPF: Value of magnitude of velocity change required at intercept, scale factor B7, units meters/centi-second, displayed in R2 of N58.

DELVTPI: Value of magnitude of velocity change required to perform the (non-midcourse) burn in P35, scale factor B7, units meters/centi-second, displayed in R1 of N58.

DHDSP: See Rendezvous Computations.

DHNCC: See Rendezvous Computations.

DVLOS: Value of required velocity increment in line-of-sight coordinates, scale factor B7, units meters/centi-second, computed for display by N59. The R1 component is in the direction of ULOS (i.e. "forward/aft", with + direction "forward"); the R2 component is along  $R2 * ULOS$  (i.e. "right/left", with + direction "right"); and the R3 component is along  $UNRM * ULOS$  (i.e. "up/down" with + direction "down").

DVTOTAL: See General Program Control.

ELEV: See Rendezvous Computations.

ERRBTMP: See Digital Autopilot TVC Routines.

ERROR<sub>1</sub>, ERROR<sub>2</sub>: See Digital Autopilot RCS Routines.

EXTVBACT: See Verb Definitions.

FIXTIME: See Orbital and Rendezvous Navigation.

HALFREVS: Single precision value of number of half-revolutions, scale factor B14, units of half-revolutions, displayed by R1 of N57 and used to load NC for subsequent use (a left shift of 8 provides NC with a value scaled B5 in units of revolutions, so value employed in computations becomes modulo 64 half-revolutions).

HAPO, HPER: Apogee and perigee altitudes above  $K_{\text{rpad}}$  computed in P30 for display in R1 and R2 of N42, scale factor B29, units meters, limited to maximum of  $K_{\text{maxnm}}$ .

INTIME: See Rendezvous Computations.

$K_{\text{ld10fps}}$ : Constant, program notation "1/10FPS", scale factor B7, units meters/centi-second. Value is  $3.048\text{E-4} \times 2^{-7}$ , corresponding to  $0.1 \times 0.3048 \times 10^{-2} \times 2^{-7}$ , where first term is value in fps, second converts to meters, third converts to centi-seconds, and fourth is scale factor.

$K_{\text{5secdp}}$ : Constant, program notation "5SECDP", scale factor B28, units centi-seconds. Value is  $500 \times 2^{-28}$ , corresponding to 5 seconds.

$K_{\text{60min}}$ : Constant, program notation "60MIN", scale factor B28, units centi-seconds. Value is  $360000 \times 2^{-28}$ , corresponding to  $60 \times 60 \times 10^2 \times 2^{-28}$ , where first term is value in minutes, second converts to seconds, third converts to centi-seconds, and fourth is scale factor.

$K_{\text{60minh}}$ : Constant, program notation "60MINH", scale factor B28, units centi-seconds. Value is  $360000 \times 2^{-28}$ , the same as  $K_{\text{60min}}$ .

$K_{\text{atiginc}}$ : Constant, program notation "ATIGINC", scale factor B28, units centi-seconds. Value is  $18000 \times 2^{-28}$ , corresponding to  $3 \times 60 \times 10^2 \times 2^{-28}$ , where first term is value in minutes, second converts to seconds, third converts to centi-seconds, and fourth is scale factor.

$K_{\text{centang}}$ : Constant, program notation "CENTANG", scale factor B0, units revolutions. Value is  $13434_8 16163_8$ , corresponding to about  $0.36111112$  revolutions or  $130^\circ$ .

$K_{\text{cs359p}}$ : Constant, program notation "CS359+", scale factor B1, octal value  $17777_8 37776_8$ , corresponding to a true value of  $(1 - 2^{-26})$ . Used in "P38" to set SNTH for "TIMETHET", giving an angle of  $\sim 90^\circ$ .

$K_{\text{decpl}}$ : Single precision constant, program notation "DEC.1", scale factor B0, value 0.1. Used to convert (by truncation) time information in units of centi-seconds to deci-seconds.

$K_{\text{dpm01}}$ : Constant, program notation "DP-.01", scale factor B0, units revolutions. Octal value is  $77777_8 61337_8$ , corresponding to  $-16440_8 \times 2^{-28}$  or  $7456 \times 2^{-28}$ . This is approximately equal to  $-0.01/360$ , i.e.  $-0.01^\circ$ .

$K_{\text{feng}}$ : Constant, program notation "FENG", scale factor B7, same units as CAPF. Value is  $9.1188544 \times 2^{-7}$ , corresponding approximately to  $20500 \times 9.80665 \times 0.45359237 \times 10^{-4} \times 2^{-7}$ , where first term is SPS thrust in pounds, second is gravity in meters/second<sup>2</sup>, third converts pounds to kilograms, fourth converts to centi-seconds<sup>2</sup>, and fifth is scale factor.

$K_{frcs2}$ : Constant, program notation "FRCS2", scale factor B7, same units as CAPF. Value is  $0.087437837 \times 2^{-7}$ , corresponding approximately to  $2 \times 99.8 \times 0.98481 \times 9.80665 \times 0.45359237 \times 10^{-4} \times 2^{-7}$ , where first term accounts for 2 RCS jets, second is RCS thrust in pounds, third is cosine  $10^\circ$  (account for canting of nozzle), fourth is gravity in meters/second<sup>2</sup>, fifth converts pounds to kilograms, sixth converts to centi-seconds<sup>2</sup>, and seventh is scale factor. The product of the first three terms is about 196.568 pounds.

$K_{marsdp}$ : Constant, program notation "MARSDP", scale factor B0, units revolutions. Value in octal is 00000<sub>8</sub> 35100<sub>8</sub>, corresponding to the complement of  $2 K_{dpm01}$ , or approximately 0.02 degrees.

$K_{maxnm}$ : See Display Computations.

$K_{posvel2}$ : Single precision constant, program notation "POSVEL2", scale factor B19, units meters. Stored (and nominal) value is  $608 \times 2^{-19}$ , corresponding to about  $1994.751 \times 0.3048 \times 2^{-19}$ , where first term is value in feet, second converts to meters, and third is scale factor.

$K_{posvel3}$ : Single precision constant, program notation "POSVEL3", scale factor B0, units meters/centi-second. Nominal value is 0.00608, corresponding to about  $1.994751 \times 0.3048 \times 0.01$ , where first term is value in fps, second converts to meters, and third to centi-seconds. The octal value is 00144<sub>8</sub>, corresponding to  $100 \times 2^{-14}$  or about 2.0025 fps.

$K_{rpad}$ : Constant, program notation "RPAD", scale factor B29, units meters. Value is  $6373338 \times 2^{-29}$ , corresponding to 6373338 meters (or 20,909,901.5747 feet or 3441.3272 nmi).

$K_{sec29p96}$ : Constant, program notation "SEC29.96", scale factor B28, units centi-seconds. Value is  $2996 \times 2^{-28}$ , corresponding to 29.96 seconds.

LASTTIG: Value of ignition time for previous burn (loaded with T<sub>ig</sub> in "P38" for display via N39), scale factor B28, units centi-seconds. Note that if previous burn used P40, then initialization actually is to cutoff time (see "IGNITION" or "S40.8"), since NOMFIG not employed.

LMYDOT: Value of OWS out-of-plane velocity computed in "PRECSET", scale factor B7, units meters/centi-second. It is computed as  $\frac{V}{pass3} \cdot UNRM$ .

MRKBUF1: See Optics Computations.

MRKRTMP: Single precision cell, scale factor B14, used to indicate whether only the gimbal trim is to be performed (value of -1), or if a gimbal drive test in addition is to be performed (value of +1).

NBRCYCLS: See Steering Computations.

NC: See Rendezvous Computations.

NC1TIG: Value of ignition time for the NC1 maneuver (P31), scale factor B28, units centi-seconds, loaded by N95. Set 0 in "P47BODY".

NC2TIG: Value of ignition time for the NC2 maneuver (P32), scale factor B28, units centi-seconds, loaded by N28. Also loaded in "NC12OUT".

NN1: See Rendezvous Computations.

NOMTIG: Value of nominal ignition time sampled in "VN1645" and "P4OS/F", scale factor B28, units centi-seconds. It is used in "S40.9" to correct Lambert for oblateness, and in "P77" to initialize the N33 display.

NOMTPI: See Rendezvous Computations.

NVWORD1: Single precision communication cell with "CLOCKTASK". If it is +0,  $T_{\text{togo}}$  is merely updated; if it is positive non-zero, "CLOCKJOB" is usually established (except in P41, where a value of 0685<sub>vn</sub> causes "DYNDISP" to be established instead) to cause display of the verb-noun; if negative non-zero, "CLOCKJOB" is established to generate an 0640<sub>vn</sub> display with option bits set to cause (for NVWORD1 = -400<sub>g</sub>) the pasting of verb 99 by "NV50DSP"; and if negative 0 (set in "S40.8" for low thrust) then the same type of display is generated as for negative non-zero, except the verb is 97 instead of 99. The option bits are set by "CLOCPLAY".

OGAD: See Digital Autopilot TVC Routines.

OMEGAC: See Steering Computations.

P4OTMP: Required time delay until "TIGBLNK" is performed, scale factor B28, units centi-seconds (used for restart protection purposes). Used sp in P47 and P48 for analogous purposes, B14 centi-seconds. It uses the same cells as the z component of DELVEET2.

pMGA: See Display Computations.

POSTTPI: Value of perigee altitude, scale factor B29, units meters, predicted to exist after the burn. Could be observed in R1 of N75.

QPRET: See Orbital Integration.

$R_{\text{act2}}$ ,  $R_{\text{act3}}$ : See Rendezvous Computations.

$R_{\text{init}}$ : See Rendezvous Computations.

$R_{\text{pass2}}$ ,  $R_{\text{pass3}}$ : See Rendezvous Computations.

$R_{\text{rect}}$ : See Orbital Integration.



$R_{targ}$ : See Rendezvous Computations.

$R_{tig}$ : Value of position vector at ignition time, scale factor B29, units meters.

RCV: See Orbital Integration.

RVEC: See Conic Routines.

SNTH: See Conic Routines.

SUBEXIT: Single precision cell used to retain return address information.

T: See Conic Routines.

$T_c$ : See Orbital Integration.

$T_{cdh}$ ,  $T_{csi}$ : See Rendezvous Computations.

$T_{et}$ ,  $T_{etcm}$ : See Orbital Integration.

$T_{go}$ : See Steering Computations.

$T_{pass4}$ : Time tag associated with target vector for Lambert computation, scale factor B28, units centi-seconds. In P35 and P36, it is the intercept time.

$T_{pptml}$ : See IMU Computations.

$T_{strt}$ : Time of start of P36 targeting computation, scale factor B28, units centi-seconds (loaded at start of "P35/P75B", but not subsequently referenced).

$T_{togo}$ : Value of time until  $T_{ig}$  (computed as  $T_{now} - T_{ig}$  in "CLOKTASK", hence negative before  $T_{ig}$  has arrived), scale factor B28, units centi-seconds. Displayed by N35, N40, N45, and N80.

$T_{tpi}$ : See Rendezvous Computations.

$T_{tpi0}$ : Value of  $T_{tpi}$  sampled in "P34" and "P35", used to derive T2TOT3, the change in  $T_{tpi}$ , for display. Scale factor B28, units centi-seconds.

T1TOT2: Value of time interval for P34 between  $T_{tpi}$  and  $T_{cdh}$  (i.e. TNSR), scale factor B28, units centi-seconds. It is computed in "P33/P73B", and to permit display in R2 of N75 is made modulo one hour there, if difference positive.

T2TOT3: Value of time interval between  $T_{tpi}$  and  $T_{tpi0}$ , scale factor B28, units centi-seconds, computed in "P33/P73B" and "P34/P74C" (for P34 and P35 respectively). To permit display in R3 of N58/N75, it is made modulo one hour.

T5LOC: See Digital Autopilot Interface Routines.

TDELTAV: See Orbital Integration.

TIME5: See Digital Autopilot Interface Routines.

TIMEC: See Rendezvous Computations.

TNUV: See Orbital Integration.

TVCEPHS, TVCPHASE: See Digital Autopilot TVC Routines.

ULOS: See Rendezvous Computations.

UNRM: See Rendezvous Computations.

UPl: See Rendezvous Computations.

UTPIT, UTYAW: See Orbital and Rendezvous Navigation.

$\underline{V}_{act2}$ ,  $\underline{V}_{act3}$ : See Rendezvous Computations.

$\underline{V}_{gbody}$ : Velocity-to-be-gained in (RCS DAP) "control" axes, scale factor B7, units meters/centi-second. See "S41.1" for the transformation used. In P41, it is updated every second prior to start of Average-G (after return from R60), due to "DYNDISP" computations. Program notation also "DELVCTL", and vector displayed in N85.

$\underline{V}_{gtig}$ : Velocity-to-be-gained at ignition time in reference coordinates, scale factor B7, units meters/centi-second. Program notation also "VGPREV", reflecting the fact that the same cells used to store previous value of velocity-to-be-gained in "VGCOMP".

$\underline{V}_{init}$ ,  $\underline{V}_{iprime}$ : See Rendezvous Computations.

$\underline{V}_{pass2}$ ,  $\underline{V}_{pass3}$ ,  $\underline{V}_{pass4}$ : See Rendezvous Computations.

$\underline{V}_{rect}$ : See Orbital Integration.

$\underline{V}_{tig}$ : Value of velocity vector at ignition time, scale factor B7, units meters/centi-second.

$\underline{V}_{tprime}$ : See Rendezvous Computations. Also used in "P34/P74C" and "P35/P75B" for temporary storage purposes, to permit detection of a change in the specified velocity increment for burn (or  $T_{tpi}$  specification, in which case is B28 centi-seconds, x component only).

VCV: See Orbital Integration.

VGDISP: Magnitude of velocity-to-be-gained vector, scale factor B7, units meters/centi-second. It is displayed in N40, N42, and N80.

VSAV: See Measurement Incorporation. Although notation suggests that it is a vector, it is used as a scalar (hence vector storage in "P47BODY" impacts VGDISP and NCLTIG also).

VVEC: See Conic Routines.

WRENDPOS, WRENDVEL: See Measurement Incorporation.

XKEP: See Orbital Integration.

XXXALT: Value of base altitude for computing apogee and perigee information, scale factor B29, units meters (set to  $K_{rpad}$ ).

YCSM: Value of CSM out-of-plane position for display in R1 of N90, scale factor B29, units meters. It is the complement of AUTOY.

YDOTC: Value of CSM out-of-plane velocity for display in R2 of N90, scale factor B7, units meters/centi-second. It is the complement of CMYDOT.

YDOTL: Value of OWS out-of-plane velocity for display in R3 of N90, scale factor B7, units meters/centi-second. It is the complement of LMYDOT.



## Conic Routines

### KEPLERN

(Entered only from "KEPPREP")

Reset overflow indicator

Start of "Kepler Subroutine"

ITERCTR = 20

ldMU =  $K_{mutab_2}$

ROOTMU =  $K_{mutab_4}$

ldROOTMU =  $K_{mutab_6}$

URRECT = unit $R_{rect}$

R1 =  $|R_{rect}|$

KEPC1 = ldROOTMU  $R_{rect} \cdot V_{rect}$

KEPC2 = ldMU R1  $|V_{rect}|^2 - 1$

ALPHA = (1 - KEPC2) / R1

If ALPHA < 0:

TS =  $K_{m50sc} / ALPHA$

If overflow, reset indicator and set TS = +MAX

If no overflow:

TS =  $\sqrt{TS}$

If ALPHA  $\geq$  0:

TS =  $K_{2pisc} / \sqrt{ALPHA}$

If overflow, reset indicator and set TS = +MAX

XMAX = TS

(Tag here "STOREMAX")



$TS_1 = \text{ldROOTMU } XMAX / \text{ALPHA}$  (period, B28 c.s., in p.d. location 0)

Proceed to "PERIODCH"

PERIODCH

If  $TS_1 \geq 0$ :

If no overflow has taken place (e.g.  $TS_1 < 2^{28}$  cs):

$TS = |\text{TAUORB}| - TS_1$

If  $TS \geq 0$ :

$\text{TAUORB} = TS \text{ sgn } \text{TAUORB}$

Proceed to "PERIODCH" (starting at 3rd line, compute TS)

$X = XKEPNEW$  (tag here "MODDONE")

If  $X \text{ sgn } \text{TAUORB} > 0$ :

If  $|X \text{ sgn } \text{TAUORB}| - XMAX < 0$ :

Skip next line

$X = \frac{1}{2} XMAX \text{ sgn } \text{TAUORB}$

If  $\text{TAUORB} \geq 0$ : (tag here "STORBND")

$XMIN = 0$

If  $\text{TAUORB} < 0$ :

$XMIN = -XMAX$

$XMAX = 0$

Proceed to "DXCOMP"

### DXCOMP

$\text{EPSILONT} = \lfloor K_{\text{bee22}} \text{TAUORB} \rfloor$  (rounded multiplication)

$\text{DELX} = X - X_{\text{KEP}}$  (program notation also  $X_{\text{PREV}}$ )

Proceed to "KEPLOOP"

### KEPLOOP

$\text{PDXSQ} = X^2$

$\text{XI} = \text{ALPHA} \text{PDXSQ}$

Perform "DELTIME"

If overflow has taken place:

Reset overflow indicator

If  $X \gg 0$ :

$\text{XMAX} = X$

If  $X \ll 0$ :

$\text{XMIN} = X$

$\text{DELX} = \frac{1}{2} \text{DELX}$

If  $\text{DELX} = 0$ : (i.e. magnitude less than  $K_{\text{epsx}}$ )

Proceed to address specified by  $\text{KEPRTN}$

$X = X - \text{DELX}$

$T = T_c$

Proceed to "BRNCHCTR"

$\text{DELT} = \text{TAUORB} - T$

If  $\text{EPSILONT} - |\text{DELT}| \geq 0$ :

Proceed to "KEPCONVG"

$\text{TS} = \text{DELX} \text{DELT} / (T - T_c)$

(numerator rounded before division,  
result shifted right 1 place by  
truncated shift for scaling)

If  $TS \leq 0$ :

$$XMAX = X$$

$$TS_1 = TS$$

If  $XMIN - (X + TS) \gg 0$ : (or overflow taken place)

$$TS_1 = 0.9 (XMIN - X) \quad (\text{rounded multiplication})$$

If  $TS \gg 0$ :

$$XMIN = X$$

$$TS_1 = TS$$

If  $XMAX - (X + TS) \leq 0$ : (or overflow taken place)

$$TS_1 = 0.9 (XMAX - X) \quad (\text{rounded multiplication})$$

$$DELX = TS_1$$

If  $DELX = 0$ : (i.e. magnitude less than  $K_{\text{epsx}}$ )

Proceed to "KEPCONVG"

$$X = X + DELX$$

$$T_c = T$$

Proceed to "BRNCHCTR"

#### BRNCHCTR

$$ITERCTR = ITERCTR - 1$$

If  $ITERCTR = 0$ :

Proceed to "KEPCONVG"

Proceed to "KEPLOOP"

#### KEPCONVG

$$RCV = (R1 - XSQCpXI_p) URRECT + (T - 1dROOTMU X^3 SpXI_p) V_{\text{rect}}$$

$$TS = \frac{ROOTMU X}{|RCV|} (XI SpXI_p - 1) URRECT$$

$$VCV = TS + (1 - XSQCpXI_p / |RCV|) V_{\text{rect}}$$

$$T_c = T$$

$$XKEP = X \quad (\text{XKEP notation also XPREV})$$

Proceed to address specified by KEPRTN (loaded by "KEPPREP")

DELTIME (Entered with PDXSQ loaded and XI available)

$$\text{SpXIp} = K_{s0} + K_{s1} \text{XI} + K_{s2} \text{XI}^2 + K_{s3} \text{XI}^3 + K_{s4} \text{XI}^4 + K_{s5} \text{XI}^5 + \\ K_{s6} \text{XI}^6 + K_{s7} \text{XI}^7 + K_{s8} \text{XI}^8 + K_{s9} \text{XI}^9$$

$$\text{CXI} = K_{c0} + K_{c1} \text{XI} + K_{c2} \text{XI}^2 + K_{c3} \text{XI}^3 + K_{c4} \text{XI}^4 + K_{c5} \text{XI}^5 + \\ K_{c6} \text{XI}^6 + K_{c7} \text{XI}^7 + K_{c8} \text{XI}^8 + K_{c9} \text{XI}^9$$

$\text{XSQCpXIp} = \text{PDXSQ CXI}$  (This is "Kepler Equation Subroutine")

$\text{TS}_1 = \text{KEPC1 XSQCpXIp}$  (triple precision)

$\text{TS}_2 = \text{KEPC2 PDXSQ SpXIp}$  (triple precision)

$\text{TS}_3 = \text{X (R1 + TS}_2\text{)}$  (addition triple precision)

$\text{T} = \text{ldROOTMU (TS}_3 + \text{TS}_1\text{)}$  (rounded shift and multiply, triple precision addition)

Return

LAMBERT Entered only from "INITVEL2"

$\text{RTNCONC} = \text{Return address}$  Start of "Lambert Subroutine"

Reset overflow indicator

$\text{ITERCTR} = 20$

$\text{ldMU} = K_{\text{mutab}_2}$

$\text{ROOTMU} = K_{\text{mutab}_4}$

$\text{ldROOTMU} = K_{\text{mutab}_6}$

$\text{EPSILONL} = K_{\text{beel}_9} \text{TDESIRE}$  (rounded multiplication)

Set bit 3(ITERLSW) of FLAGWRD1 = 1

$\text{TS}_{\text{r1}} = \text{R1VEC}$

$\text{TS}_2 = \text{R2VEC}$

Perform "GEOM"

$\text{SNTH} = \text{TS}$

PDLAMBDA = R1 / MAGVEC2

CSTH =  $TS_{\cos}$  (computed by "GEOM")

lmCSTH = 1 - CSTH (scale factor B2, truncated shift)

TS = 1 - CSTH, using rounded right shift 1 (same scaling as lmCSTH)

If TS = 0: (angle 0° or 360°)

Set bit 3(NOSOLNSW) of FLAGWRD5 = 1

Proceed to address specified by RTNCONC

MAX = SNTH / lmCSTH +  $\sqrt{2 \text{ PDLAMBDA} / \text{TS}}$  (MAX notation also  
COGAMAX)

If overflow has taken place:

MAX =  $K_{\text{cogupm}}$

Skip next 3 lines

If MAX  $\geq$  0:

If MAX -  $K_{\text{cogupm}} \geq$  0:

MAX =  $K_{\text{cogupm}}$

CSTHmrHO = CSTH - PDLAMBDA

TS =  $K_{\text{coglom}}$

If GEOMSGN  $\geq$  0:

$TS_1 = \text{CSTHmrHO} / \text{SNTH}$

If  $TS_1$  does not overflow (not  $\gg 2^5$  in magnitude):

TS =  $TS_1$

MIN = TS (notation also COGAMIN)

If bit 2(GUESSSW) of FLAGWRD1 = 0:

TWEEKIT =  $2^{-14}$

If bit 2(GUESSSW) of FLAGWRD1 = 1:

TWEEKIT =  $2^{-2}$  (i.e. 0.25)

COGA =  $\frac{1}{2} \text{ MIN} + \frac{1}{2} \text{ MAX}$

(If bit 2(GUESSSW) of FLAGWRD1 = 1):

DCOGA = COGA

Proceed to "LAMBLOOP"

LAMBLOOP

TS =  $\text{lmCSTH} / (\text{SNTH COGA} - \text{CSTHmRHO})$

If TS  $\leq$  0:

If DCOGA  $\geq$  0:

MAX = COGA

If DCOGA  $<$  0:

MIN = COGA

Proceed to "COMMONLM"

P = TS (scaled B4)

RdA =  $K_{2b6} - P (K_{1b10} + \text{COGA}^2)$  (scaled B6)

If overflow has taken place: (e.g.  $|P| \geq 2^4$  or  $|RdA| \geq 2^6$ )

MIN = COGA

Proceed to "COMMONLM"

TS = P

Perform "GETX"

TPREV = T

If bit 7(INFINFLG) of FLAGWRD8 = 1:

If DCOGA  $\geq$  0:

MAX = COGA

If DCOGA  $<$  0:

MIN = COGA

Proceed to "COMMONLM"

Perform "DELTIME"



If overflow has taken place:

T = TPREV

MAX = COGA

Proceed to "COMMONLM"

TERRLAMB = TDESIRED - T

If EPSILONL - |TERRLAMB|  $\geq$  0:

Proceed to "INITV"

ITERCTR = ITERCTR - 1

If ITERCTR = 0:

Proceed to "SUFFCHEK"

If bit 3(ITERLSW) of FLAGWRD1 = 0:

If (T - TPREV) = 0:

Proceed to "SUFFCHEK"

Perform "ITERATOR"

If DCOGA = 0: (i.e. magnitude less than  $2^{-23}$ )

Proceed to "SUFFCHEK"

COGA = COGA + DCOGA

Proceed to "LAMBLOOP"

#### COMMONLM

DCOGA =  $\frac{1}{2}$  DCOGA

If DCOGA = 0: (i.e. magnitude less than  $2^{-23}$ )

Proceed to "SUFFCHEK"

COGA = COGA - DCOGA

Proceed to "LAMBLOOP"

#### SUFFCHEK

If  $\frac{1}{4}$  TDESIRED +  $K_{lcs}$  - |TERRLAMB|  $<$  0:

Set bit 3(NOSOLNSW) of FLAGWRD5 = 1

Proceed to address specified by RTNCONC

Proceed to "INITV"

### INITV

$$TS = \text{ROOTMU} \sqrt{P / R1}$$

$$WVEC = TS \text{ COGA } UR1 + TS (UN * UR1)$$

Set bit 3(NOSOLNSW) of FLAGWRD5 = 0

If VTARGETAG = 0:

$$TS = \text{MAGVEC2}$$

Perform "LAMENTER"

$$V_{\text{target}} = TS_{\text{v}}$$

Proceed to address specified by RTNCONC

### ITERATOR (Entered only from "LAMBLOOP")

If bit 3(ITER1SW) of FLAGWRD1 = 0:      Start of "Iterator  
Subroutine"

$$TS = \text{TERRLAMB } DCOGA / (T - \text{TPREV})$$

If bit 6(ORDERSW) of FLAGWRD8 = 1:

$$TS = |TS| \text{ sgnTERRLAMB}$$

If bit 3(ITER1SW) of FLAGWRD1 = 1:

Set bit 3(ITER1SW) of FLAGWRD1 = 0

$$TS = (\text{MAX TWECKIT} - \text{MIN TWECKIT}) \text{ sgnTERRLAMB}$$

If TS < 0:

If bit 6(ORDERSW) of FLAGWRD8 = 0:

$$\text{MAX} = \text{COGA}$$

If MIN - COGA - TS > 0: (or overflow taken place)

$$TS = 0.9 (\text{MIN} - \text{COGA})$$

$$DCOGA = TS$$

Return

If bit 6(ORDERSW) of FLAGWRD8 = 0:      (TS ≥ 0)

$$\text{MIN} = \text{COGA}$$

If MAX - COGA - TS  $\leq$  0: (or overflow taken place)

TS = 0.9 (MAX - COGA)

DCOGA = TS

Return

TIMETHET This is "Time-Theta Subroutine", entered from "CDHMVR",  
"HOPALONG", "NC12J", "P34/P74C" (P35), "P38",  
RTNCONC = Return address and "RADUP"

Reset overflow indicator

TS<sub>r</sub> = RVEC

TS<sub>v</sub> = VVEC

Perform "PARAM"

If overflow has taken place:

Set bit 4(COGAFLAG) of FLAGWRD8 = 1

Proceed to "POODOO" (pattern 20607<sub>8</sub>)

TS = P

Perform "GETX"

Proceed to "COMMNOUT"

#### COMMNOUT

If bit 7(INFINFLG) of FLAGWRD8 = 1:

Proceed to "POODOO" (pattern 20607<sub>8</sub>)

Set bit 4(COGAFLAG) of FLAGWRD8 = 0

Perform "DELTIME"

If bit 9(RVSW) of FLAGWRD7 = 0: (means new R, V desired)

Perform "NEWSTATE"

Proceed to address specified by RTNCONC

NEWSTATE This is "State Vector Subroutine"

TS<sub>r</sub> = (R1 - XSQCpXI<sub>p</sub>) UR<sub>1</sub> + (T - 1dROOTMU X<sup>3</sup> SpXI<sub>p</sub>) WEC

TS = |TS<sub>r</sub>| (TS<sub>r</sub> at top of push-down list)

Proceed to "LAMENTER"

LAMENTER (Entered with TS set to radius magnitude)

R2 = TS

$$TS = \frac{ROOTMU \times}{R2} (XI \text{ SpXIp} - 1) \text{ UR1}$$

$TS_v = TS + (1 - XSQCpXIp / R2) \text{ VVEC}$  (same formulation as on page CONC-4, but different quantities enter calculation)  
Return ( $TS_v$  in MPAC)

TIMERAD This is "Time-Radius Subroutine". It is not referenced.

RTNCONC = Return address

Reset overflow indicator

$TS_r = RVEC$

$TS_v = VVEC$

Perform "PARAM"

If overflow has taken place:

Set bit 4(COGAFLAG) of FLAGWRD8 = 1

Proceed to "POODOO" (pattern 20607<sub>8</sub>)

$$PDECC = (1 - RdA) \text{ UR1} - \text{COGA} \sqrt{P (K_{2b6} - RdA)} \text{ U2} \quad \text{(computed B7, then made B3)}$$

If all components of PDECC  $\leq 2^{-18}$ , or if overflow has taken place:

Set bit 3(NOSOLNSW) of FLAGWRD5 = 1

Proceed to "POODOO" (pattern 20607<sub>8</sub>)

$$\text{COSF} = ((P \text{ R1} / \text{RDESIRED}) - 1) / |PDECC|$$

$$TS = 1 - \text{COSF}^2$$

If  $TS \leq 0$ , or if overflow has taken place:

$$\text{COSF} = 1 \text{ sgn COSF}$$

Set bit 5(APSESW) of FLAGWRD8 = 1 (trajectory does not pass through RDESIRED)

TS = 0

Skip next 2 lines

$$TS = \sqrt{TS} \text{ sgn SGNRDOT}$$

Set bit 5(APSESW) of FLAGWRD8 = 0

$PDUR_2 = \cos F \text{ unit} PDECC + TS (UN * \text{unit} PDECC)$

$CSTH = PDUR_2 \cdot UR_1$ , with magnitude limited less than 1

$SNTH = (UR_1 * PDUR_2) \cdot UN$

$TS = P$

Perform "GETX"

Set bit 3(NOSOLNSW) of FLAGWRD5 = 0

Proceed to "COMMNOUT"

APSIDES                      This is "Apsides Subroutine"

RTNCONC = Return address

Reset overflow indicator

$TS_r = RVEC$

$TS_v = VVEC$

Perform "PARAM"

Reset overflow indicator

$ECC = \sqrt{1 - P \text{ RdA}}$  (note that "SR30.1" formed absolute value  
before taking square root)

$TS_{rp} = R1 \text{ P} / (1 + ECC)$  (at top of push-down list, in OD)

$TS_{ra} = 2(R1 / \text{RdA}) - TS_{rp}$  (left in MPAC)

If  $TS_{ra} < 0$ , or if overflow has taken place:

$TS_{ra} = +MAX$

Proceed to address specified by RTNCONC (TS scaling same as RVEC)

PARAM                      (Entered with  $TS_r$  in OD and  $TS_v$  in MPAC)

RTNPRM = Return address                      Start of "Conic Parameters Subroutine"

Set bit 10(NORMSW) of FLAGWRD7 = 0

Set bit 4(COGAFLAG) of FLAGWRD8 = 0

$GEOMSGN = 27777_8$  (a positive quantity)

$TS_{r1} = TS_r$

$TS_2 = TS_v$

Perform "GEOM"

$\text{COGA} = \text{TS}_{\cos} / \text{TS}$  (TS left with sine of angle) (overflow if  $2^5$  or more)

$\text{ldMU} = \text{K}_{\text{mutab}_2}$

$\text{ROOTMU} = \text{K}_{\text{mutab}_4}$

$\text{ldROOTMU} = \text{K}_{\text{mutab}_6}$

$\text{TS}_1 = \text{ldMU R1 MAGVEC2}^2$  ( $\text{MAGVEC2} = |\text{TS}_v|$  from "GEOM")

$\text{RdA} = \text{K}_{2b6} - \text{TS}_1$

$\text{P} = \text{TS}_1 \text{TS}^2$  (TS from "GEOM" with sine of angle)

Proceed to address specified by RTNPRM

GEOM (Entered with  $\text{TS}_{r1}$  at top of pushdown list and  $\text{TS}_2$  in MPAC)

$\text{U}_2 = \text{unitTS}_2$  Start of "Geometric Parameters Subroutine"

$\text{MAGVEC2} = |\text{TS}_2|$

$\text{UR}_1 = \text{unitTS}_{r1}$

$\text{TS}_{\cos} = \text{UR}_1 \cdot \text{U}_2$  (left at top of push-down list)

$\text{R1} = |\text{TS}_{r1}|$

$\text{TS} = \text{UR}_1 * \text{U}_2$

If bit 10(NORMSW) of FLAGWRD7 = 0:

$\text{UN} = \text{unitTS sgn GEOMSGN}$

If overflow has taken place: (e.g.  $\text{TS}$  components all  $\leq 2^{-20}$ )

$\text{UN} = (\pm 1, \frac{1}{2} \text{TS}_y, \frac{1}{2} \text{TS}_z)$  (not meaningful)

$\text{TS} = |\text{TS}| \text{sgn GEOMSGN}$  (sine of angle, in MPAC)

Return

GETX (Entered with TS set to P, in MPAC)

$\text{X2} = 3$  (to be used as counter) Start of "Universal Variable Subroutine"

Set bit 1(360SW) of FLAGWRD8 = 0



$TS_1 = SNTH / (1 - CSTH)$  (scaled B5)

If overflow has taken place (e.g.  $TS_1$  magnitude  $\geq 2^5$ ):

If  $TS_1 < 0$ :

Set bit 1(36OSW) of FLAGWRD8 = 1

Proceed to "INVRSEQN"

$PDWVAL = \sqrt{TS} (TS_1 - COGA)$  (scaled B5)

If overflow has taken place (e.g.  $PDWVAL$  magnitude  $\geq 2^5$ ):

If  $PDWVAL < 0$ :

Set bit 1(36OSW) of FLAGWRD8 = 1

Proceed to "INVRSEQN"

Proceed to "WLOOP"

#### WLOOP

$TS_1 = PDWVAL^2$  (triple precision)

$TS_2 = RdA + TS_1$

If  $TS_2 < 0$ :

Reset overflow indicator

Set bit 7(INFINFLG) of FLAGWRD8 = 1

Return

$TS = PDWVAL + \sqrt{TS_2}$  (square root rounded to double precision)

If overflow has taken place: (e.g.  $TS$  magnitude  $\geq 2^5$ )

$X2 = 3$

If  $TS < 0$ :

Set bit 1(36OSW) of FLAGWRD8 = 1

Proceed to "INVRSEQN"

If  $X2 \geq 1$ :

$X2 = X2 - 1$

$PDWVAL = TS$

Proceed to "WLOOP"

TS = 1 / TS (scaled B2)

If overflow (e.g. TS magnitude  $\geq 2^2$ ):

Reset overflow indicator

Set bit 7(INFINFLG) of FLAGWRD8 = 1

Return

Proceed to "POLYCOEF"

#### INVRSEQN

$$PDW2 = \sqrt[3]{\frac{SNTH}{(1 + CSTH - SNTH \text{ COGA})}}$$

PDW3 = 1

Proceed to "1/WLOOP"

#### 1/WLOOP

TS<sub>1</sub> = PDW3<sup>2</sup> (triple precision)

TS<sub>2</sub> = RdA PDW2<sup>2</sup> (triple precision)

TS<sub>3</sub> = TS<sub>1</sub> + TS<sub>2</sub>

If TS<sub>3</sub> < 0:

Reset overflow indicator

Set bit 7(INFINFLG) of FLAGWRD8 = 1

Return

$$TS = PDW3 + \sqrt{TS_3}$$

If X2  $\geq$  1:

X2 = X2 - 1

PDW3 = TS

Proceed to "1/WLOOP"

TS = PDW2 / TS (scaled B2)

Proceed to "POLYCOEF"

# POLYCOEF

If  $TS \leq 0$ :

Reset overflow indicator

Set bit 7(INFINFLG) of FLAGWRD8 = 1

Return

PDAVAL = TS

$PDB = RdA \cdot PDAVAL^2$

$$FCTN = K_{db0} + K_{db1} PDB + K_{db2} PDB^2 + K_{db3} PDB^3 + K_{db4} PDB^4 + K_{db5} PDB^5 + K_{db6} PDB^6$$

$PDXN = 16 PDAVAL FCTN$

If bit 1(36OSW) of FLAGWRD8 = 1:

If  $RdA \leq 0$ :

Reset overflow indicator

Set bit 7(INFINFLG) of FLAGWRD8 = 1

Return

$PDXN = K_{2pisc} / \sqrt{RdA} - PDXN$

$XI = RdA \cdot PDXN^2$

$X = PDXN \sqrt{R1}$

$PDXSQ = X^2$

$KEPC1 = \sqrt{P R1} \cdot COGA$

$KEPC2 = 1 - RdA$

Set bit 7(INFINFLG) of FLAGWRD8 = 0

Return (to routine calling "GETX", of course)

## Quantities in Computations

See also list of major variables and list of routines

- ldMU: Value of the reciprocal of  $\mu$  of primary body, selected from  $K_{mutab}$ , scale factor B-34, and stored in push-down list location 14D, program notation "1/MU".
- ldROOTMU: Value of square root of ldMU of primary body, selected from  $K_{mutab}$ , scale factor B-17, and stored in push-down list location 18D, program notation "1/ROOTMU".
- lmCSTH: Value of  $(1 - CSTH)$ , scale factor B2, computed in "LAMBERT".
- ALPHA: Value of reciprocal of semi-major axis computed in "KEPLERN", scale factor B-22, units of meters<sup>-1</sup>, stored in push-down list location 8D.
- COGA: Value of cotangent of flight path angle, scale factor B5 (measured from vertical). "PARAM" computes it as angle between incoming position and velocity arguments, with overflow indicator set if an overflow here (corresponding to an angle less than about 1° 47.5', or of course closer to 180° than about 178° 12.5'). Notation also "INDEP".
- COSF: Value of cosine of the true anomaly at the desired radius in "TIMERAD", scale factor B1, stored in push-down list location 24D.
- CSTH: Value of cosine of the true anomaly difference or of the angle between present and desired position vectors, scale factor B1.
- CSTHmRHO: Value of difference between CSTH and PDLAMBDA computed in "LAMBERT", scale factor B7, program notation "CSTH-RHO".
- CXI: Value of special transcendental function computed in "DELTIME" from a series expansion of XI, scale factor B4, not explicitly stored but shown for clarity.
- DCOGA: Change in the independent variable COGA in Lambert iteration loop, scale factor B5. Program notation for the cell is also "DELINDEP", and it is stored in push-down list location 12D.
- DELT: Difference between desired time and derived time in "KEPLOOP", scale factor B28, units centi-seconds.
- DELX: Difference between new and old values of XKEP computed in "DXCOMP", scale factor B17. Updated in "KEPLOOP".
- ECC: Value of eccentricity computed in "APSIDES", scale factor B3.
- EPSILONL: Value of Lambert convergence limit computed in "LAMBERT", scale factor B28, units centi-seconds.

EPSILON<sub>T</sub>: Value of Kepler routine convergence limit computed in "DXCOMP", scale factor B28, units centi-seconds.

FCTN: Value of polynomial expansion function computed in "POLYCOEF", scale factor B1.

GEOMSGN: Single precision sign information for the sine of the true anomaly difference, used in "GEOM" (set to a positive value in "PARAM" before performing "GEOM"). It is a Lambert input. See SGNRDOT.

ITERCTR: Single precision iteration counter used to protect Kepler and Lambert iterations from excessive loops, scale factor B14. It is initialized to 20 in "KEPLERN" and "LAMBERT". In push-down cell 22D.

K<sub>1b10</sub>: Constant, program notation "D1/1024", scale factor B10, value  $1 \times 2^{-10}$ , corresponding to an equation value of 1.

K<sub>1cs</sub>: Constant, program notation "ONEBIT", scale factor B28, units centi-seconds. Value is  $1 \times 2^{-28}$ , corresponding to 1 centi-second.

K<sub>2b6</sub>: Constant, program notation "D1/32", scale factor B6, value  $1 \times 2^{-5}$ . Value corresponds to  $2 \times 2^{-6}$ , where first term is equation value and second is scale factor.

K<sub>2pisg</sub>: Constant, program notation "2PISC", scale factor B6, value  $6.2831853 \times 2^{-6}$ . Value corresponds to  $2\pi$ , with  $\pi = 3.14159265$ .

K<sub>beel9</sub>: Constant, program notation "BEE19", scale factor B0, value  $1 \times 2^{-19}$ , giving the Lambert time error ratio (used in "LAMBERT" to multiply TDESIRED to get EPSILONL).

K<sub>bee22</sub>: Constant, program notation "BEE22", scale factor B0, value  $1 \times 2^{-22}$ , giving the Kepler time error ratio (used in "DXCOMP" to multiply TAUORB to get EPSILON<sub>T</sub>).

K<sub>ci</sub> (i = 0-9): Set of constants used in "DELTIME" to evaluate CXI, scale factor B(4-6i), program notation of K<sub>c0</sub> "DELTIME +29". Values are:

<u>i</u>	<u>Stored Value</u>	<u>True Value</u>	<u>Infinite Series Value</u>
0	0.031250001	0.5 (in memory)	$(1/2!) = 0.5$
1	-0.166666719	-0.041666680	$-(1/4!) = -0.041666667$
2	0.355555413	1.38888833E-3	$(1/6!) = 1.38888889E-3$
3	-0.406347410	-2.48014777E-5	$-(1/8!) = -2.48015873E-5$
4	0.288962094	2.75575727E-7	$(1/10!) = 2.75573192E-7$
5	-0.140117894	-2.08791932E-9	$-(1/12!) = -2.08767570E-9$
6	0.049247387	1.14663008E-11	$(1/14!) = 1.14707456E-11$
7	-0.013081923	-4.75917586E-14	$-(1/16!) = -4.77947733E-14$
8	0.002806389	1.59524745E-16	$(1/18!) = 1.56192070E-16$
9	-0.000529414	-4.70214090E-19	$-(1/20!) = -4.11031762E-19$



$K_{\text{coglom}}$ : Constant, program notation "COGLOLIM", scale factor B5, value -0.999511597. Magnitude of true value corresponds to about 31.9843711, so chosen that  $(1 + \text{COGA}^2)$ , if  $\text{COGA} = K_{\text{coglom}}$ , will be less than  $2^{10}$ , thus avoiding overflow in the computation of RdA in "LAMBLOOP", where this term appears.

$K_{\text{cogupm}}$ : Constant, program notation "COGUPLIM", scale factor B5, value 0.999511597, the same decimal value (with opposite sign) as  $K_{\text{coglom}}$ .

$K_{\text{dbi}}$  ( $i = 0-6$ ): Set of constants used in "POLYCOEF" to evaluate FCTN, scale factor (of all) B1, program notation (of  $K_{\text{db0}}$ ) "POLYCOEF +10". Values are:

<u>i</u>	<u>Stored Value</u>	<u>True Value</u>	<u>Infinite Series Value</u>
0	0.5	1.0	1.0
1	-0.166666770	-0.33333354	$-(1/3) = -0.333333333$
2	0.100000392	0.200000784	$(1/5) = 0.2$
3	-0.071401086	-0.142802172	$-(1/7) = -0.142857143$
4	0.055503292	0.111006584	$(1/9) = 0.111111111$
5	-0.047264098	-0.094528196	$-(1/11) = -0.090909091$
6	0.040694204	0.081388408	$(1/13) = 0.076923076$

$K_{\text{epsx}}$ : Dummy constant used in "KEPLOOP" to indicate a value of one least increment for DELX, program value  $2^{-28}$ , scale factor B17. Equation value corresponds to  $2^{-11}$ . Could also be considered to be a program value of  $2^{-29}$  (depending on how rounding is interpreted), in which case requirement is for DELX to exceed the constant, and equation value would correspond to  $2^{-12}$ .

$K_{\text{m50sc}}$ : Constant, program notation "-50SC", scale factor B12, value  $-50 \times 2^{-12}$ , corresponding to an equation value of -50.

$K_{\text{mutab}_i}$  ( $i = 0-6$  in steps of 2): Set of constants providing functions of  $\mu$ .

<u>i</u>	<u>Stored Value</u>	<u>Scaling</u>	<u>Significance</u>
0	$3.986032\text{E}10 \times 2^{-36}$	B36	Earth $\mu$ , $\text{m}^3/\text{cs}^2$
2	$0.25087606\text{E}-10 \times 2^{34}$	B-34	Earth $1/\mu$ , $\text{cs}^2/\text{m}^3$
4	$1.99650495\text{E}5 \times 2^{-18}$	B18	Earth $\sqrt{\mu}$
6	$0.50087529\text{E}-5 \times 2^{17}$	B-17	Earth $1/\sqrt{\mu}$



$K_{si}$  ( $i = 0-9$ ): Set of constants used in "DELTIME" to evaluate  $SpX_{ip}$ , scale factor  $B(1-6i)$ , program notation of  $K_{s0}$  "DELTIME +3". Values are:

<u>i</u>	<u>Stored Value</u>	<u>True Value</u>	<u>Infinite Series Value</u>
0	0.083333334	0.166666668	$(1/3!) = 0.166666667$
1	-0.266666684	-8.33333387E-3	$-(1/5!) = -8.33333333E-3$
2	0.406349155	1.98412673E-4	$(1/7!) = 1.98412698E-4$
3	-0.361198675	-2.75572720E-6	$-(1/9!) = -2.75573192E-6$
4	0.210153242	2.50522187E-8	$(1/11!) = 2.50521084E-8$
5	-0.086221951	-1.60600899E-10	$-(1/13!) = -1.60590438E-10$
6	0.026268812	7.64523051E-13	$(1/15!) = 7.64716373E-13$
7	-0.006163316	-2.80275162E-15	$-(1/17!) = -2.81145725E-15$
8	0.001177342	8.36551806E-18	$(1/19!) = 8.22063525E-18$
9	-0.000199055	-2.20995444E-20	$-(1/21!) = -1.95729411E-20$

KEPC1: Value of coefficient in Kepler equation used in "DELTIME", scale factor  $B_{17}$ , stored in push-down list location 34D and computed in "KEPLERN" and "POLYCOEF".

KEPC2: Value of coefficient in Kepler equation used in "DELTIME", scale factor  $B_6$ , stored in push-down list location 36D and computed in "KEPLERN" and "POLYCOEF".

KEPRTN: Single precision return address from "KEPLERN" routine (preset in "KEPPREP").

MAGVEC2: Magnitude of "second" vector input to "GEOM": if entered from "PARAM", is velocity magnitude scaled  $B_7$ , units meters/centi-second; if entered from "LAMBERT", is magnitude of  $R2VEC$ , scaled  $B_{29}$ , units meters.

MAX: Value of maximum COGA computed in "LAMBERT" and subsequently updated, scale factor  $B_5$ . Stored in push-down list location 14D, with program notation also "COGAMAX".

MIN: Value of minimum allowable COGA computed in "LAMBERT" and subsequently updated, scale factor  $B_5$ . Stored in push-down list location 8D, with program notation also "COGAMIN".

P: Value of ratio of semi-latus rectum to magnitude of present ( $R_1$ ) position, scale factor  $B_4$ , computed in "PARAM" and "LAMBLOOP".

PDAVAL: Value of temporary quantity "a" computed in "GETX" computation and loaded in "POLYCOEF", scale factor  $B_2$ , loaded in push-down list location OD.

PDB: Value of temporary quantity "b" computed in "POLYCOEF", scale factor  $B_0$ , used as argument for computing FCTN.

PDECC: "A vector in direction of pericenter whose magnitude is equal to eccentricity," computed in "TIMERAD" initially with scale factor B7 and then rescaled to have scale factor B3. Its unit is stored in push-down list location OD, scale factor B1, while its magnitude is in 36D, scale factor B3.

PDLAMBDA: Value of ratio of magnitudes of R1VEC and R2VEC computed in "LAMBERT", scale factor B7, stored in push-down list location OD.

PDUR2: Unit vector in direction of final radius computed in "TIMERAD", scale factor B1, stored in push-down list location OD.

PDW2: Value of temporary quantity " $w_2$ " computed in "INVRSEQN", scale factor B-1, stored in push-down list location OD.

PDW3: Value of temporary quantity " $w_3$ " computed in "1/WLOOP", scale factor B4, stored in push-down list location 2D.

PDWAL: Value of temporary quantity " $w_1$ " computed in "WLOOP", scale factor B5, stored in push-down list location OD.

PDXN: Value of temporary quantity " $x_N$ " computed in "POLYCOEF", scale factor B6, stored in push-down list location OD. The factor of "16" in the equation comes from this scale factor: without the "16", scale factor would be B2 (scale of B6 required for subsequent computations, of course).

PDXSQ: Value of the square of the quantity "X", stored in push-down list location OD by "KEPLOOP" and "POLYCOEF", and used in "DELTIME". If X already normalized, scale factor B34: scaling information also contained in X1, which is used in "DELTIME" to achieve proper scaling of end results (not shown in programmed equations).

R<sub>rect</sub>: See Orbital Integration.

R1: Magnitude of "present" position vector (of R<sub>rect</sub> in "KEPLERN" and TS<sub>r1</sub> in "GEOM"), scale factor B29, units meters. Stored in push-down list location 32D.

R1VEC: Initial position vector used as input to Lambert routine (for point of maneuver initiation), scale factor B29, units meters.

R2: Magnitude of "desired" or final position vector, scale factor B29, units meters, loaded e.g. at start of "LAMENTER", where scaling information also in X1 (not shown).

R2VEC: Desired terminal (i.e., "target") position vector used as input to Lambert routine, scale factor B29, units meters.

RCV: See Orbital Integration. Low 5 bits 0 magnitude by "KEPCONVG".

RdA: Value of ratio of "present" (R1) position vector magnitude to semi-major axis, scale factor B6, computed in "LAMBLOOP" and "PARAM". Program notation is "R1A".

RDESIRED: Value of desired final radius used as input to "TIMERAD" routine, scale factor B29, units meters.

ROOTMU: Value of square root of  $\mu$  of primary body, selected from K<sub>mutab</sub>, scale factor B18, stored in push-down list location 16D.

RTNCONC: Single precision cell used to retain return address information from several conic routines. Program notations are "RTNAPSE", "RTNLAMB", "RTNTR", and "RTNTT".

RTNPRM: Single precision cell used to retain return address information from "PARAM".

RVEC: Vector position input to several of the conic routines, scale factor B29, units meters.

SGNRDOT: Single precision cell used double precision in "TIMERAD" to determine required sign of radial velocity at desired radius: if most significant half is non-zero, then least significant half would have no effect.

SNTH: Value of sine of true anomaly difference or of the angle between present and desired position vectors, scale factor B1.

SpXIp: Value of special transcendental function computed in "DELTIME" from a series expansion of XI, scale factor B1, program notation "S(XI)", stored in push-down list location 26D.

T: Value of time computed in "DELTIME" to go from present position to "desired" position, scale factor B28, units centi-seconds, stored in push-down list location 30D. It contains the result from the performance of "TIMERAD" or "TIMETHET" routines. Is also "DEP".

T<sub>c</sub>: See Orbital Integration. It is set in "KEPCONVG" to the value of T (less than one period).

TAUORB: See Orbital Integration. It is reduced in "PERIODCH" so as to be less than one period (for forward or backward integration).

TDESIRED: Value of desired transfer time (between R1VEC and R2VEC) used as input to "LAMBERT", scale factor B28, units centi-seconds.

TERRLAMB: Difference between desired and derived transfer time computed in "LAMBLOOP", scale factor B28, units centi-seconds; program notation also "DELDEP".

TPREV: Previous value of T in "LAMBLOOP", scale factor B28, units centi-seconds; program notation also "DEPREV".

TWEEKIT: Single precision quantity used to indicate required amount of modification on first iteration in "ITERATOR", scale factor B0, stored in push-down list location 40D. It is initialized to the appropriate value in "LAMBERT".

U2: Value of the unit of the vector whose magnitude is in MAGVEC2, scale factor B1, computed in "GEOM".

UN: Unit normal to the two vectors used as input to "GEOM", scale factor B1. If overflow is encountered in forming the unit vector, then the result is not meaningful. It can also be an input to "LAMBERT".

UR1: Value of the unit of the vector whose magnitude is in R1, scale factor B1, computed in "GEOM".

URRECT: Value of unit  $\underline{R}_{rect}$  computed in "KEPLERN", scale factor B1.

$\underline{V}_{rect}$ : See Orbital Integration.

$\underline{V}_{target}$ : See Rendezvous Computations.

VCV: See Orbital Integration. Low 8 bits have zero magnitude when computed in "KEPCONVG".

VTARGETAG: See Rendezvous Computations.

VVEC: Vector velocity input to several of the conic routines, scale factor B7, units meters/centi-second. Lambert output.

X: Storage for updated values of XKEP information in "KEPLOOP", scale factor B17, stored in push-down list location 20D. Computed in "KEPLOOP" and "POLYCOEF".

XI: Value of square of eccentric anomaly difference for ellipse, or negative of square of its hyperbolic analog for hyperbola, scaled B6. It is used as the independent variable in the "DELTIME" expansions, and is computed in "KEPLOOP" and "POLYCOEF", and stored in push-down list location 24D.

XKEP: See Orbital Integration. Program notation also "XPREV".

XKEPNEW: See Orbital Integration.

XMAX: Value of upper bound on X computed in "KEPLERN", scale factor B17, stored in push-down list location 10D.

XMIN: Value of lower bound on X computed in "PERIODCH", scale factor B17, stored in push-down list location 12D.

XSQCpXIp: Value of  $X^2$  times CXI computed in "DELTIME", scale factor B33, program notation "XSQC(XI)", stored in push-down list location 28D.

## Coordinate Transformations

### CALCGTA

$TS = \text{unit} (-X_{dc,z}, 0, X_{dc,x})$       Equivalent to  $\underline{X}_{dc} * \text{unit}\underline{Y}$

$SINTH = TS_x$

$COSTH = TS_z$

Perform "ARCTRIG"

$IGC = THETA$

$SINTH = X_{dc,y}$

$COSTH = X_{dc,x} TS_z - X_{dc,z} TS_x$

Perform "ARCTRIG"

$MGC = THETA$

$COSTH = TS \cdot \underline{Z}_{dc}$

$SINTH = TS \cdot \underline{Y}_{dc}$

Perform "ARCTRIG"

$OGC = THETA$

Return

ARCTRIG    Routine entered with SINTH and COSTH, scale factor B2. Return is with THETA, scale factor B0, units revolutions, in range  $\pm \frac{1}{2}$  (i.e.  $\pm 180^\circ$ ).

If  $|SINTH| - K_{qts4} \geq 0$ :

$THETA = (\cos^{-1} COSTH) \text{sgn } SINTH$

Return

$THETA = \sin^{-1} SINTH$

If  $COSTH \geq 0$ , Return

$THETA = \frac{1}{2} \text{sgn } SINTH - THETA$       (the  $\frac{1}{2}$  corresponds to  $180^\circ$ )

Return



### CALCGA

$$TS = \text{unit}(X_{dc} * Y_{sm})$$

$$COSTH = TS * Z_{dc}$$

$$SINTH = TS * Y_{dc}$$

Perform "ARCTRIG"

$$OGC = THETA$$

$$COSTH = (TS * X_{dc}) * Y_{sm}$$

$$SINTH = X_{dc} * Y_{sm}$$

Perform "ARCTRIG"

$$MGC = THETA$$

If bit 10(P5OFLAG) of FLAGWRDO = 0: (e.g. from "OAVECS" would have bit at 1)

If  $|MGC| - K_{pl66} \geq 0$ :

Perform "ALARM" (pattern 0401<sub>8</sub>)

Set bit 14(GLOKFAIL) of FLAGWRD3 = 1

Set bit 10(P5OFLAG) of FLAGWRDO = 0

$$COSTH = Z_{sm} * TS$$

$$SINTH = X_{sm} * TS$$

Perform "ARCTRIG"

$$IGC = THETA$$

THETAD = (OGC, IGC, MGC) converted to twos complement single precision, scale factor B-1 revolutions

Return

AXISGEN Entered with stars "A" and "B" in the "C" and "D" systems

$$STV = \text{unit}(\text{STARAC} * \text{STARBC})$$

$$STW = \text{STARAC} * STV$$

$$STV' = \text{unit}(\text{STARAD} * \text{STARBD})$$

$$STW' = \text{STARAD} * STV'$$

$$\underline{X}_{dc} = \text{unit} (\text{STARAD}_x \text{ STARAC} + \text{STV}'_x \text{ STV} + \text{STW}'_x \text{ STW})$$

$$\underline{Y}_{dc} = \text{unit} (\text{STARAD}_y \text{ STARAC} + \text{STV}'_y \text{ STV} + \text{STW}'_y \text{ STW})$$

$$\underline{Z}_{dc} = \text{unit} (\text{STARAD}_z \text{ STARAC} + \text{STV}'_z \text{ STV} + \text{STW}'_z \text{ STW})$$

$$[\text{STARAD}] = \begin{bmatrix} \underline{X}_{dc} \\ \underline{Y}_{dc} \\ \underline{Z}_{dc} \end{bmatrix} \quad \begin{array}{l} \text{(for "C" system in reference} \\ \text{coordinates and "D" system in} \\ \text{IMU coordinates, } \underline{X}_{dc} \text{ is } \underline{X}_{sm} \text{ in} \\ \text{reference coordinates)} \end{array}$$

Return

#### CDUTRIG

CDUSPOT = CDU (most significant halves, in y,z,x order)

CDUSPOT = CDUSPOT, converted to double precision ones complement,  
BO revolutions

(COSCDUY, COSCDUZ, COSCDUX) = cos CDUSPOT

(SINCDUY, SINCDUZ, SINCDUX) = sin CDUSPOT

Return

QUICTRIG (Entered from "CLOCKJOB" and "PREANGLE" with CDUSPOT set)

Inhibit interrupts

(SINCDUY, SINCDUZ, SINCDUX) =  $\sin_{sp} \text{ CDUSPOT}_{sp}$ , rescaled to scale  
factor B1

(COSCDUY, COSCDUZ, COSCDUX) =  $\cos_{sp} \text{ CDUSPOT}_{sp}$ , rescaled to scale  
factor B1

Release interrupts

Return

TRG\*NBSM      Entered with TS the vector to be rotated thru o,m,i

Perform "CDUTRIG"      (starting at second line)

$$TS_2 = TS_y \cos CDUX - TS_z \sin CDUX \quad (\text{tag to start here is "NBSM"})$$

$$TS_1 = TS_z \cos CDUX + TS_y \sin CDUX$$

$$TS = (TS_x, TS_2, TS_1)$$

$$TS_2 = TS_x \cos CDUZ - TS_y \sin CDUZ$$

$$TS_1 = TS_y \cos CDUZ + TS_x \sin CDUZ$$

$$TS = (TS_2, TS_1, TS_z)$$

$$TS_2 = TS_z \cos CDUY - TS_x \sin CDUY$$

$$TS_1 = TS_x \cos CDUY + TS_z \sin CDUY$$

$$TS = (TS_1, TS_y, TS_2)$$

Return

\*SMNB\*      Entered with TS the vector to be rotated thru i,m,o

$$TS_2 = TS_z \cos CDUY + TS_x \sin CDUY$$

$$TS_1 = TS_x \cos CDUY - TS_z \sin CDUY$$

$$TS = (TS_1, TS_y, TS_2)$$

$$TS_2 = TS_x \cos CDUZ + TS_y \sin CDUZ$$

$$TS_1 = TS_y \cos CDUZ - TS_x \sin CDUZ$$

$$TS = (TS_2, TS_1, TS_z)$$

$$TS_2 = TS_y \cos CDUX + TS_z \sin CDUX$$

$$TS_1 = TS_z \cos CDUX - TS_y \sin CDUX$$

$$TS = (TS_x, TS_2, TS_1)$$

Return

SMCDURES

$$DCDU_x = \frac{DTHETASM_x \cos CDUY - DTHETASM_z \sin CDUY}{\cos CDUZ}$$

$$DCDU_y = DTHETASM_y - DCDU_x \text{ SINCDUZ}$$

$$DCDU_z = DTHETASM_x \text{ SINCDUY} + DTHETASM_z \text{ COSCDUY}$$

Return

#### CALCSMSC

$$\underline{X}_{dc} = (\text{COSCDUY COSCDUZ}, \text{ SINCDUZ}, - \text{SINCDUY COSCDUZ})$$

$$\underline{Z}_{dc} = (\text{COSCDUY SINCDUZ SINCDUX} + \text{SINCDUY COSCDUX}, \\ - \text{COSCDUZ SINCDUX},$$

$$\text{COSCDUY COSCDUX} - \text{SINCDUY SINCDUZ SINCDUX})$$

$$\underline{Y}_{dc} = \underline{Z}_{dc} * \underline{X}_{dc}$$

Return

#### SXTNB

$$TS_1 = (E_{5-X1})_{sp}, \text{ converted to ones complement double precision,} \\ \text{scale factor B-2 revolutions (i.e. CDUT)}$$

$$TS_1 = TS_1 + K_{10dgm}, \text{ rescaled to B0 revolutions}$$

$$TS_2 = (E_{3-X1})_{sp}, \text{ converted to ones complement double precision,} \\ \text{scale factor B0 revolutions (i.e. CDUS)}$$

$$\underline{TS} = (\sin TS_1 \cos TS_2, \sin TS_1 \sin TS_2, \cos TS_1)$$

$$\underline{TS} = \begin{bmatrix} \text{NB1NB2} \end{bmatrix} \underline{TS}$$

Return (TS in 32D)

#### CALCSXA

$$\underline{TS} = \underline{STAR}$$

Perform "\*SMNB\*"

$$\underline{STAR} = \begin{bmatrix} \text{NB2NB1} \end{bmatrix} \underline{TS}$$

$$\underline{X}_{nbl} = \text{unit}\underline{X}$$

$$\underline{Y}_{nbl} = \text{unit}\underline{Y}$$

$$\underline{Z}_{nbl} = \text{unit}\underline{Z}$$

Proceed to "SXTANG1"

### SXTANG

$$TS_1 = [NB2NBL] (X_{dc,x}, Y_{dc,x}, Z_{dc,x})$$

$$TS_2 = [NB2NBL] (X_{dc,y}, Y_{dc,y}, Z_{dc,y})$$

$$TS_3 = [NB2NBL] (X_{dc,z}, Y_{dc,z}, Z_{dc,z})$$

$$X_{nbl} = (TS_{1,x}, TS_{2,x}, TS_{3,x})$$

$$Y_{nbl} = (TS_{1,y}, TS_{2,y}, TS_{3,y})$$

$$Z_{nbl} = (TS_{1,z}, TS_{2,z}, TS_{3,z})$$

Proceed to "SXTANG1"

### SXTANG1

$$TS_1 = \text{unit}(Z_{nbl} * STAR)$$

If overflow has taken place (i.e. unit vector poorly defined):

$$SAC = K_{270dg}$$

$$PAC = K_{20dgs}$$

Set bit 7(CULTFLAG) of FLAGWRD3 = 0

Return

$$SINTH = - TS_1 \cdot X_{nbl}$$

$$COSTH = TS_1 \cdot Y_{nbl}$$

Perform "ARCTRIG"

SAC = THETA, rescaled to B-1 revolutions single precision twos complement

$$TS = \cos^{-1} (STAR \cdot Z_{nbl})$$

If  $TS \geq 0$ : (as it should be, i.e. in range 0 - 180°)

If  $TS < 90^\circ$ :

TS = TS, rescaled to B-2 revolutions

$$TS = TS - K_{20dgsm}$$

PAC = TS, rescaled to B-3 revolutions single precision twos complement

Set bit 7(CULTFLAG) of FLAGWRD3 = 0

Return

Set bit 7(CULTFLAG) of FLAGWRD3 = 1

Return

#### EARROT1

$$R_{tes} = C_{unitw} * R_{ti}$$

$$R_{tnm} = R_{tes} * C_{unitw}$$

Proceed to "EARROT2"

#### EARROT2

$$TS = DTEAROT / K_{ldwie}$$

If  $|TS| \geq 1$ :

$$DTEAROT = DTEAROT - K_{ldwie} \operatorname{sgn} DTEAROT$$

Proceed to "EARROT2"

$$R_t = \operatorname{unit} (R_{ti} + (\cos TS - 1) R_{tnm} + (\sin TS) R_{tes})$$

Return

LSPOS Entered with TS = GET of information; exit  $TS_1$  in 2D, B1.

$$TS_1 = (TS + T_{eph}) / K_{cstoday} \quad (\text{time in days, B10, PDO})$$

$$TS_2 = C_{cmod} \sin (K_{komegac} TS_1 + C_{carg}) \quad (\text{B0 rev})$$

$$TS_3 = K_{losr} TS_1 + C_{loso} - TS_2 \quad (\text{B0 rev})$$

$$TS_1 = \operatorname{unit} (\cos TS_3, K_{konmat1} \sin TS_3, K_{konmat3} \sin TS_3)$$

Return

#### LAT-LONG

$$\text{ALPHAM} = |\text{ALPHAV}|$$

$$TS_1 = \text{ALPHAV}$$

$$TS_2 = TS \quad (\text{time argument when enter})$$

Perform "R-TO-RP"

$$\text{ALPHAV} = \operatorname{unit} TS$$

Perform "SETRE"



$$\text{COSTH} = K_{b2a2} \sqrt{\text{ALPHAV}_x^2 + \text{ALPHAV}_y^2}$$

$$\text{SINTH} = \text{ALPHAV}_z$$

Perform "ARCTAN"

$$\text{LAT} = \text{THETA}$$

$$\text{COSTH} = \text{ALPHAV}_x$$

$$\text{SINTH} = \text{ALPHAV}_y$$

Perform "ARCTAN"

$$\text{LONG} = \text{THETA}$$

$$\text{ALT} = \text{ALPHAM} - \text{ERADM}$$

Return

#### LALOTORV

$$\text{TS}_2 = \text{TS} \quad (\text{time argument when enter})$$

$$\text{ALPHAV} = \text{unit}(\cos \text{LONG} \cos \text{LAT}, \sin \text{LONG} \cos \text{LAT}, K_{b2a2} \sin \text{LAT})$$

$$\text{TS}_1 = \text{ALPHAV} \quad (\text{enter here from "Pl1" for } \underline{Z}_r)$$

Perform "SETRE"

Perform "RP-TO-R" ( $\text{TS}_2$  still set from entrance to "LALOTORV")

$$\text{ALPHAV} = \text{TS}$$

$$\text{ALPHAV} = (\text{ERADM} + \text{ALT}) \text{ALPHAV}$$

Return

#### GETERAD

$$\text{ERADM} = \sqrt{\frac{K_{b2xsc}}{1 - K_{ee} (1 - \text{ALPHAV}_z^2)}}$$

Return

### SETRE

$$\text{ERADM} = K_{\text{erad}}$$

If bit 13(ERADCOMP) of FLAGWRD1 = 1:

Perform "GETERAD" (writes over ERADM)

Return

### ARCTAN

$$\text{TS} = \text{SINTH}^2 + \text{COSTH}^2$$

If TS = 0:

$$\text{THETA} = 0$$

Return

$$\text{TS} = \text{SINTH} / \sqrt{\text{TS}}$$

If  $|\text{TS}| \geq 1$ :

$$\text{THETA} = \frac{1}{4} \text{sgn SINTH} \quad (\text{the } \frac{1}{4} \text{ is } 90^\circ)$$

Return

$$\text{THETA} = \sin^{-1} \text{TS}$$

If  $\text{COSTH} \geq 0$ , Return

$$\text{THETA} = \frac{1}{2} \text{sgn THETA} - \text{THETA} \quad (\text{the } \frac{1}{2} \text{ is } 180^\circ) \quad (0 \text{ considered negative})$$

Return

RP-TO-R Entered with  $\text{TS}_1$  in OD to be rotated and  $\text{TS}_2$  in 6D for the time.

Perform "EARTHMX"

$$\text{TS} = \text{TS}_1 \begin{bmatrix} \text{MMATRIX} \end{bmatrix}$$

$$\text{TS} = \text{TS} \begin{bmatrix} C_{\text{lmatrix}_0} \\ C_{\text{lmatrix}_3} \\ C_{\text{unitw}} \end{bmatrix}$$

Return

R-TO-RP Entered with same calling arguments as "RP-TO-R".

Perform "EARTHMX"

$$\underline{TS} = \begin{bmatrix} C_{lmatrix}_0 \\ C_{lmatrix}_3 \\ C_{unitw} \end{bmatrix} \underline{TS}_1$$

$$\underline{TS} = [\underline{MMATRIX}] \underline{TS}$$

Return

#### EARTHMX

$$AZ = K_{wearth} (T_{eph} + TS_2) + C_{azo}, \text{ modulo } 360^\circ \text{ (time argument modulo } 2^{33} \text{ cs, or about 994.2 days)}$$

$$[\underline{MMATRIX}] = \begin{bmatrix} \cos AZ & \sin AZ & 0 \\ -\sin AZ & \cos AZ & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Return

## Quantities in Computations

See also list of major variables and list of routines

ALPHAM: See Orbital Integration.

ALPHAV: Communication cell with coordinate transformation routines. If used with "GETERAD", contains a unit vector of position, scale factor B1; for latitude-longitude routines, contains position vector with scale factor B29, units meters, either when enter or when exit.

ALT: Value of altitude above appropriate base value, scale factor B29, units meters.

AZ: Value of rotation angle, scale factor B0, units revolutions, about the Z axis, modulo 1 revolution, computed in "EARTHMX". It is stored in push-down list location 18D (with notation "504AZ").

C<sub>azo</sub>: Erasable memory constant, program notation "AZO", scale factor B0, units revolutions. It is "the angle between the X-axis of the Basic Reference Coordinate System and the X-axis of the Earth-fixed Coordinate System (intersection of Greenwich meridian and equatorial plane of the earth)" at the beginning of the AGC year (i.e. when  $T_{eph} + T_{now}$  equals zero).

C<sub>carg</sub>: Erasable memory constant, program notation "CARG", scale factor B0, units revolutions, giving "mean anomaly of the earth-moon barycenter" at the beginning of the AGC year.

C<sub>cm0d</sub>: Erasable memory constant, program notation "CMOD", scale factor B-1, units revolutions, a "function of the mean eccentricity".

C<sub>lmatrix0</sub>: Erasable memory vector constant, program notation "LMATRIX", scale factor B0, giving first row of precession-nutation matrix.

C<sub>lmatrix3</sub>: Erasable memory vector constant, program notation "LMATRIX +6", scale factor B0, giving second row of precession-nutation matrix. The third row is C<sub>unitw</sub>.

C<sub>loso</sub>: Erasable memory constant, program notation "LOSO", scale factor B0, units revolutions, giving longitude of the sun at the beginning of the AGC year.

C<sub>unitw</sub>: See General Program Control.

CDUSPOT: Communication cells with coordinate transformation routines to specify (generally) the CDU angle information to be employed. When enter the routine, only the most significant half of each double precision word is used, considered to have a scale factor B-1, units revolutions in twos complement. Conventional order of storage of CDU angle information in CDUSPOT is y,z,x (i.e. inner, middle, outer respectively).

COSCDUX, COSCDUY, COSCDUZ: Value of cosines of information in CDUSPOT, scale factor B1 (in the same y,z,x order as CDUSPOT). If perform "QUICTRIG", a single-precision cosine routine is used, while if perform "CDUTRIG" the double precision interpretive routine is employed (taking about ten times more execution time). Updated once a second while "CLOCKJOB" is running (via "QUICTRIG" entrance), which includes the times that the TVC DAP is on.

COSTH: Value of argument cosine for "ARCTRIG", stored in push-down list location 16D, scale factor B2. Also used as communication cell with "ARCTAN", in which case only requirement is that scaling be the same as for SINTH to that routine.

DCDU: Incremental change in CDU angle due to DTHETASM (computed in "SMCDURES"), same scaling and units as DTHETASM.

DTEAROT: Value of earth rotation time for use in "EARROT2", scale factor B28, units centi-seconds.

DTHETASM: Communication cell with "SMCDURES" containing incremental angle changes measured about stable member axes. Scaling of DCDU (and units) will be the same as the scaling of this quantity: when entered from "CRS61.2A", scale factor B-3, units revolutions/decisecond.

ERADM: Value of radius of earth (from Fischer model or pad base), scale factor B29, units meters.

IGC: Value of inner gimbal angle, scale factor B0, units revolutions. The inner gimbal angle is identified with the Y axis.

$K_{ldwie}$ : Constant, program notation "1/WIE", scale factor B28, units centi-seconds. Value is  $86164.10 \times 2^{-28}$ , corresponding to 86164.10 seconds, the earth's rotation period (rounded to the nearest centi-second). The angular velocity corresponding to this time is about  $7.292115054E-5$  rad/sec (cf.  $K_{wearth}$ ).

$K_{10dgm}$ : Single precision constant, program notation "10DEGS-", scale factor B-2, units revolutions. Value is  $3600 \times 2^{-14}$ , corresponding to about  $19.7754^\circ$ . Notation arises from the fact that would be about  $10^\circ$  for the standard trunnion angle scaling of B-3 revolutions.

$K_{20dgs}$ : Constant, program notation "20DEGS-", scale factor B-3, units revolutions. Value is  $-7199 \times 2^{-14}$ , corresponding to  $-19.7754^\circ$  (since used as if it was twos complement, equivalent to  $-7200 \times 2^{-14}$  in ones complement).

$K_{20dgsM}$ : Constant, program notation "20DEG-", scale factor B-2, units revolutions. Value is  $3600 \times 2^{-14}$ , corresponding to about  $19.7754^\circ$ .



- $K_{270dg}$ : Constant, program notation "270DEG", scale factor B-1, units revolutions. Value is  $60000_8 00000_8$ , corresponding to  $-90^\circ$  in twos complement (or  $+ 270^\circ$ ).
- $K_{b2a2}$ : Constant, program notation "B2/A2", scale factor B1, value  $0.9933064884 \times 2^{-1}$ . Value corresponds to  $(6356.784/6378.166)^2 \times 2^{-1}$ , where terms in parentheses are the Fischer polar and equatorial earth radius expressed in kilometers (to the nearest meter).
- $K_{b2xsc}$ : Constant, program notation "B2XSC", scale factor B51, value  $0.0179450689$ . Value corresponds to  $(6356.784E3)^2 \times 2^{-51}$ , where first term is Fischer polar radius in meters (squared), and second is scale factor.
- $K_{cstoday}$ : Constant, program notation "CSTODAY", scale factor B32, units centi-seconds. Value is  $8640000 \times 2^{-32}$ , corresponding to  $86400 \times 100 \times 2^{-32}$ , where first term is number of seconds in (24-hour) day, second converts to centi-seconds, and third is scale factor.
- $K_{ee}$ : Constant, program notation "EE", scale factor B0, value  $6.6935116E-3$ . Value corresponds to  $(1 - K_{b2a2})$ , i.e. the square of Fischer model eccentricity (for  $K_{b2a2}$  values).
- $K_{erad}$ : Constant, program notation "ERAD", scale factor B29, units meters. Value is  $6373338 \times 2^{-29}$ , corresponding to 6373338 meters (same as  $K_{rpad}$ , see Burn Control).
- $K_{komegac}$ : Constant, program notation "KOMEGAC", scale factor B-1, units revolutions/day. Value is  $0.00273777859 \times 2^{-1}$ , corresponding to  $1.99096877E-7 \times (1/2\pi) \times 86400 \times 2^{-1}$ , where first term is rate for  $C_{cmod}$  term, units of rad/sec; second converts to revolutions; third converts to days; and fourth is scale factor.
- $K_{konmat1}$ : Constant, program notation "KONMAT1", scale factor B1, value  $0.91745773 \times 2^{-1}$ .
- $K_{konmat3}$ : Constant, program notation "KONMAT3", scale factor B1, value  $0.397833274 \times 2^{-1}$ .
- $K_{losr}$ : Constant, program notation "LOSR", scale factor B-4, units revolutions/day. Value is  $0.00273780317 \times 2^{-4}$ , corresponding to  $1.99098665E-7 \times (1/2\pi) \times 86400 \times 2^{-4}$ , where first term is sun rate, units of rad/sec; second converts to revolutions; third converts to days; and fourth is scale factor.
- $K_{pl66}$ : Constant, program notation ".166...", scale factor B0, units revolutions. Value is  $0.1666666667$ , corresponding to 60 degrees.
- $K_{qts4}$ : Constant, program notation "QTSN45", scale factor B2. Value is  $0.1768$ , corresponding to  $0.7072 \times 2^{-2}$ , where first term is  $\sin 45^\circ$  and second is scale factor.



$K_{\text{wearth}}$ : Constant, program notation "WEARTH", scale factor  $B-23$ , units revolutions/centi-second. Value is  $1.160576171E-7 \times 2^{23}$ , corresponding to  $7.29211514667E-5 \times (1/2\pi) \times 10^{-2} \times 2^{23}$ , where first term is rotation rate of earth in radians/second, second converts to revolutions, third converts to centi-seconds, and fourth is scale factor. Octal value is 37116, 32631, corresponding to about 0.9735618569, or a rate of about  $7.2921151577E-5$  rad/sec (a period of about 86164.0988 seconds).

LAT: Value of geodetic latitude, scale factor  $B0$ , units revolutions, positive for northerly latitudes.

LONG: Value of longitude, scale factor  $B0$ , units revolutions, positive for easterly longitudes.

MGC: Value of middle gimbal angle, scale factor  $B0$ , units revolutions. The middle gimbal angle is identified with the Z axis.

[MMATRIX]: Value of "modification matrix" computed in "EARTHMX", scale factor of elements  $B1$ , stored starting at push-down list location 20D.

[NB1NB2]: Matrix of constants, program notation "NB1NB2", scale factor  $B1$ . True values of elements (which are multiplied by  $2^{-1}$  for scale factor purposes) are:

$$\begin{bmatrix} 0.843175692 & 0 & 0.5376381241 \\ 0 & 1 & 0 \\ -0.5376381241 & 0 & 0.843175692 \end{bmatrix}$$

Numerical values in first row correspond to the cosine and sine of  $32^\circ 31' 23.19''$ .

[NB2NB1]: Matrix of constants, program notation "NB2NB1", scale factor  $B1$ . True values of elements (which are multiplied by  $2^{-1}$  for scale factor purposes) are:

$$\begin{bmatrix} 0.843175692 & 0 & -0.5376381241 \\ 0 & 1 & 0 \\ 0.5376381241 & 0 & 0.843175692 \end{bmatrix}$$

Matrix is the transpose of [NB1NB2].

OGC: Value of outer gimbal angle, scale factor  $B0$ , units revolutions. The outer gimbal angle is identified with the X axis. The quantity is stored such that  $OGC = (OGC, IGC, MGC)$ .

PAC: Required trunnion angle to have the star line-of-sight lie along STAR, scale factor B-3, units revolutions (double precision).

R<sub>t</sub>: Target vector output of "EARROT2", scale factor B1 (a unit vector).

R<sub>tes</sub>: Value of easterly component of R<sub>ti</sub>, scale factor B1 (dimensionless), computed in "EARROT1".

R<sub>ti</sub>: Communication cell with earth rotation routine, containing initial value of target vector (at DTEAROT = 0), scale factor B1 (a unit vector).

R<sub>tnm</sub>: Value of normal component of R<sub>ti</sub>, scale factor B1 (dimensionless), computed in "EARROT1".

SAC: Required optics shaft angle to have the star line-of-sight lie along STAR, scale factor B-1, units revolutions (double precision).

SINCDEX, SINCDUY, SINCDUZ: Values of sines of information in CDUSPOT, scale factor B1 (in the same y,z,x order as CDUSPOT). See COSCDUX.

SINTH: Value of argument sine for "ARCTRIG", stored in push-down list location 18D, scale factor B2. Also used as communication cell with "ARCTAN", in which case only requirement is that scaling be the same as for COSTH to that routine.

STAR: Unit vector, scale factor B1, specifying the (unit) vector direction to the appropriate star. It is in present stable member coordinates if "CALCSXA" is entered, and the coordinate system of [X<sub>dc</sub>] if "SXTANG" is entered.

STARAC: Unit vector, scale factor B1, used as communication cell with "AXISGEN", giving direction to "star A" in the "C" coordinate system. Stored in push-down list location 6D.

STARAD: Unit vector, scale factor B1, used as communication cell with "AXISGEN", giving direction to "star A" in the "D" coordinate system. Stored in cells starting at "STARAD".

[STARAD]: Matrix, elements with scale factor B1, the output of "AXISGEN". It defines "coordinate system D referred to coordinate system C".

STARBC: Unit vector, scale factor B1, used as communication cell with "AXISGEN", giving direction to "star B" in the "C" coordinate system. Stored in push-down list location 12D.

STARBD: Unit vector, scale factor B1, used as communication cell with "AXISGEN", giving direction to "star B" in the "D" coordinate system. Stored in cells starting at "STARAD +6".

STV, STV', STW, STW': Vectors derived in "AXISGEN", scale factor B1 (the primed set pertain to the "D" coordinate system). STV is stored in push-down list location 12D and STW in location 18D; STV' in the cell originally used for STARBD; and STW' in STARAD +12.

$T_{eph}$ : See Boost Computations.

THETA: Output angle from "ARCTRIG" or "ARCTAN", scale factor B0, units revolutions, stored in push-down list location 20D. It is in the range  $\pm \frac{1}{2}$  (i.e.  $\pm 180^\circ$ ).

$\underline{X}_{nbl}$ ,  $\underline{Y}_{nbl}$ ,  $\underline{Z}_{nbl}$ : Unit vectors, scale factor B1, used as input to "SXTANG1" to provide information on navigation base orientation in same coordinate system as STAR (when enter "SXTANG1").

## Data Input/Output

KEYRUPT1 (Entered after receipt of program interrupt #5, main DSKY)

$T_{st} = T_{now}$

$TS_1 = \text{bits } 5-1 \text{ (key code) of channel } 15$

Proceed to "KEYCOM"

KEYCOM

Set bit 15(DSKYFLAG) of FLAGWRD5 = 1 (means DSKY display output to be generated in "T4RUPT")

Establish "CHARIN" (priority  $30_8$ )

$MPAC+0_{LOCCTR} = TS_1$  (LOCCTR indexes MPAC+0 cell associated with the "CHARIN" job)

Resume

UPRUPT (Entered after receipt of program interrupt #7, uplink input)

$T_{st} = T_{now}$

Set  $TS = INLINK$  and  $INLINK = 0$

If bit 12(NODOPOL) of FLAGWRD1 = 0:

$UPSUM = UPSUM + TS \quad (\text{modulo } 2^{14})$

$UPSUM+1 = UPSUM+1 + 1 \quad (\text{modulo } 2^{14})$

Set bit 3(Uplink Activity) of channel 11 = 1

$TS_1 = \text{bits } 5-1 \text{ of } TS$

If bits 10-6 of  $TS$  are not the complement of  $TS_1$  (i.e. if these bits, shifted right 5 places, do not add with  $TS_1$  to equal  $11111_2$ ):

Set bit 4(UPLOCKFL) of FLAGWRD7 = 1

Resume

If bits 15-11 of  $TS$  are not equal to  $TS_1$  (i.e. if complement of these bits, shifted right 10 places, do not add with  $TS_1$  to equal  $11111_2$ ):

Set bit 4(UPLOCKFL) of FLAGWRD7 = 1

Resume

If  $TS_1 = 22_8$  (Error Reset):

Set bit 4(UPLOCKFL) of FLAGWRD7 = 0

Skip next line

If bit 4(UPLOCKFL) of FLAGWRD7 = 1, Resume

Proceed to second line of "KEYCOM"

## CHARIN

21d22REG = DSPLOCK

DSPLOCK = 1

If CADDRSTOR  $\neq$  0: (i.e. internal display waiting for response)

If MPAC+0  $\neq$  22<sub>8</sub>: (input character not Error Reset)

Set bit 5(Key Release) of channel 11 = 1

CHAR = MPAC+0 (loaded in "KEYCOM" with keyboard or uplink data)

If CHAR = 00<sub>8</sub>, proceed to "CHARALRM"

If CHAR = 01<sub>8</sub> - 07<sub>8</sub>, proceed to "NUM"

If CHAR = 10<sub>8</sub> - 11<sub>8</sub>:

If DSPCOUNT  $\leq$  -0, End of job

If bits 2-1 of DECBRNCH = 00<sub>2</sub>, proceed to "CHARALRM"

Proceed to "NUM"

If CHAR = 12<sub>8</sub> - 17<sub>8</sub>, proceed to "CHARALRM"

If CHAR = 20<sub>8</sub>:

CHAR = 00<sub>8</sub>

Proceed to "NUM"

If CHAR = 21<sub>8</sub>, proceed to "VERB"

If CHAR = 22<sub>8</sub>, proceed to "ERROR"

If CHAR = 23<sub>8</sub> - 30<sub>8</sub>, proceed to "CHARALRM"

If CHAR = 31<sub>8</sub>, proceed to "VBRELDSP"

If CHAR = 32<sub>8</sub>, proceed to "POSGN"

If CHAR = 33<sub>8</sub>, proceed to "NEGSGN"

If CHAR = 34<sub>8</sub>, proceed to "ENTER"

If CHAR = 35<sub>8</sub>, proceed to "CHARALRM"

If CHAR = 36<sub>8</sub>, proceed to "CLEAR"

If CHAR = 37<sub>8</sub>, proceed to "NOUN"

(Since CHAR is limited to 5 bits, this table includes all values)

## CHARALRM

Set bit 7(Operator Error) of channel 11 = 1

End of job

## NUM

If DSPCOUNT  $\leq$  -0, End of job

Perform "GETINREL"

If CLPASS  $>$  0, set CLPASS = +0

CODE = (bits 5-1 of  $K_{rtb\_CHAR}$ )

COUNT = DSPCOUNT

Perform "DSPIN"

If bits 2-1 of DECBRNCH = 00<sub>2</sub>:

TS = WDREG<sub>INREL</sub>, cycled left three places (bit 12 into bit 15 etc.)

WDREG<sub>INREL</sub> = TS + CHAR

If DSPCOUNT  $\neq$   $K_{crt\_INREL}$ :

DSPCOUNT = DSPCOUNT - 1, limited  $\geq$  +0

End of job

DSPCOUNT = - DSPCOUNT

End of job

WDREG<sub>INREL</sub> = (10) (WDREG<sub>INREL</sub>) + CHAR

If  $|WDREG_{INREL}| < 2^{14}$  (i.e. 16384):

If DSPCOUNT  $\neq$   $K_{crt\_INREL}$ :

DSPCOUNT = DSPCOUNT - 1, limited  $\geq$  +0

End of job

Perform "DECEND"

DSPCOUNT = - DSPCOUNT

End of job



## DECEND

If  $INREL - 1 \leq 0$ : (i.e. verb or noun)

DSPCOUNT = - DSPCOUNT

End of job

Perform "GETINREL" (Enter here from "ENTER")

$(WDREG_{INREL}, LPREG_{INREL}) = K_{dsc} WDREG_{INREL}$

If bits 2-1 of DECBRNCH =  $10_2$ :

$(WDREG_{INREL}, LPREG_{INREL}) = - (WDREG_{INREL}, LPREG_{INREL})$

Return

## GETINREL

Set INREL in accordance with the following table:

<u>DSPCOUNT</u>	<u>INREL</u>	<u>Function</u>
0-4	4	Register #3, Digits 5 - 1
5-9	3	Register #2, Digits 5 - 1
10-14	2	Register #1, Digits 5 - 1
15	$244_8$	Not assigned
16-17	1	Noun Register, Digits 2 - 1
18-19	0	Verb Register, Digits 2 - 1

Note that for the display registers, Digit 1 is the most significant and Digit 5 (or 2) is the least significant.

Return

## VERB

VERBREG = 0

DSPCOUNT = 19

Perform "2BLANK"

DECBRNCH =  $01_2$

REQRET = 0

ENTRET = "End of job"

End of job

### NOUN

NOUNREG = 0

DSPCOUNT = 17

Proceed to third line of "VERB"

### NEGSGN

If bits 2-1 of DECBRNCH  $\neq$  00<sub>2</sub>, End of job

If DSPCOUNT not equal to 4, 9, or 14, End of job

Perform "-ON"

DECBRNCH = DECBRNCH +  $K_{bt7_{INREL}}$  + 10<sub>2</sub>

If CLPASS  $\geq$  0, set CLPASS = +0

End of job

### -ON

Perform "GETINREL"

CODE = 0

TS =  $K_{sgt_{INREL}}$  + 1

Perform "11DSPIN" (turn off appropriate plus sign)

CODE = 2<sup>10</sup> (i.e. bit 11 = 1)

TS =  $K_{sgt_{INREL}}$

Perform "11DSPIN" (turn on appropriate minus sign)

Return

### POSGN

If bits 2-1 of DECBRNCH  $\neq$  00<sub>2</sub>, End of job

If DSPCOUNT not equal to 4, 9, or 14, End of job

Perform "+ON"

DECBRNCH = DECBRNCH +  $K_{bt7_{INREL}}$  + 01<sub>2</sub>

If CLPASS > 0, set CLPASS = +0

End of job

+ON

Perform "GETINREL"

CODE = 0

TS =  $K_{sgt_{INREL}}$

Perform "l1DSPIN" (turn off appropriate minus sign)

CODE =  $2^{10}$  (i.e. bit 11 = 1)

TS =  $K_{sgt_{INREL}} + 1$

Perform "l1DSPIN" (turn on appropriate plus sign)

Return

ERROR

DSPLOCK = 21d22REG

Inhibit interrupts

Set bit 10(Caution Reset) of channel 11 = 1

DSPTAB+11 = bits 6(Gimbal Lock) and 4(No Attitude) of DSPTAB+11, and  
flag for output at next opportunity

Set bits 13-11 (PIP2FLBT, DNIKFAIL, UPLKFAIL) of IMODES33 = 1

Set bit 10(PIPAFLBT) of IMODES30 = 1

Set bit 7(OCDFBIT) of OPTMODES = 1

Perform "C13STALL"

Set bit 10(Test DSKY Lights) of channel 13 = 0

Set bits 7(Operator Error) and 3(Uplink Activity) of channel 11 = 0

Set bit 12 of |DSPTAB| - |DSPTAB+10| = 1

Release interrupts

FAILREG+i = 0 (i = 0-1) (can zero FAILREG+2 by V23N9E E)

SFAIL = 0

End of job

## CLEAR

TS = |DSPCOUNT|

Set INREL (using TS as argument) per table in "GETINREL"

If CLPASS  $\ll$  +0:

    If INREL  $\ll$  2, End of job

    Perform "5BLANK" (starting at second line)

    CLPASS = CLPASS + 1

    End of job

INREL = INREL - 1, limited  $\gg$  0

If INREL  $\ll$  2, End of job

REQRET = REQRET + 3 ("backs up" data requests via "ABLOAD" or  
"ABCLOAD")

TS<sub>2</sub> = INREL

VERBREG = VERBREG - 1

Perform "UPDATVB"

INREL = TS<sub>2</sub>

Perform "5BLANK" (starting at second line)

CLPASS = CLPASS + 1

End of job

## 5BLANK

Perform "GETINREL"

WDREG<sub>INREL</sub> = 0

LPREG<sub>INREL</sub> = 0

CODE = 0

Set bits 1, 2, and K<sub>bt7</sub><sub>INREL</sub> of DECBRNCH = 0

COUNT = K<sub>snb</sub><sub>INREL-2</sub>

Perform "DSPIN"

DSPCOUNT = K<sub>dbk</sub><sub>INREL-2</sub>

Perform "2BLANK"

DSPCOUNT = DSPCOUNT - 2

Perform "2BLANK"

DSPCOUNT =  $K_{rdINREL-2}$

Return

#### 2BLANK

TS = integral part of ( $\frac{1}{2}$  DSPCOUNT)

DSPTAB<sub>TS</sub> =  $2^{11}$  (i.e. bit 12 = 1), and flag for output at next opportunity. If not already flagged, NOUT = NOUT + 1.

Return

#### ENTER

CLPASS = +0

ENTRET = "End of job"

If REQRET  $\geq$  +0, proceed to "ENTPASO"

If REQRET  $\neq$  - ("MMCHANG" + 1):

    If bits 2-1 of DECBRNCH  $\neq$  00<sub>2</sub>:

        If DSPCOUNT  $\geq$  +0: (i.e. 5 characters not yet input)

            Perform "DECEND" (starting at 4th line)

REQRET = - REQRET

Set bit 6(Flash) of channel 11 = 0

Proceed to address specified by REQRET

#### ENTPASO

DECBRNCH = 0

DSPCOUNT = -19

VERBSAVE = - VERBREG

If VERBREG -  $K_{owvb}$   $\geq$  0, proceed to "VERBFAN"

Proceed to "TESTNN"

# TESTNN

Perform "LODNNTAB"

If MIXBR = 2:

If NNADTEM = +0, proceed to "DSPALARM" (Tag here "MIXNOUN")

If VERBREG - 6 > 0, proceed to "VERBFAN"

Perform the following for  $\underline{I} = 3, 2, 1$ :

$TS_2 = IDAD\underline{I}TEM$

DECOUNT =  $\underline{I} - 1$

Perform "SFRUTMIX"

If  $TS_1$  indicates (via "DPTEST") double precision:

$TS_2 = TS_2 + 1$

$TS_2 = \text{bits } 11-1 \text{ of } TS_2$

$MIXTEMP_{\underline{I}-1} = E_{TS_2}$  ( $TS_2$  in erasable CADR form)

NOUNADD = "MIXTEMP"

Proceed to "VERBFAN"

If NNADTEM > 0:

NOUNCADR = NNADTEM

EBANK = bits 11-9 of NNADTEM

NOUNADD =  $1400_8 + \text{bits } 8-1 \text{ of NNADTEM}$

Proceed to "VERBFAN"

If NNADTEM = +0, proceed to "DSPALARM"

If NNADTEM = -0:

NOUNCADR = NOUNCADR + 1

EBANK = bits 11-9 of NOUNCADR

NOUNADD =  $1400_8 + \text{bits } 8-1 \text{ of NOUNCADR}$



(If NNADTEM = -0):

If VERBREG = 5, proceed to "VERBFAN"

DSPCOUNT = 4

TS = NOUNCADR

Perform "DSPOCTWD"

Proceed to "VERBFAN"

If NNADTEM < -0 (as it will):

CLPASS = -16383 (Tag here "REQADD")

If ENTRET = "End of job": (i.e. not from internal use)

Perform "REQDATZ" (upon return, proceed)

If DECBRNCH > 0, proceed to "ALMCYCLE"

DSPCOUNT = -19

If CADRSTOR  $\neq$  +0, set bit 6(Flash) of channel 11 = 1

NOUNCADR = ZREG

EBANK = bits 11-9 of NOUNCADR

NOUNADD =  $1400_8$  + bits 8-1 of NOUNCADR

Perform "LODNNTAB"

Proceed to "VERBFAN"

NOUNCADR = MPAC+2 (Tag here "INTMCTBS")

EBANK = bits 11-9 of NOUNCADR

NOUNADD =  $1400_8$  + bits 8-1 of NOUNCADR

If VERBREG = 5, proceed to "VERBFAN"

DSPCOUNT = 4

TS = NOUNCADR

Perform "DSPOCTWD"

Proceed to "VERBFAN"

#### REQDATX

DSPCOUNT = 14

Proceed to second line of "REQDATZ"

#### REQDATY

DSPCOUNT = 9

Proceed to second line of "REQDATZ"

#### REQDATZ

DSPCOUNT = 4

REQRET = - (Calling address +1) Calling address is that of "REQDATi",  
i = X, Y, or Z.

Perform "5BLANK"

Set bit 6(Flash) of channel 11 = 1

Proceed to address specified by ENTRET

#### UPDATNN

Perform "LODNNTAB"

If NNADTEM  $\geq$  +0:

NOUNCADR = NNADTEM

EBANK = bits 11-9 of NNADTEM

NOUNADD =  $1400_8$  + bits 8-1 of NNADTEM

DSPCOUNT = 17

TS = NOUNREG

Perform "DSPDECVN"

Return

#### UPDATVB

DSPCOUNT = 19

TS = VERBREG

Perform "DSPDECVN"

Return

DSPABC (verb 05)

TS = -2

Perform "COMPTST"

BUF+2 = - E<sub>NOUNADD+2</sub>

Proceed to "DSPAB"

DSPAB (verb 04)

TS = -1

Perform "COMPTST"

BUF+1 = - E<sub>NOUNADD+1</sub>

Proceed to "DSPA"

DSPA (verb 01)

TS = -0

Perform "DCOMPTST"

Perform "TSTFORDP"

BUF+0 = - E<sub>NOUNADD</sub>

Proceed to "DSPCOM2"

DSPB (verb 02)

TS = -1

Perform "DCOMPTST"

BUF+0 = - E<sub>NOUNADD+1</sub>

Proceed to "DSPCOM2"

DSPC (verb 03)

TS = -2

Perform "DCOMPTST"

BUF+0 = - E<sub>NOUNADD+2</sub>

Proceed to "DSPCOM2"

### DSPCOM2

$I = \text{VERBREG} - 3$ , limited  $\geq 0$

$\text{DSPCOUNT} = K_{\text{rd}I}$

$\text{TS} = -\text{BUF} + I$

Perform "DSPOCTWD"

If  $I = 0$ , proceed to address specified by ENTRET

$I = I - 1$

Proceed to second line of "DSPCOM2"

### COMPTST

If  $\text{MIXBR} = 1$ :

$\text{TS}_1 = \text{NNTYPTM}$

If  $\text{MIXBR} = 2$ :

$\text{TS}_1 = \text{NNADTEM}$

$\text{TS}_1 =$  bits 12-11 of  $\text{TS}_1$ , cycled left 5 places (placing in bits 2-1)

If  $\text{TS}_1 + \text{TS} < 0$ , proceed to "DSPALARM" (noun not have enough components)

Return

### DCOMPTST

If  $\text{MIXBR} = 1$ :

$\text{TS}_1 = \text{NNTYPTM}$

If  $\text{MIXBR} = 2$ :

$\text{TS}_1 = \text{NNADTEM}$

If bit 14 of  $\text{TS}_1 = 1$ :

Proceed to "DSPALARM" (noun is decimal-only)

Proceed to "COMPTST"

### TSTFORDP

If NNADTEM = -1:

TS = (bits 9-1 of NOUNCADR)

BUF+0 = - (contents of channel whose number is in TS)

Proceed to "DSPCOM2"

If MIXBR = 2, Return

TS<sub>1</sub> = bits 10-6 of NNTYPTM, cycled right 5 places (placing in bits 5-1)

If TS<sub>1</sub> indicates (via "DPTEST") double precision:

NOUNADD = NOUNADD + 1

Return

### DEC DSP (verb 06)

If MIXBR = 1:

TS<sub>1</sub> = NNTYPTM

If MIXBR = 2:

TS<sub>1</sub> = NNADTEM

TS<sub>1</sub> = bits 12-11 of TS<sub>1</sub>, cycled left 5 places (placing in bits 2-1)

DECOUNT = TS<sub>1</sub>

WDREG<sub>TS<sub>1</sub>+2</sub> = - E<sub>NOUNADD+TS<sub>1</sub></sub>

If TS<sub>1</sub> > 0, TS<sub>1</sub> = TS<sub>1</sub> - 1 and proceed to previous line

Proceed to "DSPDCPUT"

### DSPDCPUT

MPAC+i = 0 (i = 1,2)

DSPCOUNT = K<sub>rd</sub><sub>DECOUNT</sub>

MPAC<sub>sp</sub> = - WDREG<sub>DECOUNT+2</sub>

If MIXBR = 1:

TS = bits 5-1 of NNTYPTM

If MIXBR = 2:

TS = bits 5-1 of NNTYPTM if DECOUNT = 0

TS = bits 10-6 of NNTYPTM cycled right 5 places if DECOUNT = 1

TS = bits 15-11 of NNTYPTM cycled left 5 places if DECOUNT = 2

Perform "GTSFOUT"

If MIXBR = 1:

$TS_1$  = bits 10-6 of NNTYPTM, cycled right 5 places

If MIXBR = 2:

Perform "SFRUTMIX"

Proceed to "DEC DSP3"

#### DSPDCEND

Perform "DSPDECWD"

If DECOUNT = 0, proceed to address specified by ENTRET

DECOUNT = DECOUNT - 1

Proceed to "DSPDCPUT"

#### DPOUT

If MIXBR = 1:

TS = NOUNADD

If MIXBR = 2:

EBANK = bits 11-9 of  $IDAD_{ITEM_{DECOUNT+1}}$  (e.g.  $IDAD_{2TEM}$  for  $DECOUNT = 1$ )

TS =  $1400_8$  + bits 8-1 of  $IDAD_{ITEM_{DECOUNT+1}}$

$MPAC_{dp} = E_{TS_{dp}}$ , with sign agreement forced

$MPAC_{tp} = (MPAC_{dp})(SFTEMP_{1_{dp}})$

Return

#### DSPDPDEC (verb 07)

If MIXBR = 2, proceed to "DSPALARM"

$MPAC_{dp} = E_{NOUNADD_{dp}}$

DSPCOUNT = 14



MPAC+2 = 0

Force sign agreement of MPAC<sub>tp</sub>

Perform "DSP2DEC"

Proceed to address specified by ENTRET

ABCLOAD (verb 25)

TS = -2

Perform "COMPTST"

Perform "NOUNTEST"

VERBREG = 21

Perform "UPDATVB"

Perform "REQDATX" ; upon return, proceed

VERBREG = 22

Perform "UPDATVB"

Perform "REQDATY" ; upon return, proceed

VERBREG = 23

Perform "UPDATVB"

Perform "REQDATZ" ; upon return, proceed

TS = -6

Perform "ALLDC/OC"

Perform "LODNNTAB"

DECOUNT = 0

Perform "PUTCOM"

E<sub>NOUNADD</sub> = TS

DECOUNT = 1

Perform "PUTCOM"

E<sub>NOUNADD+1</sub> = TS

DECOUNT = 2

Perform "PUTCOM"

$E_{\text{NOUNADD}+2} = \text{TS}$

If  $\text{NOUNREG} \neq 07$ , proceed to "LOADLV" (bit change not specified)

Inhibit interrupts

$\text{TS} = \text{XREG} - 30_8$

If  $\text{TS} > 0$ :

$\text{EBANK} = (\text{bits } 11-9 \text{ of } \text{XREG})$

$\text{NOUNADD} = 1400_8 + (\text{bits } 8-1 \text{ of } \text{XREG})$

Channel 1 =  $E_{\text{NOUNADD}}$  (channel 1 is computer L register, see 3420.5-27)

$\text{XREG} = 0001_8$

$\text{TS} = 1$

If  $\text{TS} + 21_8 = 0$ : (i.e. original input was 07, channel 7)

Release interrupts

Proceed to "LOADLV"

If  $\text{ZREG} \leq 0$ :

Set those bits of channel  $\text{XREG} = 0$  that are 1 in YREG

If  $\text{ZREG} > 0$ :

Set those bits of channel  $\text{XREG} = 1$  that are 1 in YREG

If  $\text{XREG} = 1$ : (assumed to be due to setting above: if load  
(or if  $\leq 0$ )  $\text{XREG} = 1$  (or negative), no effect, since the  
contents of  $\text{NOUNADD} = \text{"XREG"}$ )

$E_{\text{NOUNADD}} = \text{Channel 1}$  (computer L register)

Release interrupts

Proceed to "LOADLV"

ABLOAD (verb 24)

$\text{TS} = -1$

Perform "COMPTST"

Perform "NOUNTEST"

$\text{VERBREG} = 21$

Perform "UPDATVB"

Perform "REQDATX" ; upon return, proceed

$\text{VERBREG} = 22$

Perform "UPDATVB"

Perform "REQDATY" ; upon return, proceed

TS = -5

Perform "ALLDC/OC"

Perform "LODNNTAB"

DECOUNT = 0

Perform "PUTCOM"

E<sub>NOUNADD</sub> = TS

DECOUNT = 1

Perform "PUTCOM"

E<sub>NOUNADD+1</sub> = TS

Proceed to "LOADLV"

ALOAD (verb 21)

Perform "REQDATX" ; upon return, proceed

Perform "LODNNTAB"

DECOUNT = 0

Perform "PUTCOM"

E<sub>NOUNADD</sub> = TS

Proceed to "LOADLV"

BLOAD (verb 22)

TS = -1

Perform "COMPTST"

CLPASS = -16383

Perform "REQDATY" ; upon return, proceed

Perform "LODNNTAB"

DECOUNT = 1

Perform "PUTCOM"

E<sub>NOUNADD+1</sub> = TS

Proceed to "LOADLV"

CLOAD (verb 23)

TS = -2

Perform "COMPTST"

CLPASS = -16383

Perform "REQDATZ" ; upon return, proceed

Perform "LODNNTAB"

DECOUNT = 2

Perform "PUTCOM"

$E_{\text{NOUNADD}+2} = \text{TS}$

Proceed to "LOADLV"

#### LOADLV

DECBRNCH = 0

LOADSTAT = -0

Perform "RELDSP"

DSPCOUNT = -19

Proceed to "RECALTST"

#### ALLDC/OC

$\text{TS}_1 = \text{DECBRNCH}$ , shifted right 2 places (7 if 3 decimal words,  
6 if 2 decimal words)

If  $\text{TS}_1 = 0$ , Return (all words octal)

If  $\text{TS}_1 + \text{TS} \neq 1$ , proceed to "ALMCYCLE"

Return

#### NOUNTEST (Entered from "ABCLOAD" and "ABLOAD" only)

If MIXBR = 1:

$\text{TS}_1 = \text{NNTYPTM}$

If MIXBR = 2:

$\text{TS}_1 = \text{NNADTEM}$

If bit 15 of  $\text{TS}_1 = 1$ , proceed to "DSPALARM" ("no-load" noun)

Return

#### SFRUTMIX

If DECOUNT = 0:

$\text{TS}_1 = \text{bits 5-1 of RUTMXTEM}$

If DECOUNT = 1:

TS<sub>1</sub> = bits 10-6 of RUTMXTEM, cycled right 5 places (to bits 5-1)

If DECOUNT = 2:

TS<sub>1</sub> = bits 15-11 of RUTMXTEM, cycled left 5 places (to bits 5-1)

Return

#### PUTCOM

DECRET = (calling address +1)

OVFINP = 0

MPAC<sub>dp</sub> = (WDREG<sub>DECOUNT+2</sub> , LPREG<sub>DECOUNT+2</sub>)

If MIXBR = 1:

EBANK = bits 11-9 of NOUNCADR (Tag here "PUTNORM")

NOUNADD = 1400<sub>8</sub> + bits 8-1 of NOUNCADR

If DECBRNCH > 0, proceed to "PUTDECSF"

If bit 14 of NNTYPTM = 1, proceed to "ALMCYCLE" (decimal only)

TS<sub>1</sub> = bits 10-6 of NNTYPTM, cycled right 5 places

If TS<sub>1</sub> indicates (via "DPTEST") double precision:

DECOUNT = 0

NOUNADD = NOUNADD + 1

DECOUNT = NOUNADD + DECOUNT

E<sub>DECOUNT-1</sub> = 0 (most significant half of word)

TS = MPAC+0

Proceed to address specified by DECRET

If NNADTEM = -1:

If NOUNCADR ≠ 7: (note that no operator indication of ignored input is provided)

TS = (bits 9-1 of NOUNCADR) (note that ought not to = 7)

Set (channel whose number is in TS) = MPAC+0

Proceed to "LOADLV"

(If MIXBR = 1):

TS = MPAC+0

Proceed to address specified by DECRET

NOUNCADR = bits 11-1 of IDADITEM<sub>DECOUNT+1</sub>

EBANK = bits 11-9 of NOUNCADR

NOUNADD =  $1400_8$  + bits 8-1 of NOUNCADR - DECOUNT

If DECBRNCH > 0, proceed to "PUTDECSF"

If bit 14 of NNADTEM = 1, proceed to "ALMCYCLE" (decimal only)

Perform "SFRUTMIX"

If TS<sub>1</sub> indicates (via "DPTEST") double precision:

NOUNADD = NOUNADD + 1

DECOUNT = DECOUNT + NOUNADD

$E_{DECOUNT-1} = 0$  (most significant half of word)

TS = MPAC+0

Proceed to address specified by DECRET

TS = MPAC+0

Proceed to address specified by DECRET

#### PUTDECSF

If MIXBR = 1:

TS = bits 5-1 of NNTYPTM

If MIXBR = 2:

TS = bits 5-1 of NNTYPTM if DECOUNT = 0

TS = bits 10-6 of NNTYPTM cycled right 5 places if DECOUNT = 1

TS = bits 15-11 of NNTYPTM cycled left 5 places if DECOUNT = 2

Perform "GTSFIN"



If MIXBR = 1:

$TS_1$  = bits 10-6 of NNTYPTM, cycled right 5 places

If MIXBR = 2:

Perform "SFRUTMIX"

Proceed to "PUTDCSF2"

#### BINROUND

$MPAC_{dp} = MPAC_{dp} + MPAC+1$  set OVFINP to 1 sgn MPAC if overflow

If OVFINP non-zero, proceed to "ALMCYCLE"

$TS = MPAC+0$

Proceed to address specified by DECRET

#### DPINSF

$MPAC_{tp} = (MPAC_{dp})(SFTEMP1_{dp})$

$MPAC_{tp} = MPAC_{tp} + MPAC+2$  set OVFINP to 1 sgn MPAC if overflow

If OVFINP non-zero, proceed to "ALMCYCLE"

If MIXBR = 1:

$E_{NOUNADD+1} = MPAC+1$

If MIXBR = 2:

$E_{NOUNADD+DECOUNT+1} = MPAC+1$

$TS = MPAC+0$

Proceed to address specified by DECRET

#### MONITOR (verbs 11-17)

$TS_1$  = bits 13-1 of NOUNCADR

If ENTRET = "End of job" (i.e. externally initiated monitor):

Set bit 14 of  $TS_1 = 1$

MONSAVE2 = 0

$TS_2$  = bits 7-1 of VERBREG (cycled left 7 places) + NOUNREG

DSPLOCK = 0

If CADRSTOR = 0:

    If DSPLIST = 0:

        Set bit 5(Key Release) of channel 11 = 0 (would have been  
turned on if new monitor  
keyed in over old one)

Inhibit interrupts

If MONSAVE = 0:

    Call "MONREQ" in 0.01 second

MONSAVE =  $TS_2$

MONSAVE1 =  $TS_1$

Release interrupts

Proceed to address specified by ENTRET

#### MONREQ

$T_{st} = T_{now}$

If MONSAVE1  $\ll$  -0:

    MONSAVE = 0

    MONSAVE1 = 0

    End of task

Call "MONREQ" in  $K_{mond}$  seconds

Establish "MONDO" (priority  $30_8$ )

End of task

#### MONDO

If MONSAVE1  $\ll$  -0, End of job

If DSPLOCK  $\succ$  0:

    Set bit 5(Key Release) of channel 11 = 1

    End of job

NOUNREG = bits 7-1 of MONSAVE

Perform "UPDATNN"

VERBREG = (bits 14-8 of MONSAVE, shifted right 7 places) - 10

ENTRET = "PASTEVB"

MPAC+2 = bits 13-1 of MONSAVE1

Proceed to "TESTNN" (will return via ENTRET to "PASTEVB")

#### PASTEVB

TS = bits 14-8 of MONSAVE2

If TS = 0: (non-zero if "MARKMONR" display interface routine used)

TS = MONSAVE

VERBREG = (bits 14-8 of TS, shifted right 7 places)

Perform "UPDATVB"

REQRET = +0

TS = MONSAVE2

Perform "BLANKSUB": if return to calling address +1, proceed  
otherwise, proceed

End of job

#### DSPFMEM (verb 27)

DSPCOUNT = 14

If bits 15-11 of NOUNCADR  $\leq 27_8$ : (fixed memory address in bank  
27<sub>8</sub> or below; NOUNCADR loaded in  
TS = E<sub>NOUNCADR</sub> FCADR format by using V27 N01)

If bits 15-11 of NOUNCADR  $> 27_8$ : (fixed memory address in bank 30<sub>8</sub>  
or above; DSPTEML+2 loaded by  
Channel 7 = bits 7-5 of using V23 N25 before V27 N01 that  
DSPTEML+2 is used to load FCADR NOUNCADR)

TS = E<sub>NOUNCADR</sub> (address information also determined by Channel  
7 contents)

Restore Channel 7 to previous value

Perform "DSPOCTWD"

End of job

#### DSPSIGN

If MPAC+0  $\geq +0$ :

Perform "+ON"

If MPAC+0  $\leq -0$ :

MPAC<sub>dp</sub> = - MPAC<sub>dp</sub>

Perform "-ON"

Return

#### DSPDECWD

Perform "DSPSIGN"

$MPAC_{dp} = MPAC_{dp} + K_{dcrd}$       If overflow,  $MPAC_{dp} = +MAX$

WDCNT = 4

Proceed to "DSPDCWD1"

#### DSPDCWD1

$MPAC_{tp} = K_{bncn} MPAC_{tp}$

CODE = (bits 5-1 of  $K_{rtb_{MPAC+0}}$ )

COUNT = DSPCOUNT

$MPAC_{tp} = 2^{14} MPAC_{tp}$       (overflow ignored: this preserves the fractional part of the multiplication in the first line)

DSPCOUNT = COUNT - 1

Perform "DSPIN"

If WDCNT > 0:

WDCNT = WDCNT - 1

Proceed to "DSPDCWD1"

DSPCOUNT = -19

Return

#### DSPDC2NR

Perform "DSPSIGN"

WDCNT = 1

Proceed to "DSPDCWD1"

#### DSP2DEC

CODE = 0

TS = 3

Perform "11DSPIN"      (turn off R2 minus sign)

TS = 4

Perform "LLDSPIN" (turn off R2 plus sign)

Perform "DSPSIGN"

WDCNT = 9

Proceed to "DSPDCWD1"

#### DSPDECVN

MPAC+0 =  $K_{vncon}$  TS

MPAC+1 = 0

WDCNT = 1

Proceed to "DSPDCWD1"

#### DSPOCTWD

TS<sub>1</sub> = TS

DSPCOUNT = DSPCOUNT +  $2^{13}$  (sets bit 14 = 1)

WDCNT = 4

Proceed to "WDAGAIN"

#### WDAGAIN

Cycle TS<sub>1</sub> left 3 places and store in TS<sub>1</sub> (bits 15-13 become 3-1)

TS = bits 3-1 of TS<sub>1</sub>

CODE = (bits 5-1 of  $K_{rtb_{TS}}$ )

COUNT = DSPCOUNT

DSPCOUNT = DSPCOUNT - 1

Perform "DSPIN"

If WDCNT > 0:

WDCNT = WDCNT - 1

Proceed to "WDAGAIN"

DSPCOUNT = -19

Return

### DSPIN

DSREL = bits 5-2 of COUNT, shifted right 1 place (i.e.  $\frac{1}{2}$  COUNT)

If bit 1 of COUNT = 0:

COUNT = 0

Proceed to "DSPIN1"

Shift CODE left 5 places (bits 5-1 become bits 10-6)

If bit 14 of COUNT = 1:

COUNT = 3

Proceed to "DSPIN1"

COUNT = 1

Proceed to "DSPIN1"

### DSPIN1

Inhibit interrupts

TS = | DSPTAB<sub>DSREL</sub> |

If COUNT = 0, set bits 5-1 of TS = bits 5-1 of CODE

If COUNT = 1, set bits 10-6 of TS = bits 10-6 of CODE

If COUNT = 2, set bit 11 of TS = bit 11 of CODE

If COUNT = 3, set bits 11-6 of TS = bits 11-6 of CODE

If TS  $\neq$  | DSPTAB<sub>DSREL</sub> | :

DSPTAB<sub>DSREL</sub> = TS, flagged for output at next opportunity

If not already flagged, NOUT = NOUT + 1

Release interrupts

Return

### 11DSPIN

DSREL = TS

COUNT = 2

Proceed to "DSPIN1"



### DSPALARM

If ENTRET = "NVSUBEND": (i.e. internal request)  
    Proceed to "POODOO" (pattern 21501<sub>8</sub>)  
If ENTRET = "PASTEVB": (i.e. from "MONDO")  
    MONSAVE1 = 40000<sub>8</sub> (sets bit 15 to 1)  
    Set bit 7(Operator Error) of channel 11 = 1  
    Proceed to "PASTEVB" (to put monitor verb back in VERBREG)  
Set bit 7(Operator Error) of channel 11 = 1  
End of job

### ALMCYCLE

Set bit 7(Operator Error) of channel 11 = 1  
REQRET = - VERBSAVE (sets to a positive number)  
VERBREG = - VERBSAVE  
Perform "UPDATVB"  
Proceed to "ENTER"

### MMCHANG (verb 37)

Perform "REQMM"  
If DSPCOUNT  $\neq$  -16, proceed to "ALMCYCLE" (2 digits not entered)  
MPAC+0 = NOUNREG  
NOUNREG = 0  
DSPCOUNT = 17  
Perform "2BLANK"  
DSPCOUNT = -19  
TS = MPAC+0  
Proceed to "V37"

### REQMM

REQRET = - (Calling address + 1)  
DSPCOUNT = 17  
NOUNREG = 0

Perform "2BLANK"

Set bit 6(Flash) of channel 11 = 1

DECBRNCH =  $01_2$

Proceed to address specified by ENTRET

VBRQEXEC (verb 30)

Perform "RELDSP"

If N26dPRI = 0:

Proceed to "DSPALARM"

If bit 1 of N26dPRI = 1:

Establish a job with 2CADR information in N26d2CAD<sub>dp</sub>, with a VAC area, and with priority given by bits 14-10<sub>dp</sub> of N26dPRI

If bit 1 of N26dPRI = 0:

Establish a job with 2CADR information in N26d2CAD<sub>dp</sub>, with no VAC area, and with priority given by bits 14-10<sub>dp</sub> of N26dPRI

End of job

VBRQWAIT (verb 31)

Perform "RELDSP"

If N26dPRI = 0:

Proceed to "DSPALARM"

Call task in N26dPRI centi-seconds with starting address given by N26d2CAD<sub>dp</sub> (in 2CADR format)

End of job

VBRESEQ (verb 32)

LOADSTAT = -0

Proceed to second line of "VBPROC"

VBPROC (verb 33; also entered from "PROCKEY")

LOADSTAT = -1

MONSAVE1 = 40000<sub>g</sub> (sets bit 15 to 1 and bit 14 to 0)

Perform "RELDSP"

Set bit 6(Flash) of channel 11 = 0

Proceed to "RECALTST"

VBTERM (verb 34)

LOADSTAT = -2

Proceed to second line of "VBPROC"

PROCKEY (Established by "PROCEEDE")

If  $|22 - \text{VERBREG}| \leq 1$ : (includes case of e.g. V21 PRO,  
but not V25 PRO)

Proceed to "CHARALRM"

REQRET = 0

DSPCOUNT = -19

Proceed to "VBPROC"

VBRELDSP (Entered from "CHARIN")

Set bit 3(Uplink Activity) of channel 11 = 0

If 21d22REG > 0:

If bit 14 of MONSAVE1 = 1: (external monitor)

DSPLOCK = 0 (causes monitor to be started again  
due to "MONDO" logic)

If CADRSTOR = 0:

If DSPLIST = 0:

Set bit 5 (Key Release) of channel 11 = 0

End of job

Perform "RELDSP"

If CADRSTOR = +0, End of job

Proceed to "PINBRNCH"

NVSUB

$TS_1 = 0$  (L-register, MONSAVE2 information)

Proceed to "NVMONOPT"

NVMONOPT (Entered with TS, verb-noun pattern or blanking code, in A)

If bit 14 of MONSAVE1 = 1, or if DSPLOCK > 0:

Return (to calling address +1, indicating system "busy")

MONSAVE2 =  $TS_1$

MONSAVE1 = 40000<sub>8</sub> (sets bit 15 to 1)

ENTRET = "NVSUBEND"

If  $|TS| = 0$ , proceed to "DSPALARM"

If TS < 0: (error if -5 or less)

i = |TS| + 6

Set DSPTAB+0 - DSPTAB+i = 2<sup>11</sup> (i.e. blank) and flag for output  
at next opportunity. If individual DSPTAB cell not already  
flagged, NOUT = NOUT + 1

If TS ≤ -3:

VERBREG = 0

If TS ≤ -2:

NOUNREG = 0

CLPASS = 0

DSPCOUNT = -19

Set bit 6(Flash) of channel 11 = 0

REQRET = 0

Proceed to "NVSUBEND"

TS<sub>1</sub> = bits 7-1 of TS (original input to routine, giving noun)

TS<sub>2</sub> = bits 14-8 of TS, shifted right 7 places (giving verb)

If TS<sub>1</sub> = 0:

VERBREG = TS<sub>2</sub>

Perform "UPDATVB"

REQRET = 0

Proceed to "NVSUBEND"

If TS<sub>2</sub> = 0:

NOUNREG = TS<sub>1</sub>

Perform "UPDATNN"

Proceed to "NVSUBEND"

TS<sub>3</sub> = MPAC+2 (MPAC+2 contains machine address for address-to-  
be specified nouns)

VERBREG = TS<sub>2</sub>

Perform "UPDATVB"

NOUNREG = TS<sub>1</sub>

Perform "UPDATNN"

LOADSTAT = +0

CLPASS = 0

REQRET = 0

MPAC+2 = TS<sub>3</sub>

Proceed to "ENTPASO"

#### NVSUBEND

Proceed to ("NVMONOPT" calling address +2)

#### JAMTERM

REQRET = 34

VERBREG = 34

DSPCOUNT = -19

Proceed to "VBTERM"

#### BLANKSUB (Entered with TS, bits 1-3, set to blank R1 - R3 respectively)

If bit 14 of MONSAVEL = 1, or if DSPLOCK > 0:

Return (to calling address +1, indicating system "busy")

TS<sub>4</sub> = (bits 3-1 of TS) (TS<sub>4</sub> in NVTEMP cell)

If TS<sub>4</sub> = 0:

Return (to calling address +2)

TS<sub>3</sub> = DSPCOUNT (in BUF+2 cell)

If bit 1 of TS<sub>4</sub> = 1:

DSPCOUNT = 14

Perform "5BLANK"

If bit 2 of TS<sub>4</sub> = 1:

DSPCOUNT = 9

Perform "5BLANK"

If bit 3 of  $TS_4 = 1$ :

DSPCOUNT = 4

Perform "5BLANK"

DSPCOUNT =  $TS_3$

Return (to "BLANKSUB" calling address +2)

NEWMODEX (Entered with  $TS$  = new major mode pattern)

MODREG =  $TS$

Establish "DSPMMJB" (priority  $30_8$ )

Return

DSPMMJB

$TS_2$  = DSPCOUNT

DSPCOUNT = 21

If MODREG = -0, perform "2BLANK"

If MODREG  $\neq$  +0:

$TS$  = MODREG

Perform "DSPDECVN"

DSPCOUNT =  $TS_2$

End of job

RECALTST

If CADRSTOR = 0, End of job

Inhibit interrupts

Awaken job with starting address id = CADRSTOR

CADRSTOR = 0

$MPAC_{LOCCTR_{dp}} = (VERBREG, NOUNREG)$

Release interrupts

Perform "RELDSP"

End of job



RELDSP

Set bit 14 of MONSAVEL = 0

If DSPLIST  $\neq$  0:

Awaken job with starting address id = DSPLIST

DSPLIST = 0

Set bit 5(Key Release) of channel 11 = 0

DSPLOCK = 0

Return

## Quantities in Computations

See also list of major variables and list of routines

21d22REG: Temporary storage cell for previous value of DSPLOCK, used to avoid changing DSPLOCK if an Error Reset code is received, etc.

BUF: Temporary storage cell.

CADRSTOR: Single precision cell indicating, if non-zero, that a job has been put to sleep in "ENDIDLE". The contents of the cell are the address to which return is made: the cell is loaded with "IDLERET1" in "ENDIDLE" when putting the job to sleep.

CHAR: Input character (from DSKY or uplink), scale factor B14 (located in bits 5-1).

CLPASS: Control cell for "CLEAR": if negative, a single-component load verb (22 or 23) is being used, so Clear inputs must not "back up" (successive clears should not reset previously loaded registers). If CLPASS = +0, this indicates that the register indicated by DSPCOUNT has had a character (number or sign) loaded in it, so that register would be cleared by an input of the Clear code. Scaled B14.

CODE: Cell used to contain the required relay pattern code for loading DSPTAB<sub>I</sub> and causing the proper character to be displayed by DSKY, B14.

COUNT: Cell used in "DSPIN" to contain identification of the output character to be changed. Scale factor B14.

DECBRNCH: Cell used to control the processing of data inputs: the least significant two bits are 00<sub>2</sub> for an octal number, 01<sub>2</sub> for a positive decimal number (including verb, noun, and program number inputs), and 10<sub>2</sub> for a negative number. Bits 3-5 respectively are one if registers 3-1 are loaded with decimal information, and are used to check that multiple-component load verbs employ either all-decimal or all-octal input.

DECOUNT: Cell used for indexing purposes to control processing of multiple-component verbs, scale factor B14. Has values 0-2 for first, second, and third component respectively.

DECRET: Exit address after performance of "PUTCOM".

DSPCOUNT: Display counter, scale factor B14, identifying digit of data input expected next, as defined by table in "GETINREL". If is negative, no numerical inputs are accepted. It is set to -19 to inhibit all numerical inputs (and the Clear code) after the completion of certain functions. Values of 20-21 are for display of program number (MODREG).

DSPLIST: Single precision erasable memory cell indicating, if non-zero, that a job has been put to sleep in "NVSUBUSY" while waiting for the display system to be released by the crew. The contents of the cell are the starting address of the job.

DSPLOCK: Word used to indicate, if non-zero, that some external use (DSKY or uplink) of the display system has been made, and therefore internal attempts (via "NVMONOPT") to use the display system are delayed. When the display system is "released" (via "RELDSP"), DSPLOCK is set 0. Scale factor is B14.

DSREL: Indexing cell used to specify the DSPTAB cell to be modified in "DSPIN", scale factor B14.

EBANK: Hardware erasable memory bank register, required together with another address (see NOUNADD) to specify completely the memory location of an arbitrary erasable memory cell. See 3420.5-27 for more details.

ENTRET: Return address information from display program. Set to "End of job" for keyboard or uplink input, or to the address of additional computations if from "NVMONOPT" or "MONDO". Tags ENTRET and ENTEXIT both used in the program for this cell.

FAILREG: See General Program Control.

IDADITEM: See Noun Definitions.

IMODES30, IMODES33: See IMU Computations.

INLINK: Computer special erasable memory cell (address 0045<sub>8</sub>) containing complete word (received one bit at a time) from the uplink receiver. Sixteen bits are received by the computer (additional bits are used by the receiver for spacecraft and system identification), with the first bit a binary one. When this first bit is shifted out of bit 15 (the most significant end) of the INLINK register, program interrupt #7 is generated, causing "UPRUPT" to be entered.

INREL: Indexing cell used to select internal computer word for the display system, set in accordance with table in "GETINREL", B14.

$K_{\text{bncn}}$ : Single precision constant, program notation "BINCON", scale factor B14, value 10. Leaves integral part of product (in "DSPDCWD1") in MPAC+0.

$K_{\text{bt}7I}$ : Table of individual bits, giving values equal to bit 5, bit 4, and bit 3 for  $I = 2-4$  respectively.

$K_{\text{crt}I}$ : Table of constants, program notation "CRITCON", giving value of DSPCOUNT for last character in verb, noun, and R1-R3 registers. Values (for  $I = 0-4$  respectively) are 18, 16, 10, 5, and 0, B14.

$K_{dbk_I}$ : Table of constants, program notation "DOUBLK", values 13, 9, and 3 for  $I = 0-2$  respectively, scale factor B14.

$K_{dcrd}$ : Constant, program notation "DECROUND -1", scale factor B0, octal value 00000<sub>8</sub> 02476<sub>8</sub>. Considering quantity involved to have a maximum value (for display purposes) of 0.99999, the constant has an equivalent value of  $2476_8 \times 2^{-28}$ , or about  $\frac{1}{2} \times 10^{-5}$ .

$K_{dsc}$ : Constant, program notation "DECON", scale factor B-14, value  $10^{-5} \times 2^{14}$ . Constant performs the function of converting integer decimal input to a fraction (it is not used for verb, noun, or program number inputs).

$K_{mond}$ : Single precision constant, program notation "MONDEL", scale factor B14, units centi-seconds. Value is 00144<sub>8</sub>, corresponding to a decimal value of 100 or a monitor period of one second.

$K_{owvb}$ : Single precision constant, program notation "LOWVERB", scale factor B14, value 28. VERBREG contents of this value or above bypass "TESTNN".

$K_{rd_I}$ : Table of constants, program notation "R1D1", values 14, 9, and 4 for  $I = 0-2$  respectively, scale factor B14.

$K_{rtb_I}$ : Table of constants information, program notation "RELTAB", containing in least significant 5 bits the required value of CODE for the digit value specified by I:

<u>I</u>	<u>Value</u>	<u>I</u>	<u>Value</u>
0	25 <sub>8</sub>	5	36 <sub>8</sub>
1	03 <sub>8</sub>	6	34 <sub>8</sub>
2	31 <sub>8</sub>	7	23 <sub>8</sub>
3	33 <sub>8</sub>	8	35 <sub>8</sub>
4	17 <sub>8</sub>	9	37 <sub>8</sub>

See 3420.5-27 for more details.

$K_{sgt_I}$ : Table of constants, program notation "SGNTAB", values 5, 3, and 0 for  $I = 2-4$  respectively, scale factor B14.

$K_{snb_I}$ : Table of constants, program notation "SINBLANK", values 14, 5, and 4 for  $I = 0-2$  respectively, scale factor B14.

$K_{vncon}$ : Single precision constant, program notation "VNDSPCON", scale factor B0, octal value 00244<sub>8</sub>, corresponding to  $164 \times 2^{-14}$ , or about 0.01. Used to convert two-digit integer information to a fraction for subsequent processing by "DSPDCWD1".

LOADSTAT: Cell used in "ENDRET" to determine the proper response if a job was put to sleep in "ENDIDLE". It is set to +0 if waiting for an input; -1 if a proceed input is received; -2 if a terminate input is received; -0 if an "otherwise" (data load, V32E, etc.) is received; and +1 in "ENDEXT" to indicate use for terminating extended verbs. Scale factor is B14.

LOCCTR: Indexing parameter used to locate cells within the particular job register set assigned to a job which has been established (or a job which has been awakened). See 3420.5-27 for more details.

LPREG<sub>I</sub>: Set of cells, defined only for I = 2-4 (XREGLP, YREGLP, and ZREGLP respectively), containing the least significant half of product for decimal input information.

MIXBR: See Noun Definitions.

MIXTEMP: Set of cells loaded if a display verb is used with the values of the memory cells specified by a "mixed" noun.

MONSAVE: Cell used to contain verb (bits 14-8) and noun (bits 7-1) information for monitor-type verbs. Cell is zero if no monitor function is being performed.

MONSAVE1: Cell used to contain in bits 13-1 the value of NOUNCADR for monitor verbs. Bit 15 (the sign) is set 1 if it is desired to terminate the monitor function, and bit 14 is set 1 if the monitor function was externally (DSKY or uplink) initiated.

MONSAVE2: Cell which can be set non-zero if internally initiated monitor started via "NVMONOPT", and used in "PASTEVB" to provide the capability for "pasting" a verb other than that used to generate display information, and to provide the capability for blanking a monitor display. Verb information in bits 14-8; blanking bits in bits 3-1 for R3, R2, and R1 respectively.

N26d2CAD: Pair of cells loaded by R2 and R3 of N26, used to specify in 2CADR format (see 3420.5-27) the starting address for the job/task to be initiated by V30E/V31E. Program notation "N26/2CAD".

N26dPRI: Single precision cell used to specify the priority for a V30 or (B14 cs) the required delay time for a V31 before N26d2CAD step is entered. It is loaded by R1 of N26. A value of zero is used to lock out both V30 and V31, to avoid erroneous actions if keyed in inadvertently. Program notation "N26/PRI".

NNADTEM, NNTYPEM: See Noun Definitions.

NOUNADD: Cell containing erasable memory address information derived from noun data. If bits 12 and 11 are zero and bits 10 and 9 are one (i.e. in range  $1400_8 - 1777_8$ ), then the contents of EBANK determine which "bank" of 256 erasable memory cells is selected, while the contents of NOUNADD determine which cell within the bank is referenced. If NOUNADD is less than  $1400_8$ , a unique cell is selected and EBANK is ignored. See 3420.5-27 for more details.

NOUNCADR: Cell containing the complete noun addressing information, obtained either from the noun tables or from an additional piece of input information. If the quantity is known to be in erasable memory, bits 11-9 are loaded into EBANK and bits 8-1 are added to  $1400_8$  to form NOUNADD.

NOUNREG: Cell containing (in bits 7-1) the decimal value of the noun, which specifies "to what". The verb indicates "do what". Scaled B14.

NOUT: See General Program Control.

OPTMODES: See Optics Computations.

OVFINP: Single precision cell, program notation "MPAC+6", used to retain the fact that overflow has taken place in the "Pinball" DSKY processing routines (mainly inputs). It is set 0 in "PUTCOM", B14.

REQRET: Cell controlling the interpretation of an "enter" command. If it is zero or any positive value, an initial pass for processing of the verb and noun information is performed; if negative, the cell is considered to contain the complement of the address to which return should be made when a word of data input has been completed.

RUTMXTEM: See Noun Definitions.

SFAIL: Cell used in computer self-check routine to contain return address from error subroutine (hence can be used to identify reason for failure of self-check). See Testing Routines. Scale factor B14.

SFTEMP1: See Noun Definitions.

T<sub>st</sub>: Value of T<sub>now</sub> sampled shortly after program interrupt acted upon (uplink, waitlist for monitor, or either keyboard). It can be displayed by noun 65. Scale factor B28, units centi-seconds.

UPSUM: Single precision cell, scale factor B14, used for uplink summing purposes (in "UPRUPT") if bit 12(NODOPO1) of FLAGWRD1 = 0. It can be used to check the validity of e.g. erasable memory programs loaded via uplink.

UPSUM+1: Single precision cell, scale factor B14, used to count the number of uplink interrupts processed if bit 12(NODOPO1) of FLAGWRD1 = 0. It is the cell after UPSUM in memory.

VERBREG: Cell containing (in bits 7-1) the value of the verb (c.f. NOUNREG). Scale factor is B14.

VERBSAVE: Cell loaded in "ENTPASO" with complement of VERBREG, used in "ALMCYCLE" to permit another attempt to be made to provide the required information. Loaded in "HMSIN" with - 25. Scaled B14.

WDCNT: Counter used in digit display routines to cause the proper number of digits to be displayed, scale factor B14.

WDREG<sub>I</sub>: Set of registers, identified as VERBREG, NOUNREG, XREG, YREG, and ZREG for I = 0-4 respectively.

XREG, YREG, ZREG: Cells containing information (with LPREG<sub>I</sub> if decimal) associated with display registers R1, R2, and R3 respectively. Noun table information for noun 07 is such as to assign the data loaded for that noun to these same cells, for processing at the end of "ABCLOAD".



Format for DSPTAB+0 - DSPTAB+10

<u>Cell</u>	<u>Bit 11</u>	<u>Bits 10-6</u>	<u>Bits 5-1</u>
DSPTAB+0	-R3S	R3D4	R3D5
DSPTAB+1	+R3S	R3D2	R3D3
DSPTAB+2		R2D5	R3D1
DSPTAB+3	-R2S	R2D3	R2D4
DSPTAB+4	+R2S	R2D1	R2D2
DSPTAB+5	-R1S	R1D4	R1D5
DSPTAB+6	+R1S	R1D2	R1D3
DSPTAB+7			R1D1
DSPTAB+8		ND1	ND2
DSPTAB+9		VD1	VD2
DSPTAB+10		MD1	MD2

For a positive DSPTAB cell, bit 12 is always 1. To flag a change, the complete word is complemented.

R1, R2, and R3 are the three digital display registers, with D5 the least significant digit and D1 the most significant. Each register has an associated sign bit, indicated by -RiS or +RiS. If no sign is to be indicated, neither sign bit is set. The codes for the individual numbers (blank and 0-9) appear on page DATA-37 (under K<sub>rtb</sub>).

The two-digit noun, verb, and mode (or program) registers are indicated by "N", "V", and "M" respectively. D2 is the least significant digit and D1 the most significant. The codes for the individual numbers are the same as for the digital display registers.

## Display Interface Routines

BLANKET (TS set when enter with bits to be set 1 in MPAC+4 of the display job: bits 3-1 = 1 to blank R3, R2, and R1 respectively. Tag is also "LINUS")

$TS = (-MPAC+4) \cap TS$  (sets those bits of  $TS = 1$  that are 0 in  $MPAC+4$  of present job and 1 in  $TS$ )

$MPAC+4_{MPAC+5} = MPAC+4_{MPAC+5} + TS$  ("GODSPRS+1" set  $MPAC+5$  of current job to LOCCTR of job established there)

Return

### CLEANDSP

$MPAC+1 = 0$  (causes "NVDSP" to establish "JAMTERM", returning to calling address +1 with blanking in "ENDRET")

$MPAC+4 = 20010_8$  (RESETREQ, FLREQ) (bit 14 means repeat of display flash requested)

Proceed to "GOFLASH2+1"

### CLEARMRK

$EXTVBACT = 0$

Set bit 1(XDSPFLAG) of  $FLAGWRD_4 = 0$

Return

CLOCPLAY Entered from "CLOCKJOB" for V97/V99 generation

$MPAC+1 = TS$

$MPAC+4 = 24030_8$  (RESETREQ, BURNREQ, PERFREQ, FLREQ)

Proceed to "GOFLASH2+1"

ENDEXT (Tags also "ENDEXTVB", "ENDMARK", and "TERMEXTV")

Perform "CLEARMRK"

$LOADSTAT = 1$

If bits 14(PRIODLE) and 13(NORMIDLE) of  $FLAGWRD_4 = 00_2$ :

Proceed to "NORMRET" (neither priority or normal display waiting for response)

Set bit 6(PINBRFLG) of  $FLAGWRD_4 = 1$

If bit 14(PRIODLE) of  $FLAGWRD_4 = 1$ : (priority display waiting for response)

Proceed to second line of "REDOPRIO"

Proceed to "PLAYJUM1"

EXDSPRET (TS set to verb-noun pattern)

MPAC+1 = TS

MPAC+4 = 40040<sub>8</sub> (MKEXREQ, DOTHNRET) (bit 6 means return request  
after display generated)

Proceed to "GOFLASH2+1"

GODSP (TS set to verb-noun pattern)

MPAC+1 = TS

MPAC+4 = 0

Proceed to "GOFLASH2+1"

GODSPR (TS set to verb-noun pattern)

MPAC+1 = TS

MPAC+4 = 0

TS<sub>4</sub> = 0 (Causes return to calling address +1)

Proceed to second line of "GODSPRS+1"

GODSPRET (TS set to verb-noun pattern)

MPAC+1 = TS

MPAC+4 = 00040<sub>8</sub> (DOTHNRET) (means return request after  
display generated)

Proceed to "GOFLASH2+1"

GOFLASH (TS set to verb-noun pattern)

MPAC+1 = TS

MPAC+4 = 00010<sub>8</sub> (FLREQ)

Proceed to "GOFLASH2+1"

GOFLASHR (TS set to verb-noun pattern)

MPAC+1 = TS

MPAC+4 = 00010<sub>8</sub> (FLREQ)

Proceed to "GODSPRS+1"

GOMARK2 (TS set to verb-noun pattern)

MPAC+1 = TS

MPAC+4 = 40030<sub>g</sub> (MKEXREQ, PERFREQ, FLREQ)

Proceed to "GOFLASH2+1"

GOMARK4 (TS set to verb-noun pattern)

MPAC+1 = TS

MPAC+4 = 40036<sub>g</sub> (MKEXREQ, PERFREQ, FLREQ, R3BLNK, R2BLNK)

Proceed to "GOFLASH2+1"

GOMARKFR (TS set to verb-noun pattern)

MPAC+1 = TS

MPAC+4 = 40010<sub>g</sub> (MKEXREQ, FLREQ)

Proceed to "GODSPRS+1"

GOPERF1 (TS set to octal checklist)

DSPTM1 = TS

MPAC+1 = 0125<sub>vn</sub>

MPAC+4 = 00036<sub>g</sub> (PERFREQ, FLREQ, R3BLNK, R2BLNK)

Proceed to "GOFLASH2+1"

GOPERF1R (TS set to octal checklist)

DSPTM1 = TS

MPAC+1 = 0125<sub>vn</sub>

MPAC+4 = 00036<sub>g</sub> (PERFREQ, FLREQ, R3BLNK, R2BLNK)

Proceed to "GODSPRS+1"

GOPERF2R (TS set to verb-noun pattern)

MPAC+1 = TS

MPAC+4 = 00030<sub>8</sub> (PERFREQ, FLREQ)

Proceed to "GODSPRS+1"

GOPERF4 (TS set to octal OPTION1; OPTION2 already loaded)

OPTION1 = TS

MPAC+1 = 0406<sub>vn</sub>

MPAC+4 = 00014<sub>8</sub> (FLREQ, R3BLNK)

Proceed to "GOFLASH2+1"

GOXDSPF (TS set to verb-noun pattern) Tag also "GOMARKF"

MPAC+1 = TS

MPAC+4 = 40010<sub>8</sub> (MKEXREQ, FLREQ)

Proceed to "GOFLASH2+1"

INITDSP (Entered if restart logic finds an x.1 restart phase setting, indicating a restart-protected normal display)

EBANK = (bits 11-9 of DSPFLG+2)

Channel 7 = (bits 7-5 of RESTREG)

TS = (bits 14-10 of RESTREG)

Change priority of present job to TS ("INITDSP" established with 14<sub>8</sub>)

TS = CADRFLSH+2 - 3 (normal calling sequence for display routines such that this returns to step loading TS with information before transfer to display routine)

Proceed to address specified by TS

KLEENEX

MPAC+1 = 0 (see note with "CLEANDSP")

MPAC+4 = 40010<sub>8</sub> (MKEXREQ, FLREQ)

Proceed to "GOFLASH2+1"

MARKMONR (TS set to verb-noun pattern)

MPAC+1 = TS

MPAC+4 = 40210<sub>8</sub> (MKEXREQ, DECMON, FLREQ)

Proceed to "GOFLASH2+1"

PINBRNCH Entered from "IDLERET1" if noun used with load verb not what was requested, and from "VBRELDSP" if key release button depressed with no external monitor suspended and with CADDRSTOR  $\neq$  0. Also from other places (e.g. "AIM/END").

Release interrupts (may have been inhibited before entrance)

MPAC+2 = MARK2PAC

If bits 15(MARKIDLE), 14(PRIOIDLE), and 13(NORMIDLE) of FLAGWRD<sub>4</sub> all = 0: (no display waiting for response)

TS = -3 (blank all display registers except program)

Perform "NVSUB": if busy, proceed; otherwise, proceed

End of job

If bit 15(MARKIDLE) of FLAGWRD<sub>4</sub> = 1:

Proceed to "MARKPLAY"

Set bit 6(PINBRFLG) of FLAGWRD<sub>4</sub> = 1

If bit 14(PRIOIDLE) of FLAGWRD<sub>4</sub> = 1: (priority display waiting for response)

Proceed to second line of "REDOPRIO"

Proceed to "PLAYJUM1"

PRIODSP (TS set to verb-noun pattern)

MPAC+1 = TS

MPAC+4 = 00110<sub>8</sub> (PRIOREQ, FLREQ)

Proceed to "GOFLASH2+1"

PRIODSPR (TS set to verb-noun pattern)

MPAC+1 = TS

MPAC+4 = 00110<sub>8</sub> (PRIOREQ, FLREQ)

Proceed to "GODSPRS+1"



REFLASH (TS set to verb-noun pattern)

MPAC+1 = TS

MPAC+4 = 20010<sub>8</sub> (RESETREQ, FIREQ) (bit 14 means repeat of display  
flash requested)

Proceed to "GOFLASH2+1"

REGODSP (TS set to verb-noun pattern)

MPAC+1 = TS

MPAC+4 = 20000<sub>8</sub> (RESETREQ)

Proceed to "GOFLASH2+1"

REGODSPR (TS set to verb-noun pattern)

MPAC+1 = TS

MPAC+4 = 20000<sub>8</sub> (RESETREQ)

TS<sub>4</sub> = 0 (Causes return to calling address +1)

Proceed to second line of "GODSPRS+1"

VNFLASH (TS set to verb-noun pattern)

MPAC+1 = TS

Increment calling address (i.e. QREG) by 1

Set bit 2(VNFLAG) of FLAGWRD<sub>4</sub> = 1

Proceed to second line of "GOFLASH"

VNFLASHR (TS set to verb-noun pattern)

MPAC+1 = TS

Increment calling address (i.e. QREG) by 1

Set bit 2(VNFLAG) of FLAGWRD<sub>4</sub> = 1

MPAC+4 = 00010<sub>8</sub> (FIREQ)

TS<sub>4</sub> = 0 (Causes return to calling address +"1" (actually 2  
due to QREG increment))

Proceed to second line of "GODSPRS+1"

GODSPRS+1 (Entered for those display interface routines, except "MARKMONR", whose tags end in "R", to cause "MAKEPLAY" to be established and then return to calling routine)

TS<sub>4</sub> = 3 (Causes return to calling address +4; bypassed if from "GODSPR", "REGODSPR", or "VNFLASHR")

If bit 4 of MPAC+4 = 1: (FLREQ)

Establish "MAKEPLAY" (with VAC area, and with priority and channel 7 setting same as current job)

If bit 4 of MPAC+4 = 0: (FLREQ)

Establish "MAKEPLAY" (with no VAC area, with priority same as current job, and channel 7 setting to 3)

Set bits 11-9 of MPAC+4 = EBANK

MPAC+3 = (calling address +1, in FCADR format, of display interface routine, where "calling address" is that of the routine's CADR, i.e. cell after TC BANKCALL, except for "VNFLASHR")

Set MPAC+1 through MPAC+4 of the "MAKEPLAY" job (determined by using LOCCTR for indexing) equal to MPAC+1 through MPAC+4 of the present job

MPAC+5 = LOCCTR (for use in "BLANKET": appropriate interrupt inhibits done)

Proceed to address specified by (MPAC+3 + TS<sub>4</sub>) (return to caller)

GOFLASH2+1 (Entered for those display interface routines whose tags do not end in "R", but which require display activity. "MARKMONR" an exception).

Set bits 11-9 of MPAC+4 = EBANK

MPAC+3 = (calling address +1, in FCADR format, of display interface routine: see explanation with "GODSPRS+1")

Proceed to "MAKEPLAY" (after releasing interrupts)

MAKEPLAY (Established by "GODSPRS+1", and entered directly from "GOFLASH2+1", as common starting point for all processing of those display interface routines requiring display activity)

MPAC+7 = (bits 14-10 of PRIORITY) (priority of present job)

Change priority of present job to 33<sub>8</sub>

If bit 15(MKEXREQ) of MPAC+4 = 1: (i.e. mark/extended verb)

COPINDEX = 1 (Tag here "MAKEMARK")

DSPFLG+1 = MPAC+4

If bits 6(DOTHNRET) and 4(FLREQ) of MPAC+4  $\neq 00_2$ :

CADRFLSH+1 = MPAC+3 (calling address information  
needed for flash or return)

NWORD+1 = MPAC+1 (verb-noun information)

If bits 14(PRIODLE), 13(NORMIDLE), 12(PDSPFLAG), 8(NRMWTKEY),  
and 7(PROWTKEY) of FLAGWRD<sub>4</sub> all = 0: (no priority or

normal display waiting for  
response or busy DSKY to be  
released)

If bit 9(MRKWTKEY) of FLAGWRD<sub>4</sub> = 1, End of job (mark  
display waiting for  
busy DSKY to be released)

Proceed to "MARKPLAY"

If bits 14(PRIODLE), 12(PDSPFLAG), and 7(PROWTKEY) of  
FLAGWRD<sub>4</sub> all = 0: (i.e. bits 13 and/or 8 are 1 only)

Set bit 3(MKOVNORM) of FLAGWRD<sub>4</sub> = 1

MPAC+0 = 1

Proceed to "JOBXCHS+1" (remove present normal display  
from DSKY interface registers,  
then go to "MARKPLAY")

If bits 11(MARKWAIT) and 5(RUPTMARK) of FLAGWRD<sub>4</sub>  $\neq 00_2$ :

End of job (mark display already in system awaiting  
generation or restoration)

Set bit 11(MARKWAIT) of FLAGWRD<sub>4</sub> = 1

MPAC+0 = 0

Awaken job (if any) with starting address id = "MARKPLAY"

If LOCCTR  $\geq$  0: (indicates a job found)

Set starting address of awakened job to "End of job"

Put present job to sleep (starting address id = "MARKPLAY")

If bit 7(PRIOREQ) of MPAC+4 = 1: (i.e. priority display)

COPINDEX = 0 (Tag here "MAKEPRIO")

If bit 14(RESETREQ) of MPAC+4 = 0: (no repeat display)

If bits 14(PRIODLE) and 7(PROWKEY) of FLAGWRD4  $\neq$  00<sub>2</sub>:

Proceed to "POODOO" (pattern 21502<sub>8</sub>) (display in system already)

If bit 14(RESETREQ) of MPAC+4 = 1:

If CADDRFLSH+0 = MPAC+3:

If DSPLOCK > 0, End of job

If bits 15(MARKIDLE) and 9(MRKWKEY) of FLAGWRD4  $\neq$  00<sub>2</sub>:

MPAC+0 = 0

Proceed to "JOBXCHS+1" (remove present mark display from DSKY interface registers, then go to "OKTOCOPY")

If bits 13(NORMIDLE) and 8(NRMWKEY) of FLAGWRD4  $\neq$  00<sub>2</sub>:

MPAC+0 = 1

Proceed to "JOBXCHS+1" (remove present normal display from DSKY interface registers, then go to "OKTOCOPY")

Proceed to "OKTOCOPY"

If bits 15(MKEXREQ) and 7(PRIOREQ) of MPAC+4 = 00<sub>2</sub>: (as they will, normal display)

COPINDEX = 2 (Tag here "IFLEGAL")

If bit 14(RESETREQ) of MPAC+4 = 0:

If bit 4(FLREQ) of DSPFLG+2 = 1:

If bits 13(NORMIDLE), 10(NORMWAIT), 8(NRMWKEY), and 4(RUPTNORM) of FLAGWRD4  $\neq$  0000<sub>2</sub>:

Proceed to "POODOO" (pattern 21502<sub>8</sub>) (normal display already in system)

If bit 14(RESETREQ) of MPAC+4 = 1:

If CADDRFLSH+2 = MPAC+3:

If DSPLOCK > 0, End of job

(If bits 15(MKEXREQ) and 7(PRIOREQ) of MPAC+4 = 00<sub>2</sub>):

DSPFLG+2 = MPAC+4 (Tag here "OKTOPLAY")

If bits 6(DOTHNRET) and 4(FLREQ) of MPAC+4 ≠ 00<sub>2</sub>:

CADRFLSH+2 = MPAC+3 (calling address information)

NVWORD+2 = MPAC+1 (verb-noun information)

RESTREG = MPAC+7 + Channel 7 (in bits 7-5, superbank value)

If any of bits 15(MARKIDLE), 14(PRIODLE), 12(PDSPFLAG),  
11(MARKWAIT), 9(MRKWTKEY), 7(PROWTKEY), 5(RUPTMARK),  
or 1(XDSPFLAG) of FLAGWRD<sub>4</sub> = 1:

Set bit 10(NORMWAIT) of FLAGWRD<sub>4</sub> = 1

MPAC+0 = 1

Awaken job (if any) with starting address id = "PLAYJUM1"

If LOCCTR > 0: (indicates a job found)

Set starting address of awakened job to "End of job"

Put present job to sleep (starting address id = "PLAYJUM1")

Set bit 5(Key Release) of channel 11 = 0

If bit 8(NRMWTKEY) of FLAGWRD<sub>4</sub> = 1: (bits 9,7 known 0)

Set TS = DSPLIST and DSPLIST = 0 (normal display wait-  
ing for DSKY to be

If bit 8(NRMWTKEY) of FLAGWRD<sub>4</sub> = 0: released)

Set TS = CADRSTOR and CADRSTOR = 0

Awaken job (if any) with starting address id = TS

If LOCCTR > 0: (indicates a job found)

Set starting address of awakened job to "End of job"

Proceed to "PLAYJUM1"

OKTOCOPY (Entered after completion of DSKY interface registers  
resets, if needed, to allow for a priority display)

COPINDEX = 0

DSPFLG+0 = MPAC+4

If bits 6(DOTHNRET) and 4(FLREQ) of MPAC+4 ≠ 00<sub>2</sub>:

CADRFLSH+0 = MPAC+3 (calling address information)

NVWORD+0 = MPAC+1 (verb-noun information)

Set bit 5(Key Release) of channel 11 = 0

If bits 9(MRKWTKEY), 8(NRMWTKEY), and 7(PROWTKEY) of FLAGWRD4  $\neq$  000<sub>2</sub>:

Set TS = DSPLIST and DSPLIST = 0 (only bit 7 of interest)

If bits 9(MRKWTKEY), 8(NRMWTKEY), and 7(PROWTKEY) of FLAGWRD4 = 000<sub>2</sub>:

Set TS = CADRSTOR and CADRSTOR = 0 (only bit 7 of interest)

Awaken job (if any) with starting address id = TS

If LOCCTR  $\geq$  0: (indicates a job found)

Set starting address of awakened job to "End of job"

Proceed to "REDOPRIO"

JOBXCHS+1 Entered to replace the present contents of the DSKY interface registers (CADRSTOR and DSPLIST) with 0 for use by a display pre-empting the display system: a priority display replaces a mark/extended verb or normal display; a mark/extended verb display replaces a normal display.

Set bit 5(Key Release) of channel 11 = 0

If bits 9(MRKWTKEY), 8(NRMWTKEY), and 7(PROWTKEY) of FLAGWRD4  $\neq$  000<sub>2</sub>:

Set TS = DSPLIST and DSPLIST = 0 (only bits 9,8 of interest)

If bits 9(MRKWTKEY), 8(NRMWTKEY), and 7(PROWTKEY) of FLAGWRD4 = 000<sub>2</sub>:

Set TS = CADRSTOR and CADRSTOR = 0 (only bits 9,8 of interest)

Awaken job (if any) with starting address id = TS

If LOCCTR  $\geq$  0: (indicates a job found)

Set starting address of awakened job to "XCHSLEEP"

Set MPAC+0 of awakened job (using LOCCTR for indexing) equal to MPAC+0 of the present job

If MPAC+0 = 0:

Set bits 15(MARKIDLE) and 9(MRKWTKEY) of FLAGWRD4 = 0

Set bit 5(RUPTMARK) of FLAGWRD4 = 1 (mark display interrupted)

If MPAC+0 = 1:

Set bits 13(NORMIDLE) and 8(NRMWTKEY) of FLAGWRD4 = 0

Set bit 4(RUPTNORM) of FLAGWRD4 = 1 (normal display interrupted)



If bit 3(MKOVNORM) of FLAGWRD<sub>4</sub> = 1, proceed to "MARKPLAY"

Proceed to "OKTOCOPY"

#### XCHSLEEP

If MPAC+0 = 0:

TS = "MARKPLAY"

If MPAC+0 = 1:

TS = "PLAYJUM1"

Awaken job (if any) with starting address id = TS

If LOCCTR > 0: (indicates a job found)

Set starting address of awakened job to "End of job"

Put present job to sleep with starting address id = TS (has MPAC+i registers set for functions of present job rather than previously sleeping job, which are lost)

REDOPRIO Entered when it is concluded that a priority display on the DSKY should take place (provided DSKY not busy).

PRIOTIME = TIME1

COPINDEX = 0 (Tag here "KEEPPRIO")

Proceed to "NVDSP"

MARKPLAY Entered when it is concluded that a mark/extended verb display on the DSKY should take place (provided DSKY not busy).

Set bit 3(MKOVNORM) of FLAGWRD<sub>4</sub> = 0

Set bit 1(XDSPFLAG) of FLAGWRD<sub>4</sub> = 1

TS<sub>1</sub> = 0 (L register, for "NVMONOPT" use)

If bit 8(DECMON) of DSPFLG+1 = 1: ("MARKMONR" entrance)

TS<sub>1</sub> = (bits 14-8 of NWORD+1)

Skip next line (i.e. always complement NWORD+1)

If bit 5(PERFREQ) of DSPFLG+1 = 1:

NWORD+1 = - NWORD+1

COPINDEX = 1

Proceed to second line of "NVDSP"

PLAYJUM1 Entered when it is concluded that a normal display on the DSKY should take place (provided system not busy).

COPINDEX = 2

Proceed to "NVDSP"

NVDSP Entered (with COPINDEX set to indicate type of display) to initiate display on DSKY (provided system not busy due to keyboard/uplink use). Tag also "GOPLAY".

$TS_1 = 0$  (L register, for "NVMONOPT" use)

$MPAC+6 = K_{octmsk_{COPINDEX}}$  (enter here from "MARKPLAY")

$MPAC+4 = DSPFLG_{COPINDEX}$

EBANK = (bits 11-9 of MPAC+4)

$TS_1 = TS_1 + (\text{bits } 3-1 \text{ of } MPAC+4)$  (bits 3-1 for blanking)

Set bit 13(2NDPERF) of  $DSPFLG_{COPINDEX} = 0$

$MPAC2SAV = MPAC+2$  (contains machine address for address-to-

$MARK2PAC = MPAC+2$  be-specified nouns)

If  $NVWORD_{COPINDEX} > 0$ :

$TS = NVWORD_{COPINDEX}$

If  $NVWORD_{COPINDEX} = 0$ : (e.g. "CLEANDSP" or "KLEENEX" entered)

Establish "JAMTERM" (priority  $32_8$ ) (note that DSPLOCK not checked)

Proceed to second line of "FLASHSUB"

If  $NVWORD_{COPINDEX} < 0$ : (set in "MARKPLAY")

$NVWORD+1 = - NVWORD+1$

$TS = (\text{bits } 7-1 \text{ of } NVWORD+1) + 01200_8$  (verb 05 scaled)

If bit 8(DECMON) of  $DSPFLG_{COPINDEX} = 1$ : (i.e. from "MARKMONR")

$TS = TS + 2600_8$  (changes verb to 16)

Proceed to "NV5ODSP"

NV5ODSP

Perform "NVMONOPT": if return to calling address +1(busy), proceed to "REST"  
otherwise, proceed (TS and  $TS_1$  set)

Set bit 6(Flash) of channel 11 = 0

MPAC+2 = MPAC2SAV

MPAC+6 =  $K_{\text{octmsk}_{\text{COPINDEX}}}$

MPAC+4 =  $\text{DSPFLG}_{\text{COPINDEX}}$

EBANK = bits 11-9 of MPAC+4

Set bits 9(MRKWIKEY), 8(NRMWIKEY), and 7(PROWIKEY) of FLAGWRD4 = 0

TS = MPAC+4 (bits 3-1 are R3BLNK, R2BLNK, and R1BLNK)

Perform "BLANKSUB": if return to calling address +1 (busy, as should not be), proceed to "NVDSP" otherwise, proceed

If bits 13(2NDPERF) and 5(PERFREQ) of MPAC+4 =  $01_2$ : (Tag here "PERFCHEK")

Set bit 13 of  $\text{DSPFLG}_{\text{COPINDEX}} = 1$

$\text{TS}_1 = 0$  (L register, for "NVMONOPT" use)

If bit 15(MKEXREQ) of  $\text{DSPFLG}_{\text{COPINDEX}} = 1$ :

TS = bits 14-8 of NVWORD+1 (i.e.  $xx00_{\text{vn}}$ ) (Tag here "MARKPERF")

Proceed to "NV5ODSP"

If bit 12(BURNREQ) of  $\text{DSPFLG}_{\text{COPINDEX}} = 0$ : (i.e. not from "CLOCKJOB")

TS =  $5000_{\text{vn}}$

Proceed to "NV5ODSP"

TS =  $9700_{\text{vn}} - \text{NVWORD1}$  ( $9700_{\text{vn}}$  for NVWORD1 = 0,  $9900_{\text{vn}}$  for NVWORD1 = -400<sub>8</sub>: NVWORD1 should not be confused with NVWORD+i).

Proceed to "NV5ODSP"

If bit 4(FLREQ) of MPAC+4 = 1, proceed to "FLASHSUB" (Tag here "GOANIDLE")

If bit 6(DOTHNRET) of MPAC+4 = 1:

TS =  $\text{CADRFLSH}_{\text{COPINDEX}}$

Change priority of this job to that in bits 14-10 of MPAC+7  
(restoring original value)

Proceed to address specified by TS

If bits 7-1 of NVWORD<sub>COPINDEX</sub> = 0: (i.e. xx00<sub>vn</sub>)

Proceed to "FLASHSUB"

End of job

REST Entered from beginning of "NV50DSP" for a busy return from  
"NVMONOPT" (DSKY or uplink being used externally)

If CADRSTOR  $\neq$  0, End of job

If COPINDEX = 0:

Set bit 7(PROWTKEY) of FLAGWRD<sub>4</sub> = 1 (priority display)

If COPINDEX = 1:

Set bit 9(MRKWTKEY) of FLAGWRD<sub>4</sub> = 1 (mark display)

If COPINDEX = 2:

Set bit 8(NRMWTKEY) of FLAGWRD<sub>4</sub> = 1 (normal display)

Proceed to "NVSUBUSY"

#### NVSUBUSY

If CADRSTOR  $\neq$  0, proceed to "POODOO" (pattern 21206<sub>g</sub>)

If DSPLIST  $\neq$  0, proceed to "POODOO" (pattern 21206<sub>g</sub>)

Set bit 5(Key Release) of channel 11 = 1

If COPINDEX = 0:

DSPLIST = "REDOPRIO"

If COPINDEX = 1:

DSPLIST = "MARKPLAY"

If COPINDEX = 2:

DSPLIST = "PLAYJUM1"

Put present job to sleep (starting address id = DSPLIST)

#### FLASHSUB

Set bit 6(Flash) of channel 11 = 1

MPAC+3 = COPINDEX

If COPINDEX = 0:

Set bit 14(PRIODLE) of FLAGWRD<sub>4</sub> = 1

If COPINDEX = 1:

Set bit 15(MARKIDLE) of FLAGWRD<sub>4</sub> = 1

If COPINDEX = 2:

Set bit 13(NORMIDLE) of FLAGWRD<sub>4</sub> = 1

If CADDRSTOR  $\neq$  0:

If bits 15(MARKIDLE) and 6(PINBRFLG) of FLAGWRD<sub>4</sub> = 00<sub>2</sub>:

Proceed to "POODOO" (pattern 21502<sub>8</sub>)

End of job

Proceed to "ENDIDLE"

#### ENDIDLE

If CADDRSTOR  $\neq$  0, proceed to "POODOO" (pattern 21206<sub>8</sub>)

If DSPLIST  $\neq$  0, proceed to "POODOO" (pattern 21206<sub>8</sub>)

CADDRSTOR = "IDLERET1"

Put present job to sleep (starting address id = CADDRSTOR)

#### IDLERET1

If MPAC+0  $\neq$  21, 22, or 23: (loaded in "RECALTST" with verb)

Proceed to "OKTOENT"

TS = (bits 7-1 of NVWORD<sub>MPAC+3</sub>) (MPAC+3 has COPINDEX)

If MPAC+1 = TS: (loaded in "RECALTST" with noun)

Proceed to "OKTOENT"

Proceed to "PINBRNCH" (noun loaded not what display routine asking)

OKTOENT Entered after response to flash

If bit 15(MARKIDLE) of FLAGWRD<sub>4</sub> = 1: (mark display waiting)

Set bit 3(MKOVNORM) of FLAGWRD<sub>4</sub> = 0

Proceed to "ENDRET" ("NORMRET" processing done e.g. via "ENDEXT")

If bit 14(PRIOIDLE) of FLAGWRD<sub>4</sub> = 0:

Proceed to "NORMRET"

If NVWORD+0 = 0: (priority "JAMTERM"-type action, e.g. from "R61CSM")

Proceed to "NORMRET"

TS = PRIOTIME - TIME1 (priority display, enforce minimum delay)

If TS > 0:

TS = 163.84 - TS seconds (correct for TIME1 overflow)

If ( $|TS| + K_{m2sec}$ ) ≤ 0:

Proceed to second line of "REDOPRIO" (response too near initiation)

Proceed to "NORMRET"

NORMRET Entered from "OKTOENT" and "ENDEXT"

If bits 11(MARKWAIT) and 5(RUPTMARK) of FLAGWRD<sub>4</sub> ≠ 00<sub>2</sub>:

Set bits 11(MARKWAIT) and 5(RUPTMARK) of FLAGWRD<sub>4</sub> = 0

TS = "MARKPLAY"

Awaken job with starting address id = TS

Proceed to "ENDRET"

If bits 10(NORMWAIT) and 4(RUPTNORM) of FLAGWRD<sub>4</sub> ≠ 00<sub>2</sub>:

Set bits 10(NORMWAIT) and 4(RUPTNORM) of FLAGWRD<sub>4</sub> = 0

TS = "PLAYJUM1"

Awaken job with starting address id = TS

Proceed to "ENDRET"

If bits 6(DOTHNRET) and 4(FLREQ) of DSPFLG+2 ≠ 00<sub>2</sub>:

Proceed to "ENDRET" (both bits set by "ROO")

If NVWORD+2 = 0, proceed to "ENDRET"



Establish "PLAYJUM1" (priority 15<sub>g</sub>)

Proceed to "ENDRET"

#### ENDRET

If LOADSTAT > 0, End of job (e.g. from "ENDEXT")

If LOADSTAT = +0, Error

If bit 2(VNFLAG) of FLAGWRD4 = 1:

If MPAC+3 = 2: (i.e. normal display)

Set bit 2(VNFLAG) of FLAGWRD4 = 0 (Tag here "VNRET")

TS = LOADSTAT + 1

If TS > 0: (LOADSTAT was -0, "otherwise" response of an  
e.g. data load)

MPAC+3 = CADRFLSH+2 - 3

If TS = 0: (LOADSTAT was -1, a PRO response)

MPAC+3 = CADRFLSH+2 - 1

If TS < 0: (LOADSTAT was -2, a V34E response)

MPAC+3 = "GOTOPOOH"

Proceed to "ENDRET2"

MPAC+3 = CADRFLSH<sub>MPAC+3</sub> + LOADSTAT + 2

Proceed to "ENDRET2"

#### ENDRET2

Set those bits of 15-12 and 6 in FLAGWRD4 to 0 that are 1 in MPAC+6

(If MPAC+6 indicates priority, bits 14(PRIODLE) and 6(PINBRFLG) reset; if indicates mark, bit 15(MARKIDLE) reset; if normal, bits 13(NORMIDLE) and 6(PINBRFLG) reset. MPAC+6 loaded near start of "NV5ODSP").

TS = -3 (blank all display registers except program)

Perform "NVSUB": if return to calling address +1(busy), proceed  
otherwise, proceed

Change priority of present job to that specified in bits 14-10  
of MPAC+7

Proceed to address specified by MPAC+3

## Quantities in Computations

See also list of major variables and list of routines

CADRFLSH<sub>i</sub> (i = 0,1,2): Set of three single precision erasable memory cells used to retain return address information (in FCADR format, with bits 15-11 giving the FBANK and bits 10-1, plus 2000<sub>8</sub>, giving the S-register contents) from the display interface routine (for bits 6(DOTHNRET) or 4(FLREQ) of DSPFLG<sub>i</sub> being 1). CADRFLSH+0 is the address for a priority display, CADRFLSH+1 for a mark/extended verb display, and CADRFLSH+2 is for a normal display. For most flashing three-response displays (all except "VNFLASH(R)"), the cells contain the information for the terminate return address, and are incremented in "ENDRET" for proceed or "otherwise" (e.g. data enter or recycle) responses. For "VNFLASH(R)", the cells contain one greater than the PRO return address (for "INITDSP" reasons): a terminate response exits directly to "GOTOPOOH", while an "otherwise" response causes the return to be 3 less than the present contents, i.e. the acquisition of the verb-noun pattern. CADRFLSH+2 also has the program notation "TEMPFLSH". "INITDSP", entered if restart logic invoked, causes transfer to CADRFLSH+2 - 3 (the "setting" of TS for the standard display routine calling sequence): consequently, if CADRFLSH+2 is set to "XXXX" +3, this logic will cause "XXXX" to be entered (and is so used at the end of "ROO").

CADRSTOR: See Data Input/Output. It is 0 if no job in "ENDIDLE", and otherwise is set to "IDLERET1".

COPINDEX: Single precision cell, scale factor B14, used to retain indexing information on the type of display being processed (0, 1, and 2 for priority, mark/extended verb, and normal respectively). It is assigned to the LOC cell of the job register set (which is written over with job starting information if the job is put to sleep).

DSPFLG<sub>i</sub> (i = 0,1,2): Set of three single precision erasable memory cells used to retain values of control-bit information for individual displays (priority, mark/extended verb, and normal respectively). FLAGWRD<sub>4</sub> is used to contain control and status information for the display routines as a whole. When a display interface routine is initially entered, cell MPAC+4 is used to store temporarily the required settings for the pertinent DSPFLG cell. Individual bits in the DSPFLG cells have the following significance(words also "EBANKSAV", "MARKFLAG", and "EBANKTEM"):

<u>Bit</u>	<u>Symbol</u>	<u>Meaning</u>
15	MKEXREQ	Mark or extended verb request. Loaded only in DSPFLG+1 (and, of course, MPAC+4): bit in other DSPFLG cells expected to be 0.
14	RESETREQ	Repeat request (causes checks for already existing priority or normal displays to be bypassed: otherwise, an abort (pattern 21502 <sub>8</sub> ) would be caused if a display of the same type already in system). Bit can also be set via the "BLANKET" routine (if a proper "R" display is used), as well as by the "RExxxxx" interface routines.

<u>Bit</u>	<u>Symbol</u>	<u>Meaning</u>
13	2NDPERF	Second part of "perform" display sequence has been started (to "paste" a code verb, e.g. 50, after completion of a display loading). Bit set 0 at the start of "NVDSP" and to 1 in "NV50DSP" if bit 5(PERFREQ) is 1 (and this bit, of course, is 0).
12	BURNREQ	Special display verb to request ignition authorization should be generated. Verb 99 is used if NVWORD1 = - 400 <sub>8</sub> ; if NVWORD1 = 0, then verb 97 is "pasted" instead. See "CLOCPLAY".
11-9		Value of EBANK when display routine entered (restored in "NVDSP" to ensure proper value when job calling routine, if appropriate, is started again).
8	DECMON	Decimal monitor verb required for perform-type display (verb pasted over with one in calling sequence information), set for entrance to "MARKMONR". No requirement for resetting bit exists (DSPFLG <sub>1</sub> loaded completely for new display).
7	PRIOREQ	Priority display requested (interface routine "PRIODSP(R)" used). Loaded only in DSPFLG+0.
6	DOETHNRET	Return requested after performance of the DSKY loading (interface routine "EXDSPRET" or "GODSPRET" used), to allow, for example, a verb "paste" without a flash.
5	PERFREQ	Perform-type display requested (see bit 13).
4	FLREQ	Flashing-type display (with associated requirement for processing of crew response) requested.
3	R3BLNK	Blanking of register R3 requested (done by transfer to "BLANKSUB" in "NV50DSP"). Bit can be set (usually) by use of an "R" display interface routine, followed by transfer to "BLANKET" with TS = lxx <sub>2</sub> .
2	R2BLNK	Blanking of register R2 requested (can be set by transfer to "BLANKET" with TS = xlx <sub>2</sub> ). See bit 3 discussion.
1	R1BLNK	Blanking of register R1 requested (can be set by transfer to "BLANKET" with TS = xxl <sub>2</sub> ). See bit 3 discussion.

DSPLIST: See Data Input/Output. It is a DSKY interface cell used to retain return address information if the display system is busy due to crew or uplink use (loaded in "NVSUBUSY" which is entered from "REST"). For index convenience, the cell is stored to make it the same as CADRSTOR+1.

DSPLOCK: See Data Input/Output (set 1 at start of "CHARIN" to indicate that the display system is busy with DSKY or uplink inputs, and hence should not be used by internal routines, not even for a priority display). It is checked automatically at the start of "NVSUB" and "BLANKSUB", causing return to calling address +1 for a non-zero value. Same check at start of "NVMONOPT" also.

EBANK: See Data Input/Output (only bits 11-9 are defined).

EXTVBACT: See Verb Definitions.

FLAGWRD4: Single precision flagword cell whose individual bits are used to contain control and status information for the display routines as a whole (DSPFLG<sub>i</sub> is used for individual displays). It is set zero in "STARTSB2"<sup>1</sup>. Individual bits have the following significance:

<u>Bit</u>	<u>Symbol</u>	<u>Meaning</u>
15	MARKIDLE	A mark/extended verb display is waiting for a response (is the job asleep in "ENDIDLE"). Bit set in "FLASHSUB" prior to entering "ENDIDLE", and used in "OKTOENT" to identify the type of response received, whereupon it is reset (in "ENDRET2") prior to returning to caller. Could also be reset in "JOBXCHS+1" if display pre-empted by a priority display, whereupon bit 5(RUPTMARK) would be set 1 to record this fact.
14	PRIOIDLE	A priority display is waiting for a response (is the job asleep in "ENDIDLE"). Bit set in "FLASHSUB" prior to entering "ENDIDLE", and used in "OKTOENT" to identify the type of response received, whereupon it is reset (in "ENDRET2") prior to returning to caller.
13	NORMIDLE	A normal display is waiting for a response (is the job asleep in "ENDIDLE"). Bit set in "FLASHSUB" prior to entering "ENDIDLE", and used in "OKTOENT" to identify the type of response received, whereupon it is reset (in "ENDRET2") prior to returning to caller. Could also be reset in "JOBXCHS+1" if display pre-empted by a priority or mark/extended verb display, whereupon bit 4(RUPTNORM) would be set 1 to record this fact.

<u>Bit</u>	<u>Symbol</u>	<u>Meaning</u>
12	PDSPFLAG	Bit set 1 to indicate that R60 display should be of the priority type (checked in "CHKLINUS", and set in "R61CSM" if a maneuver must be done by R60 means, then reset).
11	MARKWAIT	A mark/extended verb display is waiting to be initiated. Bit is set in "MAKEPLAY" if a priority display presently on DSKY (assuming bit 5 (RUPTMARK) is 0), and job put to sleep with starting address id = "MARKPLAY". Job awakened, and bit reset, in "NORMRET" after completion of present display effort. Helps give display sequence for DSKY of: crew/uplink use, priority display, interrupted mark/extended verb display, waiting mark/extended verb display, waiting normal display, interrupted normal display. An interrupted and a waiting display of the same type, however, cannot be in system at same time. In addition, waiting or interrupted normal displays will <u>not</u> be restarted by an end to a mark/extended verb display until e.g. "ENDEXT" is entered, and bit 1 of FLAGWRD4 serves to keep out normal displays once a mark/extended verb display attempt initiated (until "CLEARMRK" done).
10	NORMWAIT	A normal display is waiting to be initiated. The bit is set 1 in "MAKEPLAY" if a priority or mark/extended verb activity in the display system is indicated (including bit 1(XDSPFLAG)), and the job is put to sleep with starting address id = "PLAYJUM1". Job awakened, and bit reset, in "NORMRET" after completion of present display effort, provided bits 11 and 5 are 0.
9	MRKWTKEY	A mark/extended verb display attempt found the display system busy (so must wait for a key release). Bit is set in "REST" to indicate that a display of this type in DSPLIST (via "NVSUBUSY"). When job is awakened (e.g. by "RELDSP"), it starts at "MARKPLAY", and the bit would be reset in "NV5ODSP" after successful initiation of the display. Bit could also be reset in "JOBXCHS+1" if a mark attempt was pre-empted by a priority display (cf. bit 15, since it would be reset at same time).
8	NRMWTKEY	A normal display attempt found the display system busy (see bit 9 discussion). When awakened, starts at "PLAYJUM1"; could also be reset if normal display pre-empted (cf. bit 13, since it would be reset at the same time).
7	PROWTKEY	A priority display attempt found the display system busy (see bit 9 discussion). When awakened, starts at "REDOPRIO".

<u>Bit</u>	<u>Symbol</u>	<u>Meaning</u>
6	PINBRFLG	Bit set to 1 in "PINBRNCH" if it is concluded that interference with the existing display has taken place (e.g. computations at the end of "FLASHSUB" conclude that a data enter verb was received but the associated noun was <u>not</u> that originally requested by the display). Bit also set 1 in "ENDEXT" if a priority or normal display waiting for response (bits 14 and/or 13 $\neq$ 0). Bit reset 0 in "ENDRET2" upon successful conclusion of a priority or normal display (bit not set in "PINBRNCH" if bit 15 (MARKIDLE) = 1). Used in "FLASHSUB" to bypass abort otherwise caused if CADDRSTOR $\neq$ 0.
5	RUPTMARK	A mark/extended verb display (or display attempt, e.g. if system busy) has been interrupted by a priority display. Bit is set in "JOBXCHS+1" (cf. bits 15 and 9, which would be reset) if the mark/extended verb activity is pre-empted by a priority display request. Job awakened, and bit reset, in "NORMRET" (same effect there as bit 11).
4	RUPTNORM	A normal display (or display attempt, e.g. if system busy) has been interrupted by a priority display or by a mark/extended verb display. Bit is set in "JOBXCHS+1" (cf. bits 13 and 8, which would be reset then), and used in "NORMRET" to awaken job, provided bits 11 and 5 are 0 (same effect there as bit 10), whereupon it is set 0.
3	MKOVNORM	Mark display is to pre-empt a normal display: set as described for bit 4 setting for mark/extended verbs. Bit set 1 in "MAKEPLAY" to control proper exit from "JOBXCHS+1", namely to "MARKPLAY", where bit is set 0 again. Also reset in "OKTOENT".
2	VNFLAG	Bit set 1 by "VNFLASH(R)" to indicate a special use of display interface (see MPAC+3), and reset in "ENDRET" at the conclusion of processing for it.
1	XDSPFLAG	Bit set 1 in "MARKPLAY" to signify that a mark/extended verb display has been initiated (or an attempt of that type). Used in "MAKEPLAY" to keep normal displays from using the DSKY if bit is 1. Bit reset by "CLEARMRK" (entered e.g. from "ENDEXT").

$K_{m2sec}$ : Single precision constant, program notation "-2SEC", scale factor  $B14$ , units centi-seconds. Value is  $-200 \times 2^{-14}$ , corresponding to -2 seconds.

$K_{octmsk}$ : Set of three single precision octal constants used as bit masks, program notation ( $i = 0$ ) "PRIOCT", values  $2014_8$ ,  $4242_8$ , and  $11254_8$  respectively. The setting of bit 3 = 1 is a holdover from a former flagbit routine, and is no longer employed.



LOADSTAT: See Data Input/Output. Values of -0, -1, and -2 are set for "otherwise" (e.g. V32E or data enter), PRO, and V34E responses respectively.

LOCCTR: Single precision cell set in executive system to -1 if the "job wake" routine found no job with the indicated starting address id; otherwise, it is set to an index parameter (B14) for locating the core register set of the awakened job (should not be zero if the "job wake" routine itself was entered from a job). See 3420.5-27 for details. It is set similarly when a job established.

MARK2PAC: Single precision cell used to retain the value of MPAC+2 when "NVDSP" was last entered, and used to load the proper value when "PINBRNCH" is entered. MPAC+2 is used to contain the machine address for address-to-be-specified nouns used internally.

MPAC<sub>i</sub> (i = 0-7): Set of cells associated with each job register (a collection of 12 erasable memory cells assigned to each job) set, used in the interpretive language for vector and scalar accumulator functions, and used in machine language as temporary storage cells. Assignments of significance for the display interface routines are listed below (other quantities also use these cells, but are listed separately). Note that each job has its own job register set (hence there could be distinct MPAC+7 values, for example, for several different display interface functions simultaneously, each waiting for the necessary action to be completed).

MPAC+0: Index cell used by "XCHSLEEP" and "JOBXCHS+1", set to 0 to change mark/extended verb displays and 1 for normal displays. Program notation is "FACEREG". Cell loaded in "RECALTST" with the verb pattern (VERBREG) when "RECALTST" entered.

MPAC+1: Cell loaded with verb/noun pattern when display interface routine entered (frequently the TS setting, mechanized as the hardware accumulator). Program notation "PLAYTEML". Cell loaded in "RECALTST" with the noun pattern (NOUNREG) when "RECALTST" entered.

MPAC+2: Cell loaded with machine address for address-to-be-specified nouns.

MPAC+3: Cell loaded with calling address information (i.e. the "immediate return" for non-flash or the "terminate return" for flash). It is loaded with address of step "L" in the sequence (program notation "PLAYTEM3"):

L-3	CA	(TS information, e.g. verb/noun)
L-2	TC	BANKCALL (routine to change program banks)
L-1	CADR	XXXX (address of interface routine)
L		(terminate return)
L+1		(proceed return)
L+2		(enter/recycle return)
L+3		(return for "R" flash-type displays, usually)

This sequence does not apply to "VNFLASH(R)" sequences. For

these two calling routines, the entrance point is in fixed-fixed memory, meaning that the "BANKCALL" interface routine is not necessary. Hence MPAC+3 is loaded with address of step "L" in the following sequence:

L-3	CA	(TS information, e.g. verb/noun) (enter/V32
L-2	TC	"VNFLASH(R)" return)
L-1		(proceed return)
L		(return for "VNFLASHR")

The setting of "L" as indicated is accomplished by incrementing the Q register at the entrance to the routine: this allows "INITDSP" restart logic to transfer to L-3 to obtain the TS information. The return to L-1 for a PRO and L-3 for an "otherwise" is accomplished by the logic in "ENDRET"; a V34E causes direct exit to "GOTOPOOH".

The same cell is loaded with the value of COPINDEX at the start of "FLASHSUB", where has notation COPMPAC, to retain COPINDEX information when job put to sleep.

MPAC+4: Cell located with appropriate DSPFLG pattern in display interface routine. Program notation "PLAYTEM4" and "TEMPOR2".

MPAC+5: Cell loaded in "GODSPRS+1" with the value of LOCCTR for the display job established (for "R" type displays, except "MARKMONR"), to permit "BLANKET" to be entered to set additional bits of MPAC+4, such as bits 3-1 (RiBLNK).

MPAC+6: Cell used to contain the masking bit pattern ( $K_{octmsk}$ ) for the display, employed in "ENDRET2" to reset the proper bits of FLAGWRD4. Program notation "GENMASK".

MPAC+7: Cell loaded in "MAKEPLAY" with the priority of the using routine, for use in "ENDRET2" to restore the same priority. This works for "R" type displays also, since "GODSPRS+1" establishes "MAKEPLAY" with the same priority as that of the calling job. Program notation is "USERPRIO": cell strictly speaking is not part of the "MPAC" set, but instead is used in interpretive language for indicating type of computation ("MODE" value) being performed.

MPAC2SAV: Cell (part of job register set, corresponding to MPAC+9) used to retain value of MPAC+2 at the start of "NVDSP", for restoration after "NVMONOPT" (which uses the cell for temporary storage purposes) has been performed.

NVWORD<sub>i</sub> (i = 0,1,2): Set of three single precision erasable memory cells used to retain values of the noun-verb pattern (bits 14-8 for verb and 7-1 for noun) for priority, mark/extended verb, and normal displays respectively. Program notations "NVWORD", "MARKNV", and "NVSAVE" respectively (NVWORD+2 and DSPFLG+2 set 0 in "DOFSTART").

NVWORD1: See Burn Control.

OPTION1, OPTION2: Pair of single precision cells displayed in octal by using noun 06 (implemented in some cases by loading OPTION2 before entering the display interface routine "GOPERF4", and setting TS to OPTION1). OPTION1 is used to contain the "option code identification" (for the purpose of the display, such as specification of IMU orientation) and OPTION2 is used to contain the "option selection" consistent with OPTION1 (such as "IMU orientation to preferred alignment"). A manual input changing only OPTION1 (leaving OPTION2 unchanged) will be interpreted as a response, and can not be used to change the subsequent program logic.

PRIORITY: See General Program Control.

PRIOTIME: Single precision value of least significant half of computer clock (TIME1), scale factor BL4, units centi-seconds, used to retain the time when "REDOPRIO" last entered. Cell is used in "OKTOENT" to check that the present magnitude of the difference between PRIOTIME and TIME1 (after allowing for TIME1 overflow) is greater than  $-K_{m2sec}$ : if not, it is suspected that the response was associated with the interrupted display rather than the priority display, and consequently the priority display is repeated (without, however, resetting PRIOTIME). Since only single precision times are used, a time lapse of  $163.84n + 2 - 0$  seconds (n an integer) is interpreted as being in the "too quick" zone.

RESTREG: Single precision cell used to retain restart information for normal displays. It is initialized to  $30000_8$  in "DOFSTART", meaning priority  $30_8$ , and is used in "INITDSP" (entered if a restart encountered with a x.1 restart phase setting) to properly restore value of PRIORITY and Channel 7 (FEXT or SUPERBNK). The cell is also loaded in "MAKEPLAY" (for a normal display only).

## Display Computations

GET+MGA Entered with  $TS$  = active vehicle velocity increment

$$TS_1 = \sin^{-1} (\text{unit} TS \cdot \text{REFSMMAT}_3)$$

If  $TS_1 < 0$ :

$$TS_1 = TS_1 + 1 \quad (\text{the } 1 \text{ is } 360^\circ)$$

$$pMGA = TS_1$$

Return

GET.LVC Entered with  $TS$  = active vehicle velocity increment

$$TS_3 = - \text{unit} R_{\text{init}}$$

$$TS_2 = \text{unit}(TS_3 * V_{\text{init}})$$

$$TS_1 = \text{unit}(TS_2 * TS_3)$$

If bit 5(REVFLAG) of FLAGWRD2 = 1: (e.g. from "N81DISP")

Set bit 5(REVFLAG) of FLAGWRD2 = 0

Return

$$\text{DELVLVC} = \begin{bmatrix} TS_1 \\ TS_2 \\ TS_3 \end{bmatrix} \quad TS$$

Return

CALCN83 Entered after Average-G in P47 and P48, from "TIGON" setting

$$DV47TEMP = \frac{V}{g_{\text{body}}} + \text{DELVREF} \quad (\frac{V}{g_{\text{body}}} \text{ notation also "DELVCTL"})$$

$$TS = DV47TEMP$$

Perform "S41.1"

$$\text{DELVIMU} = TS$$

If bit 5(P48FLAG) of FIGWRD11 = 1:

If bit 11(CYCLFLAG) of FIGWRD11 = 0:

Establish "R27JOBA" (priority 17<sub>g</sub>)

If bit 5(P48FLAG) of FLGWRD11 = 0:

$VMAGI = |V|$  (writes over least significant half of UTAW,  
R1 of N78)

$HDT = unitR \cdot V$

$ALTI = |R| - K_{rpad}$

Inhibit interrupts (Tag here "CLOZUP")

$V_{gbody} = DV47TEMP$

Proceed to "SERVEXIT" (where interrupts released as part of restart  
logic)

R27JOBA Established by "CALCN83"

Set bit 11(CYCLFLAG) of FIGWRD11 = 1

Perform "R27GO"

Set bit 11(CYCLFLAG) of FIGWRD11 = 0

End of job

CALCN85 Entered after Average-G in P40 (from "P4ORCS" setting) and  
P41 (from "P4OSXTY" setting)

Perform "UPDATEVG"

$TS = V_{gtig}$

Perform "S41.1"

$V_{gbody} = TS$

Proceed to "SERVEXIT"

V64PERF Established by "R64" for a V64E. This is R64.

$TS = 0694_{vn}$

Proceed to "GOXDSPF": if terminate, proceed to "ENDEXT"  
if proceed, proceed  
otherwise, proceed to previous line

$X1 = - "MRKBUF1"$

Perform "SXTNB"



COSTH =  $TS_x$

SINTH =  $TS_y$  (due to vector store order)

UTPIT =  $-\sin^{-1} TS_z$

Perform "ARCTAN"

UTYAW = THETA

TS = 0678<sub>vn</sub>

Proceed to "GOXDSPF": if terminate, proceed to "ENDEXT"  
if proceed, proceed  
otherwise, proceed to previous line

Proceed to "ENDEXT"

V82CALL Entered from "V82PERF" for a verb 82. This is R30.

If bit 1(AVEGFLAG) of FLAGWRD1 = 1:

Establish "V82GON1" (priority 07<sub>g</sub>, with VAC area)

Perform jobs of "higher" priority if waiting ("V82GON1" will  
be since VAC-area ones of a given priority are considered  
higher than non-VAC jobs)

TS = 1644<sub>vn</sub>

Proceed to "GOXDSPF": if terminate, set bit 5 of EXTVBACT = 0  
and End of job  
if proceed, set bit 5 of EXTVBACT = 0  
and End of job  
otherwise, proceed to previous line

OPTIONX = 2 (Average-G not running if come here)

OPTIONX+1 = 1 (means computation for this vehicle)

TS = 0412<sub>vn</sub>

Proceed to "GOXDSPF": if terminate, proceed to "ENDEXT"  
if proceed, proceed  
otherwise, proceed to previous line

Call "TICKTEST" in 0.08 seconds

Proceed to "V82GOFLP"

V82GOFLP

V82FLAGS = 0

Establish "V82GOFF1" (priority 07<sub>g</sub>, with VAC area)



If bits 2-1 of V82FLAGS = 00<sub>2</sub>: (set after data from "V82GOFF1")

Delay 1 second (by putting job to sleep via "DELAYJOB")

Proceed to second previous line (recheck bits 2-1)

TS = 1644<sub>vn</sub>

Proceed to "GOXDSPF": if terminate, set bit 5 of EXTVBACT = 0  
and End of job  
if proceed, set bit 5 of EXTVBACT = 0  
and End of job  
otherwise, proceed to "V82GOFLP"

V82GON1 Established at start of "V82CALL" if Average-G on.

Set  $R_{one} = R$  and  $V_{one} = V$  (both from same Average-G cycle)

Perform "SR30.1"

If MODREG = 11:

Perform "INTSTALL"

Perform "DELRSP" (continues below if MODREG  $\neq$  0, as it should)

If bit 5 of EXTVBACT = 0: (means display response received)

Proceed to "ENDEXT"

Delay 1 second (by putting job to sleep via "DELAYJOB")

Proceed to "V82GON1"

V82GOFF1 Established by "V82GOFLP" if Average-G off.

DSPTMX = 0

TS = 0616<sub>vn</sub>

Proceed to "GOXDSPF": if terminate, proceed to "ENDEXT"  
if proceed, proceed  
otherwise, proceed to previous line

TS = DSPTMX

If DSPTMX = 0:

TS = T<sub>now</sub>

T<sub>decl</sub> = TS

T<sub>start82</sub> = TS (note that time origins remain present time, not N16 time)

If  $|\text{OPTIONX}+1| \leq 1$ :

Perform "CSMPREC"

If  $|\text{OPTIONX}+1| > 1$ :

Perform "LEMPREC"

$\underline{R}_{\text{one}} = \underline{R}_{\text{att}}$  (Tag here "BOTHSHIP")

$\underline{V}_{\text{one}} = \underline{V}_{\text{att}}$

Perform "SR30.1"

If MODREG = 0:

Perform "INTSTALL"

Perform "DELR SPL" (to next line if MODREG = 0, as it should)

$T_{\text{start82}} = T_{\text{now}} - T_{\text{start82}}$

If mTPER = 0: (i.e. perigee not  $\geq K_{\text{minpere}}$ )

$\text{TFF} = \text{TFF} + T_{\text{start82}}$

$\text{V82FLAGS} = 00002_8$  (bit 2 = 1)

End of job

$\text{mTPER} = \text{mTPER} + T_{\text{start82}}$

$\text{V82FLAGS} = 00001_8$  (bit 1 = 1)

End of job

TICKTEST Called near end of "V82CALL" if Average-G not on.

If bit 5 of EXTVBACT = 0: (means display response received)

Establish "ENDEXT" (priority  $25_8$ )

End of task

Call "TICKTEST" in 1 second

If bits 2-1 of V82FLAGS =  $00_2$ : (set after data from "V82GOFF1")

End of task

If bits 2-1 of V82FLAGS = 01<sub>2</sub>: (perigee  $\geq K_{\text{minpere}}$ )

$$mTPER = mTPER + K_{\text{lsec}}$$

End of task

If bits 2-1 of V82FLAGS = 10<sub>2</sub>: (perigee not  $\geq K_{\text{minpere}}$ )

$$TFF = TFF + K_{\text{lsec}}$$

End of task (If bits = 11<sub>2</sub>, TFF incremented by 0.03 seconds, but this should not be found)

SR30.1 Entered from "V82GON1" and "V82GOFF1"

$$TS = K_{\text{ldrtmue}}$$

Perform "TFFCONIC"

$$TS_1 = 1 + \sqrt{|1 - TFFALFA \cdot TFFNP|} \quad (\text{corresponds to } (1 + e)) \quad B2$$

$$RPER = TFFNP / TS_1$$

$$TS = TS_1 \cdot TFFldALF$$

If  $TS > 0$ :

If  $|TS| \leq K_{\text{rapmax}}$ :

$$RAPO = TS$$

Skip next line

$$RAPO = K_{\text{rapmax}}$$

$$HAPOX = (RAPO - K_{\text{rpad}}), \text{ limited } \leq K_{\text{maxnm}}$$

$$TS_2 = RPER - K_{\text{rpad}}$$

$$HPERX = TS_2, \text{ limited } \leq K_{\text{maxnm}}$$

If  $TS_2 - K_{\text{minpere}} < 0$ :

$$mTPER = +0$$

If  $TS_2 - K_{\text{minpere}} \geq 0$ :

$$TS = RPER$$

Perform "CALCTPER"

$$mTPER = - TS$$

$$TS = K_{\text{minpere}} + K_{\text{rpad}}$$

Perform "CALCTFF"

$$TFF = -TS$$

Return

DELR SPL Entered if MODREG = 00 or 11 (from "V82GOFF1" or "V82GON1")

If  $TFF \geq 0$ , proceed to "CANTDO" (have gone past interface altitude)

If  $TFF = -MAX$ , proceed to "CANTDO" (don't intersect interface alt.)

$$TS = \sqrt{2 (TFRMAG - TFRTERM) / (TFRMAG TFRTERM)} - TFFVSQ$$

$$VTERM = TS / TFFdRTMU$$

$$GAMTERM = \cos^{-1} (\sqrt{TFFNP} / (TS TFRTERM)) \quad (\text{argument limited } < 1)$$

$$COSDELF = 1 - 2 | TFFTEM / (TFRMAG TFRTERM (1 + TFFX)) |$$

Perform "AUGEKUGL"

$$THETAPR = \cos^{-1} (COSDELF) + PHIE$$

$$LAT = LATSPL$$

$$LONG = LINGSPL$$

$$ALT = DELVLVC_x$$

$$ALT = 0$$

$$TS = T_{\text{pptm}}$$

If bit 6(V37FLAG) of FLAGWRD7 = 0: (i.e. Average-G not running)

$$TS = T_{\text{start82}}$$

$$TS = TS + T_{\text{ent}} - TFF$$

Set bit 13(ERADCOMP) of FLAGWRD1 = 0

Perform "LALOTORV"

$$RSPmRREC = THETAPR - \cos^{-1} (\text{unitR}_{\text{one}} \cdot \text{unitALPHA}\underline{V}) \quad (+ \text{ if overshoot})$$

Perform "INTWAKE" (starting at 3rd from last line, awaken jobs)

Return

# CANTDO

LAT = LATSPL

LONG = LNGSPL

ALT = DELVLVC<sub>x</sub>

ALT = 0

TS = T<sub>pptm</sub>

If bit 6(V37FLAG) of FLAGWRD7 = 0:

TS = T<sub>start82</sub>

Set bit 13(ERADCOMP) of FLAGWRD1 = 0

Perform "LALOTORV"

RSPmRREC = - cos<sup>-1</sup> (unitR<sub>one</sub> · unitALPHA<sub>V</sub>)

Perform "INTWAKE" (starting at 3rd from last line, awaken jobs)

Return

AUGEKUGL Entered from "DELR SPL" with VTERM in OD and GAMTERM in 2D.

TS = VTERM - K<sub>v2lk</sub>

If TS < 0:

PDK1 = K<sub>yklk2<sub>-10</sub></sub> (VTERM - K<sub>v32k<sub>-10</sub></sub>) + K<sub>cklk2<sub>-10</sub></sub>

PDK2 = K<sub>yklk2<sub>-4</sub></sub> (VTERM - K<sub>v32k<sub>-4</sub></sub>) + K<sub>cklk2<sub>-4</sub></sub>

Proceed to "PHICALC-11"

TS = TS - K<sub>v32k<sub>-6</sub></sub> (notation also "V(3K)" )

If TS < 0:

PDK1 = K<sub>yklk2<sub>-8</sub></sub> (VTERM - K<sub>v32k<sub>-8</sub></sub>) + K<sub>cklk2<sub>-8</sub></sub>

PDK2 = K<sub>yklk2<sub>-4</sub></sub> (VTERM - K<sub>v32k<sub>-4</sub></sub>) + K<sub>cklk2<sub>-4</sub></sub>

Proceed to "PHICALC-11"

$$TS = TS - K_{v4k}$$

If  $TS < 0$ :

$$PDK1 = K_{yklk2_{-8}} (VTERM - K_{v32k_{-8}}) + K_{cklk2_{-8}}$$

$$PDK2 = K_{yklk2_{-2}} (VTERM - K_{v32k_{-2}}) + K_{cklk2_{-2}}$$

Proceed to "PHICALC-11"

$$TS = TS - K_{v32k_{-10}} \quad (\text{notation also "V(400)"})$$

If  $TS < 0$ :

$$PDK1 = K_{yklk2_{-6}} (VTERM - K_{v32k_{-6}}) + K_{cklk2_{-6}}$$

$$PDK2 = K_{yklk2_{-2}} (VTERM - K_{v32k_{-2}}) + K_{cklk2_{-2}}$$

Proceed to "PHICALC-11"

$$PDK1 = K_{yklk2_{-6}} (VTERM - K_{v32k_{-6}}) + K_{cklk2_{-6}}$$

$$PDK2 = K_{yklk2_0} (VTERM - K_{v32k_0}) + K_{cklk2_0}$$

Proceed to "PHICALC-11"

#### PHICALC-11

$$PHIE = PDK1 / (GAMTERM - PDK2)$$

If overflow has not taken place:

If  $PHIE \geq 0$ :

If  $PHIE - K_{maxphic} < 0$ :

Skip next line

$$PHIE = K_{maxphic}$$

If  $VTERM - K_{v26k} < 0$ :

$$T_{ent} = (K_{tless26} / VTERM) PHIE$$

Return (to routine calling "AUGEKUGL")

$$T_{ent} = K_{tgr26con} PHIE$$

Return (to routine calling "AUGEKUGL")



R31CALL            Established by "V83PERF" and "V85PERF" for verbs 83,85:  
                 gives R31 and R34 respectively. Also established by "P37".

Establish "V83CALL" (priority 03<sub>g</sub>)

Delay 1 second (by putting job to sleep via "DELAYJOB")

If bit 12 of EXTVBACT = 0:        (set 1 at end of pass through "COMPDISP")

     Proceed to second line of "R31CALL"

If bit 4(R31FLAG) of FLAGWRD9 = 1:

     TS = 1654<sub>vn</sub>

If bit 4(R31FLAG) of FLAGWRD9 = 0:

     TS = 1653<sub>vn</sub>

Proceed to "GOXDSPF": if terminate, set bit 5 of EXTVBACT = 0  
                         and End of job  
                         if proceed, set bit 5 of EXTVBACT = 0  
                         and End of job  
                         otherwise, proceed to 5th line of "R31CALL"

V83CALL            Established by "R31CALL"

TS = T<sub>now</sub>            (tag here "STATEXTP")

If bit 6(V37FLAG) of FLAGWRD7 = 1:

     Set  $\underline{R}_{one} = \underline{R}$ ,  $\underline{V}_{one} = \underline{V}$ , and TS = T<sub>pptm</sub> (all from same Average-G  
   cycle)

BASETIME = TS        (Tag here "BOTHGO")

T<sub>decl</sub> = BASETIME

Perform "LEMPREC"

BASEOT $\underline{P}$  =  $\underline{R}_{att1}$

BASEOT $\underline{V}$  =  $\underline{V}_{att1}$

If bit 6(V37FLAG) of FLAGWRD7 = 1, proceed to "COMPDISP"

T<sub>decl</sub> = T<sub>att</sub>

Perform "CSMPREC"

BASETH $\underline{P}$  =  $\underline{R}_{att1}$

BASETH $\underline{V}$  =  $\underline{V}_{att1}$

Proceed to "HAVEBASE"

# COMPDISP

$$TS_1 = R_{att} - R_{one}$$

$$RANGE = |TS_1|$$

Shift  $TS_1$  left by one less than number of leading zeros in RANGE

$$RRATE = (V_{att} - V_{one}) \cdot unitTS_1$$

Perform "CDUTRIG"

$$TS_2+3 = CDUS$$

$$TS_2+5 = CDUT$$

$$X1 = - "TS_2"$$

Perform "SXTNB" (loads  $TS$ )

If bit 4(R3LFLAG) of FLAGWRD9 = 1:

$$TS = unitX$$

Perform "TRG\*NBSM" (starting at second line)

$$TS_1 = TS \left[ REFSMMAT \right] \quad (\text{stored in 6D})$$

$$TS_2 = unit \left( TS_1 - (TS_1 \cdot unitR_{one}) unitR_{one} \right) \quad (\text{stored in 12D})$$

$$TS_3 = (unit(R_{one} * V_{one}) * R_{one}) \cdot TS_2$$

$$RTHETA = \cos^{-1} \left( (TS_2 \cdot TS_1) \operatorname{sgn} TS_3 \right)$$

If  $(R_{one} \cdot TS_1) < 0$ : (Note that V16 monitor could observe the uncorrected value)

$$RTHETA = (1 - 2^{-28}) - RTHETA$$

If bit 5 of EXTVBACT = 0: (means response received)

If MODREG  $\neq$  37:

Proceed to "ENDEXT"

Perform "CLEARMRK"

Proceed to 2nd line of "GOTOPOOH"

Set bit 12 of EXTVBACT = 1

Proceed to "HAVEBASE"

## HAVEBASE

If bit 6(V37FLAG) of FLAGWRD7 = 1: (Average-G data available)

Set  $\underline{R}_{one} = \underline{R}$ ,  $\underline{V}_{one} = \underline{V}$ , and  $TS = T_{pptm}$  (all from same Average-G cycle)

Perform "INTSTALL"

Set bit 4(CONICINT) of FLAGWRD3 = 0

$T_{decl} = TS$

If bit 6(V37FLAG) of FLAGWRD7 = 0:

$T_{decl} = T_{now}$

Perform "INTSTALL"

$\underline{RCV} = \underline{BASETHP}$

$\underline{VCV} = \underline{BASETHV}$

Set bit 4(CONICINT) of FLAGWRD3 = 1

$T_{et} = \underline{BASETIME}$

Perform "INTEGRVS"

$\underline{R}_{one} = \underline{R}_{att}$

$\underline{V}_{one} = \underline{V}_{att}$

Perform "INTSTALL"

Set bit 4(CONICINT) of FLAGWRD3 = 1

$T_{decl} = T_{att}$

$\underline{RCV} = \underline{BASEOTP}$  (Tag here "OTHINT")

$\underline{VCV} = \underline{BASEOTV}$

$T_{et} = \underline{BASETIME}$

Perform "INTEGRVS"

Proceed to "COMPDISP"

R36 Established by "V90PERF" for verb 90. This is R36.

$\underline{DSPTMX} = T_{ig}$

TS = 0616<sub>vn</sub>

Proceed to "GOXDSPF": if terminate, proceed to "ENDEXT"  
if proceed, proceed  
otherwise, proceed to previous line

TS = DSPTEMX

If DSPTEMX = 0:

TS = T<sub>now</sub>

T<sub>decl</sub> = TS

Perform "LEMPREC" (Tag here "R36A")

$V_{pass36} = V_{att}$

$UNP_{36} = unit(V_{att} * unitR_{att})$

T<sub>decl</sub> = T<sub>att</sub>

Perform "CSMPREC"

$UNA_{36} = unit(V_{att} * unitR_{att})$

RANGE =  $R_{att} \cdot UNP_{36}$  (here is cross-range)

RRATE =  $V_{att} \cdot UNP_{36}$  (here is cross-range)

RRATE2 =  $V_{pass36} \cdot UNA_{36}$

TS = 0696<sub>vn</sub>

Proceed to "GOXDSPF": if terminate, proceed to "ENDEXT"  
if proceed, proceed to "ENDEXT"  
otherwise, proceed

Proceed to second line of "R36"

S61.2 Entered from "NEWNRVN"

TS = K<sub>ldrtmu</sub>

If  $C_{emsalt} - K_{290kft} \geq 0$ :

TS = K<sub>ldrtmue</sub>

Perform "TFFCONIC"

TS = K<sub>rtrial</sub>

Perform "CALCTFF"

$$\text{COSDELF} = 1 - 2 \sqrt{\text{TFFTEM} / (\text{TFRMAG} \text{TFRTERM} (1 + \text{TFFX}))}$$

$$\text{SINDELF} = (\sqrt{1 - \text{COSDELF}^2}) \text{sgn TFFTEM}$$

$$\text{URH} = (\text{URONE} * \text{UNI}) \text{SINDELF} + \text{URONE} \text{COSDELF}$$

$$\text{ALPHAV}_z = \text{URH} \cdot \text{C}_{\text{unitw}}$$
Perform "GETERAD"
$$\text{TS} = \text{ERADM} + \text{C}_{\text{emsalt}}$$
Perform "CALCTFF"
$$\text{TTE1} = \text{TS}$$

$$\text{TTE2} = - (\text{TTE1} + \text{MM})$$

$$\text{TTE} = \text{TTE2} + \text{T}_{\text{pptm}}$$

$$\text{COSDELF} = 1 - 2 \sqrt{\text{TFFTEM} / (\text{TFRMAG} \text{TFRTERM} (1 + \text{TFFX}))}$$

$$\text{SINDELF} = (\sqrt{1 - \text{COSDELF}^2}) \text{sgn TFFTEM}$$

$$\text{URH} = (\text{URONE} * \text{UNI}) \text{SINDELF} + \text{URONE} \text{COSDELF}$$

$$\text{ALPHAV}_z = \text{URH} \cdot \text{C}_{\text{unitw}}$$
Perform "GETERAD"
$$\text{TS} = \cos^{-1} (\text{URH} \cdot \underline{R}_t)$$

$$\text{DTEAROT} = \text{K}_{\text{ktetal}} \text{TS} + \text{TTE1}$$
Perform "EARROT2"
$$\text{TS} = \cos^{-1} (\text{URH} \cdot \underline{R}_t)$$

$$\text{DTEAROT} = \text{K}_{\text{ktetal}} \text{TS} + \text{TTE1} \quad (2\text{nd time})$$
Perform "EARROT2"
$$\text{RTGO} = \cos^{-1} (\text{URH} \cdot \underline{R}_t)$$

$$\text{TS} = \sqrt{2(\text{TFRMAG} - \text{TFRTERM}) / (\text{TFRMAG} \text{TFRTERM}) - \text{TFFVSQ}}$$

$$\text{VTERM} = \text{TS} / \text{TFFdRTMU}$$

$$\text{GAMTERM} = \cos^{-1} \left( \sqrt{\text{TFFNP}} / (\text{TS} \text{TFRTERM}) \right) \quad (\text{argument limited } < 1)$$

$$\text{VIO} = \text{VTERM} + \text{K}_{\text{vemscon}} / (\text{GAMTERM} \text{VTERM})$$

$$\text{TFRTERM} = \text{ERADM} + \text{K}_{\text{minpere}} \quad (\text{K}_{\text{minpere}} \text{ also notation "300KFT"})$$

$$\text{TS} = \sqrt{2(\text{TFRMAG} - \text{TFRTERM}) / (\text{TFRMAG} \text{TFRTERM}) - \text{TFFVSQ}}$$

$$\text{VTERM} = \text{TS} / \text{TFFdRTMU}$$

$$GAMAE = \cos^{-1} \left( \sqrt{TFFNP} / (TS \text{ TFRTERM}) \right) \quad (\text{argument limited } < 1)$$

$$VBARSQ = \left( (VTERM - K_{36kf}) / K_{20kf} \right)^2$$

$$TS = \frac{K_{r2} (K_{m6p05d} + K_{r1} VBARSQ + GAMAE)}{(VBARSQ / K_{r4}) + 1} + K_{r3}$$

If TS < 0:

$$TS = 0$$

$$GMAX = TS$$

$$TS = ERADM + K_{400kft}$$

Perform "CALCTFF"

$$COSDELF = 1 - 2 \left| TFFTETM / (TFRMAG \text{ TFRTERM} (1 + TFFX)) \right|$$

$$SINDELF = (\sqrt{1 - COSDELF^2}) \text{ sgn } TFFTETM$$

$$URH = (URONE * UNI) SINDELF + URONE COSDELF$$

$$ALPHAV_z = URH \cdot C_{unitw}$$

Perform "GETERAD"

$$TFRTERM = ERADM + K_{400kft}$$

$$TS = \sqrt{2(TFRMAG - TFRTERM) / (TFRMAG \text{ TFRTERM}) - TFFVSQ}$$

$$VPRED = TS / TFFdRTMU$$

$$GAMMAEI = \left( -\cos^{-1} \left( \sqrt{TFFNP} / (TS \text{ TFRTERM}) \right) \right)_{sp} \quad (\text{argument limited } < 1)$$

$$JJ = RTGO_{sp} \quad (\text{for telemetry, one cell beyond GAMMAEI})$$

Return (to routine calling "S61.2")

#### TFFCONIC

$$TFFdRTMU = TS$$

$$TFRMAG = |R_{one}|$$

$$V_{onepr} = TFFdRTMU V_{one}$$

$$TFFNP = \left| (TFRMAG V_{onepr}) * unitR_{one} \right|^2$$

$$TFFVSQ = -|V_{onepr}|^2$$

$$TFFALFA = 2 / TFRMAG + TFFVSQ$$



$$TFFRTALF = \sqrt{TFFALFA}$$

$$TS = TFFRTALF^2 \text{ sgn } TFFALFA$$

If  $TS = 0$ :

$$TFFldALF = 0$$

If  $TS \neq 0$ :

$$TFFldALF = 1/TS$$

Return

#### CALCTPER

Set bit 1(TFFSW) of FLAGWRD7 = 1 (means perigee time desired)

Proceed to second line of "CALCTFF"

#### CALCTFF

Set bit 1(TFFSW) of FLAGWRD7 = 0 (means perigee time not desired)

TFRTERM = TS (TS set when enter with perigee/terminal radius)

$$TS_1 = 2 - TFRTERM TFFALFA$$

$$TS_2 = TFRTERM TS_1 - TFFNP$$

If bit 1(TFFSW) of FLAGWRD7 = 1: (means perigee time desired)

$$TS_2 = 0$$

If  $TS_2 < 0$ :

TS = +MAX (no free-fall conic to TFRTERM)

Return

$$TFFQ2 = -\sqrt{TS_2}$$

$$TFFQ1 = \frac{V_{onepr}}{R_{one}}$$

If  $TFFQ1 < 0$ : (vehicle "inbound")

$$TFFNUM = TFRTERM - TFRMAG$$

$$TFFDEN = TFFQ1 + TFFQ2$$

If  $TFFQ1 \geq 0$ :

$$TFFNUM = TFFQ2 - TFFQ1$$

$$TFFDEN = TS_1 - TFRMAG TFFALFA$$

If  $|TFFDEN| < K_{b22im}$ :

If  $TFFALFA > 0$ :

$TFFRZ = 0$

Proceed to second line of "TFFELL"

$TS = 0$  (near perigee for hyperbola or parabola)

Return

$TS = TFFNUM \ TFFRTALF / TFFDEN$

If  $|TS| \geq 1$ :

Proceed to "TFFELL"

$TFFZ = TFFNUM / TFFDEN$

$TFFTEM = TFFNP \ TFFZ^2 \ \text{sgn} \ TFFZ$

$TFFX = TFFALFA \ TFFZ^2$

$TS = K_{tx0} + K_{tx1} \ TFFX + K_{tx2} \ TFFX^2 + K_{tx3} \ TFFX^3 + K_{tx4} \ TFFX^4 + K_{tx5} \ TFFX^5$

$TS_1 = TFFZ (TFRMAG + TFRTERM - 2 \ TFFZ^2 \ TS)$

If  $TS_1 \geq 0$ , proceed to "ENDTFF"

If  $TS_1 \ \text{sgn} \ TFFQ1 \geq 0$ , proceed to "ENDTFF"

If  $TFFldALF \leq 0$ , proceed to "ENDTFF"

$TS_1 = TS_1 + K_{2pib5} \ TFFldALF / TFFRTALF$  (corrects for period)

Proceed to "ENDTFF"

#### ENDTFF

If overflow has taken place since start of TFFZ calculation:

$TS = +MAX$

Return

$TS = TFFdRTMU \ TS_1$  ( $TS_1$  scaled B45)

Return

#### TFFELL

$TFFRZ = TFFDEN / TFFNUM$

TFFX = TFFldALF TFFRZ<sup>2</sup>, with magnitude limited <1.

TFFTEM = TFFNP TFFldALF sgn ( TFFRZ TFRMAG + TFFQ1)

TS = K<sub>tx0</sub> + K<sub>tx1</sub> TFFX + K<sub>tx2</sub> TFFX<sup>2</sup> + K<sub>tx3</sub> TFFX<sup>3</sup> + K<sub>tx4</sub> TFFX<sup>4</sup> +  
K<sub>tx5</sub> TFFX<sup>5</sup>

TS<sub>1</sub> = 2 (TFFX TS - 1) TFFRZ TFFldALF

TS<sub>1</sub> = TFFldALF ( TS<sub>1</sub> +  $\frac{1}{2}$  K<sub>2pib5</sub>  $\sqrt{\text{TFFldALF}}$  - (TFFQ2 - TFFQ1) )

If overflow has taken place since start of TFFTEM calculation:

TS = +MAX

Return

TS = TFFdRTMU TS<sub>1</sub>

Return

## Quantities in Computations

See also list of major variables and list of routines

ALPHAV: See Coordinate Transformations. Only the Z component is required for "GETERAD".

ALT: See Coordinate Transformations. It is loaded briefly in "DELR SPL" with the X component of DELVLVC because this is the "third component" of LATSPL and LINGSPL and a vector operation is used (this transient loading might be noticed if a monitor verb were being used with a noun that displays ALT, and therefore is shown in the programmed equations).

ALTI: See Boost Computations.

BASEOTP: Value of OWS position vector at BASETIME computed in "V83CALL", scale factor B29, units meters. Subsequent computations of OWS position in R31/R34 start with this position and perform precision integration (if Average-G is on) or conic integration (if Average-G is off).

BASEOTV: Value of OWS velocity vector at BASETIME computed in "V83CALL", scale factor B7, units meters/centi-second. See BASEOTP.

BASETHP: Value of CSM position vector at BASETIME computed in "V83CALL", scale factor B29, units meters. If Average-G is on, the quantity is not used, but instead state vector information from Average-G is employed. If Average-G is off, however, subsequent computations of CSM position in R31/R34 start with this position and perform the integration.

BASETHV: Value of CSM velocity vector at BASETIME computed in "V83CALL", scale factor B7, units meters/centi-second. See BASETHP.

BASETIME: Value of time tag associated with BASEOTP/V and BASETHP/V, scale factor B28, units centi-seconds.

C<sub>emsalt</sub>: Erasable memory constant, program notation "EMSALT", scale factor B29, units meters, giving the altitude above the Fischer ellipse where the Entry Monitor System is to be initialized. The latitude argument for determining the Fischer ellipse value is determined from  $K_{rtrial}$ .

C<sub>unitw</sub>: See General Program Control.

CDUS, CDUT: See Optics Computations.

COSDEL: Cosine of the "transfer angle" between TFRMAG and TFRTERM, scale factor B1. Program notation "CDEL/2", stored in push-down list location 14D.

COSTH: See Coordinate Transformations.

DELVIMU: Velocity increment in control axes computed in "CALCN83" for P47 and P48, scale factor B7, units meters/centi-second. The value can be zeroed by entry of a recycle verb at the display (P47) of the parameter, or at any P48 display. The cumulative sum of velocity, in reference coordinates, is kept in  $V_{gbody}$  cells (note that in this case they are not control coordinates).

DELVLC: See Burn Control.

DELVREF: See General Program Control.

DSPTEMX: Display temporary cells used to contain time for e.g. R30 and R36, scale factor B28, units centi-seconds, displayed by N16. The cells are the same as DSPTEM2+1<sub>dp</sub>: the "X" signifies "extended verbs".

DTEAROT: See Coordinate Transformations.

DV47TEMP: Value of summed velocity increment loaded at start of "CALCN83" and transferred to  $V_{gbody}$  at end, scale factor B7, units meters/centi-second in reference coordinates. Used for restart protection purposes (for the summing process).

ERADM: See Coordinate Transformations.

EXTVBACT: See Verb Definitions.

GAMAE: Value of flight path angle at 300000 feet, used to compute GMAX, "positive down", stored in push-down list location 2D, units of revolutions.

GAMMAEI: Single precision value of flight path angle at 400000 feet, scale factor B0, units revolutions (will be negative). It is the "angle between the inertial velocity vector and local horizontal" at this altitude above the Fischer radius. For telemetry convenience, the "least significant half" of GAMMAEI is JJ, loaded at the end of "S61.2" with RTGO<sub>sp</sub>.

GAMTERM: Value of flight path angle e.g. at TFRTERM, scale factor B0, units revolutions, computed in a subroutine. Angle is measured as described for GAMMAEI, except that it is positive (hence could be considered as absolute value of flight path angle). When used as a communication cell with "AUGEKUGL", is retained in push-down list location 2D, and is at the same point as VTERM.

GMAX: Value of predicted maximum drag, scale factor B14, units of 0.01 g's. Computed double precision, but least significant half is not used in generating the display, so is written over by the most significant half of VMAGI after start of "CM/POSE" computations.

HAPOX: Value of apocenter altitude computed in "SR30.1", scale factor B29, units meters, displayed by N44. The "X" signifies "extended verb", since cell distinct from one in e.g. P30, HAPO.

HDOT: See Boost Computations.

HPERX: Value of pericenter altitude computed in "SR30.1", scale factor B29, units meters, displayed by N44. See HAPOX.

JJ: See Entry Computations. Loaded with RTGO<sub>sp</sub> at end of "S61.2" for telemetry purposes (see GAMMAEI).

K<sub>ldrtmu</sub>: Constant, program notation "1/RTMU", scale factor B-17, value  $0.5005750271E-5 \times 2^{17}$ . Constant used only for initialization of TFFdRTMU in "S61.2" computations, and corresponds to the reciprocal of the square root of  $(3.98603223E10 \times 1.0012)$  (with binary scale factor), where first term is earth  $\mu$  (meters<sup>3</sup>/cs<sup>2</sup>) and second is a factor for "near earth trajectory", representing an increase of 0.12%. If  $C_{emsalt} \gg K_{290kft}$ , however, then K<sub>ldrtmue</sub> is used.

K<sub>ldrtmue</sub>: Constant, program notation "1/RTMUE", scale factor B-17, value  $0.50087529E-5 \times 2^{17}$ , corresponding to the reciprocal of the square root of unmodified earth  $\mu$  (same as K<sub>mutab<sub>6</sub></sub> in Conic Routines).

K<sub>lsec</sub>: Constant, program notation "1/SEC", scale factor B28, units centi-seconds. Value is  $100 \times 2^{-28}$ , corresponding to 1 second (as used). It is added to mTPER or TFF in "TICKTEST" to update time of display (if Average-G is not running) so as to reflect more closely the "time to" the indicated trajectory point (both times normally would be minus and the addition causes them to "count down").

K<sub>2pib5</sub>: Constant, program notation "PI/16", scale factor B5, value  $3.141592653 \times 2^{-4}$ , corresponding to  $2\pi \times 2^{-5}$  (also used with scale factor B4, shown as  $\frac{1}{2} K_{2pib5}$ , to give effect of  $\pi$ ).

K<sub>20kf</sub>: Constant, program notation "20KFT/S", scale factor B6, units meters/centi-second. Value is  $121.92 \times 2^{-7}$ , corresponding to  $20,000 \times 0.3048 \times 10^{-2} \times 2^{-6}$ , where first term is "basic" value in fps, second converts from feet to meters, third converts to centi-seconds, and fourth is scale factor.

K<sub>36kf</sub>: Constant, program notation "36KFT/S", scale factor B7, units meters/centi-second. Value is  $109.728 \times 2^{-7}$ , corresponding to  $36,000 \times 0.3048 \times 10^{-2} \times 2^{-7}$ , where first term is "basic" value in fps, second converts from feet to meters, third converts to centi-seconds, and fourth is scale factor.

K<sub>290kft</sub>: Constant, program notation "290KFT", scale factor B29, units meters. Value is  $88392 \times 2^{-29}$ , corresponding to  $290,000 \times 0.3048 \times 2^{-29}$ , where first term is "basic" value in feet, second converts from feet to meters, and third is scale factor.



$K_{400kft}$ : Constant, program notation "400KFT", scale factor B29, units meters. Value is  $121920 \times 2^{-29}$ , corresponding to  $400,000 \times 0.3048 \times 2^{-29}$ , where first term is "basic" value in feet, second converts from feet to meters, and third is scale factor.

$K_{b22im}$ : Constant, program notation "LIM(-22)", octal value 37777<sub>8</sub> 37700<sub>8</sub>, but used in program in such a way (a forcing of overflow) that effective value corresponds to 100<sub>8</sub>, or  $2^{-22}$ . If TFFDEN scaling is B3, this represents a "true" value of  $2^{-19}$ ; if TFFDEN scaling is B16, this represents a "true" value of  $2^{-6}$ .

$K_{cklk2}$  (i = -10 to 0 in increments of 2): Value of additive constant i in computing PDK1 (i = -10, -8, -6) and PDK2 (i = -4, -2, 0) in "AUGEKUGL", program notation (for i = 0) "CKLK2", scale factors all B0.

For i = 0, -2, -4, constants are in units of revolutions:

<u>i</u>	<u>Value</u>	<u>Equivalent to</u>
0	6.6666666E-3	2.4 degrees
-2	2.7777777E-3	1.0 degrees
-4	-8.8888888E-3	-3.2 degrees

For i = -6, -8, -10, constants are in units of revolutions/360, to give a PHIE result in units of revolutions. Conversion value between nautical miles and (rev/360) is  $7.776E6$ , which is  $21600 \times 360$ . The 21600 in turn is  $2\pi \times 3437.7468$  nmi (about 6366707 meters as contrasted with 6373338 meters for pad radius and scaling of RSPmRREC for display). Values are:

<u>i</u>	<u>Value</u>	<u>Equivalent to</u>
-6	3.08641975E-4	2400/7.776E6 (2400 nmi)
-8	3.08641975E-4	Same as i = -6
-10	7.07304526E-4	5500/7.776E6 (5500 nmi)

$K_{ktetal}$ : Constant, program notation "KTETAL", scale factor B28, units centi-seconds/revolution. Value is  $42.1844723 \times 2^{-14}$ , corresponding to  $1100 \times 100 \times 2\pi \times 2^{-28}$ , where first term is "basic" value (seconds/radian), second converts to centi-seconds, third converts to revolutions, and fourth is scale factor. Cf.  $K_{kteta}$  in Entry Computations.

$K_{m6p05d}$ : Constant, program notation "-6.05DEG", scale factor B0, units revolutions. Value is -0.016805556, corresponding to -6.05 degrees.

$K_{maxnm}$ : Constant, program notation "MAXNM", scale factor B29, units meters. Octal value is 01065<sub>8</sub> 05603<sub>8</sub>, corresponding to 18519.814 kilometers, or about 9999.8995 nmi (DSKY routine display 9999.9 nmi).

$K_{maxphic}$ : Constant, program notation "MAXPHIC", scale factor B0, units revolutions. Value is 0.09259298, corresponding to  $\sim 2000/21600$ , where first term is value in nmi and second (see  $K_{cklk2}$ ) converts to rev.

$K_{minpere}$ : Constant, program notation "MINPERE", scale factor B29, units meters. Value is  $91440 \times 2^{-29}$ , corresponding to  $300,000 \times 0.3048 \times 2^{-29}$ , where first term in "basic" value in feet (about 49.37365 nmi), second converts to meters, and third is scale factor. Constant also referenced by tag "300KFT".

$K_{r1}$ : Constant, program notation "KR1", scale factor B-2, units revolutions. Value is  $-0.026666667$ , corresponding to  $-2.4^\circ$ .

$K_{r2}$ : Constant, program notation "KR2", scale factor B18, octal value 214508 000018. Without the non-zero least significant half, value corresponds to  $4 \times 360 \times 10^2 \times 2^{-18}$  (or  $9000 \times 2^{-14}$ ), where first term is "basic" value, second converts from revolutions to degrees, third scales to 0.01 g, and fourth is scale factor.

$K_{r3}$ : Constant, program notation "KR3", scale factor B14, value  $1000 \times 2^{-14}$ . Value corresponds to  $10 \times 10^2 \times 2^{-14}$ , where the first term is equal to  $10 - 10(x - .3)$  for x, the nominal L/D, equal to 0.3; the second term scales to 0.01 g, and the third is the scale factor: formula obtained from previous programs where  $x \neq 0.3$ .

$K_{r4}$ : Constant, program notation "KR4", scale factor B-2, value 0.833333333. Since it divides VBARSQ, has an effect equivalent to multiplying it by  $(5/6)^{-1} \times 2^2 = 4.80$ .

$K_{rapmax}$ : Constant, program notation "NEARONE", scale factor B29, units meters. Value is  $(1 - 2^{-28})$ , corresponding to  $(2^{29} - 2)$  meters.

$K_{rpad}$ : See Burn Control.

$K_{rtrial}$ : Constant, program notation "RTRIAL", scale factor B29, units meters. Value stored in memory is binary equivalent of 6460098 meters: according to program comments, this is intended to be 284643 feet above a "pad" of 6373338 meters (actually is about 284645.67 feet).

$K_{tgr26con}$ : Constant, program notation "TGR26CON", scale factor B28, units centi-seconds/revolution. Used to compute  $T_{ent}$  from PHIE for velocity equal to or greater than 26,000 fps (hence the notation). Value is  $7.2E5 \times 2^{-28}$ , corresponding to  $(1/3) \times 21600 \times 10^2 \times 2^{-28}$ , where first term is "basic" value (for PHIE in nmi and answer in seconds), second converts from revolutions to nmi (see  $K_{cklk2}$ ), third converts to centi-seconds, and fourth is scale factor.

$K_{tless26}$ : Constant, program notation "TLESS26", scale factor B35, value  $5.70146688E7 \times 2^{-35}$ . Used to compute  $T_{ent}$  from PHIE and VTERM for velocity less than 26,000 fps (hence the notation). Decimal value corresponds to  $8660 \times 21600 \times 0.3048$ , where first term is "basic" value (for PHIE in nmi and velocity in fps), second converts from revolutions to nmi (see  $K_{cklk2}$ ), and third converts velocity distance unit from meters to feet (conversion of time unit to seconds from centi-seconds not done since time units of  $T_{ent}$  are centi-seconds).

$K_{tx0} - K_{tx5}$ : Coefficients in a series expansion used in time-of-flight calculation, all with scale factor B0. For positive argument, function is  $(\sqrt{x} - \tan^{-1} \sqrt{x}) / x^{3/2}$ ; for negative argument, function is  $(\sqrt{-x} - \tanh^{-1} \sqrt{-x}) / -x^{3/2}$ . According to the program comments, the polynomial was fitted for 278 flight over the range 0 - 1, and has a maximum error of 2E-5. Program notation "T(X) +2".

Constant	Value	Infinite-Series Value
$K_{tx0}$	3.333333333E-1	1/3
$K_{tx1}$	-1.999819135E-1	-1/5
$K_{tx2}$	1.418148467E-1	1/7
$K_{tx3}$	-1.01310997E-1	-1/9
$K_{tx4}$	5.609004986E-2	1/11
$K_{tx5}$	-1.536156925E-2	-1/13

$K_{v21k}$ : Constant, program notation "V(21K)", scale factor B7, units meters/centi-second. Value is  $64.008 \times 2^{-7}$ , corresponding to  $21000 \times 0.3048 \times 10^{-2} \times 2^{-7}$ , where first term is value in fps, second converts to meters, third to centi-seconds, and fourth is scale factor.

$K_{v26k}$ : Constant, program notation "V(26K)", scale factor B7, units meters/centi-second. Value is  $79.248 \times 2^{-7}$ , corresponding to  $26000 \times 0.3048 \times 10^{-2} \times 2^{-7}$ , where first term is value in fps, second converts to meters, third converts to centi-seconds, and fourth is scale factor.

$K_{v32k_i}$  ( $i = -10$  to 0 in increments of 2): Values of subtractive velocity  $i$  constant in computing PDK1 ( $i = -10, -8, -6$ ) and PDK2 ( $i = -4, -2, 0$ ) in "AUGEKUGL", program notation (for  $i = 0$ ) "V(32K)", scale factor B7, units meters/centi-second. To convert a value in fps to meters/centi-second, multiply by  $0.3048 \times 10^{-2} \times 2^{-7}$ , where first term converts to meters, second converts to centi-seconds, and third is scaling.

$i$	Notation	Value	Equivalent to
0	V(32K)	$97.536 \times 2^{-7}$	32,000 fps
-2	---	$85.344 \times 2^{-7}$	28,000 fps
-4	V(24K)	$73.152 \times 2^{-7}$	24,000 fps
-6	V(3K)	$9.144 \times 2^{-7}$	3,000 fps
-8	V(28K)	$85.344 \times 2^{-7}$	28,000 fps
-10	V(400)	$1.2192 \times 2^{-7}$	400 fps

$K_{v4k}$ : Constant, program notation "V(4K)", scale factor B7, units meters/centi-second. Value is  $12.192 \times 2^{-7}$ , corresponding to  $4000 \times 0.3048 \times 10^{-2} \times 2^{-7}$ , where first term is value in fps, second converts to meters, third converts to centi-seconds, and fourth is scale factor.

K<sub>vemscn</sub>: Constant, program notation "VEMSCN", scale factor B14, value  $-0.0389676 \times 2^{-14}$ . Value corresponds to  $(-1.51E6) \times (0.3048/100)^2 \times (1/360) \times 2^{-14}$ , where first term is "basic" value (for velocities in fps and angles in degrees), second term converts denominator to units of fps and also converts final answer from fps to meters/centi-second, third term converts angle in denominator from revolutions to degrees, and fourth term is the scale factor. Stored value is  $77777_8 76601_8$ , corresponding to about  $-0.03894043 \times 2^{-14}$ : this gives a basic value of about  $-1.508945E6$ .

K<sub>yklk2</sub> (i = -10 to 0 in increments of 2): Value of constant multiplying i velocity information in computing PDK1 (i = -10, -8, -6) and PDK2 (i = -4, -2, 0) in "AUGEKUGL", program notation (for i = 0) "YK1K2", scale factors all B-7. For i = 0, -2, -4, basic value is for data in degrees and fps, and must be multiplied by  $(100/0.3048) \times (1/360)$  to convert to proper velocity information and revolutions.

<u>i</u>	<u>Value</u>	<u>Equivalent to</u>
0	$2.59733157E-4 \times 2^7$	$2.85E-4$
-2	$9.56911636E-4 \times 2^7$	$1.05E-3$
-4	$1.11639691E-3 \times 2^7$	$1.225E-3$

For i = -6 and -10, the value of the constant is 0. For i = -8, value is  $-1.86909989E-5 \times 2^7$ . This value corresponds to  $-0.443 \times (100/0.3048) \times (1/7.776E6) \times 2^7$ , where first term is "basic" value (to give a PHIE in nautical miles), second converts velocity from meters/centi-second to feet/second, third is the conversion factor discussed with K<sub>cklk2</sub>, and fourth is scale factor.

LAT: See Coordinate Transformations.

LATSPL: Input information on target latitude, program notation "LAT(SPL)", scale factor B0, units revolutions. Can be displayed in R1 of N61, and should form part of prelaunch erasable load for use in computing RSPmRREC. Cell also used to input target information for entry equations. Information is geodetic latitude.

LNGSPL: Input information on target longitude, program notation "LNG(SPL)", scale factor B0, units revolutions. Can be displayed in R2 of N61 (see LATSPL).

LONG: See Coordinate Transformations.

MM: See Entry Preparation.

MRKBUF1: See Optics Computations. N94 loads MRKBUF1+3 and MRKBUF1+5, hence address of MRKBUF1 causes selection of proper optics angle information to be made in "SXTNB" (the use of which is required by "V64PERF").

mTPER: Value of time from per.alt., scale factor B28, units centi-seconds, program notation "-TPER". It is set 0 if HPERX is below  $K_{minpere}$ . If V82 is entered with Average-G off, time on display is updated every second by "TICKTEST" (after being corrected near end of "V82GOFF1" for computation delay since sampling time of state vector) by being incremented by 1 second every second (hence display could become positive if perigee passed, while a termination and reselection of V82 would give a negative time again, namely time until the next perigee). If Average-G is on, however, computation is recycled every second (plus computing delays) via "V82GON1", and no updating of mTPER is otherwise performed. If HPERX is below  $K_{minpere}$ , update is of TFF instead of mTPER.

OPTIONX: Cells displayed by N12, and intended for use in a fashion similar to OPTION1, OPTION2, but with extended verbs. Same cells as DSPTMX+1. Program notation for OPTIONX+1 also "OPTIONEV".

PDK1: Value of equation quantity " $K_1$ " (numerator term for PHIE in "AUGEKUGL" routine), units of " $\frac{1}{2}$  revolutions/360" (or revolutions<sup>2</sup>, so that division by revolutions gives result in revolutions), scale factor B0, stored in push-down list location 4D.

PDK2: Value of equation quantity " $K_2$ " (term in denominator for PHIE in "AUGEKUGL" routine), scale factor B0, units revolutions, stored in push-down list location 6D.

PHIE: Value of "empirically estimated entry range", scale factor B0, units revolutions, stored in push-down list location 4D. A factor of 21600 is used with constants expressed in nmi to convert to revolutions in deriving PHIE (see "AUGEKUGL" quantities).

pMGA: Value of "positive middle gimbal angle", program notation "+MGA", scale factor B0, units revolutions. Set to cause a display of -0.02 degrees if nominal flow would require a display but the bit indicating valid [REFSMAT] is not set; set to cause a display of -0.01 degrees if computation flow does not require angle display (e.g. not the final pass through rendezvous computations). Otherwise, should be a positive number in range 0-90 degrees (or 270-360 degrees). Quantity therefore shows the value of the middle gimbal angle (if positive) when the +X axis is aligned with the impulsive thrust direction.

R<sub>init</sub>: See Rendezvous Computations. Cell also used as a communication one with "GET.LVC" to define local-vertical coordinate system.

R<sub>one</sub>: Position vector used to achieve a constant state vector for use within display computations, scale factor B29, units meters. Also loaded in "NEWRVN" with present position vector for use in "S61.2" computations. Loaded in "HAVEBASE" with CSM state vector information at required time, scale factor B29, units meters, for use in "COMPDISP" for R31/R34.

R<sub>t</sub>: See Coordinate Transformations.

RANGE: Value of magnitude of vector difference between OWS and CSM position vectors computed in "COMPDISP" (for R31/R34), scale factor B29, units meters. Also used to contain out-of-plane position information, same units and scaling, computed in R36. The cell is displayed in R1 of N53, N54, and N96 (as well as being time-shared with other information).

RAPO: Value of apocenter radius computed in "SR30.1", scale factor B29, units meters, stored in push-down list location 16D.

RCV: See Orbital Integration.

RPER: Value of pericenter radius computed in "SR30.1", scale factor B29, units meters, stored in push-down list location 14D.

RRATE: Value of component of velocity difference between CSM and OWS vehicles (negative if approaching), scale factor B7, units meters/centi-second. Also used for CSM out-of-plane velocity information for R36, and displayed (with this tag) in R2 of N53, N54, and N96.

RRATE2: Value of OWS out-of-plane velocity information computed in R36, scale factor B7, units meters/centi-second, displayed in R3 of N96. Same cell used for e.g. RTHETA.

RSPmRREC: Value of downrange recovery range error (predicted flight central angle minus target central angle, scaled to revolutions), scale factor B0, units revolutions, computed in "DELRSP" for R30 computations in POO or P11. Target central angle is computed as an arc cosine in the range between 0 and  $\frac{1}{2}$  (0 and 180°), with no account taken of flight direction. Program notation "RSP-RREC". If TFF is  $\geq 0$  or -MAX, the predicted flight central angle is considered 0. Cell is displayed in R1 of N50 using K<sub>sfol15</sub> (see Noun Definitions).

RTGO: Predicted central angle between the altitude specified by C<sub>emsalt</sub> and splash, scale factor B0, units revolutions, computed in "S61.2". The most significant half is loaded into JJ for telemetry purposes at end of "S61.2".

RTHETA: Display angle information computed for R31 and R34, scale factor B0, units revolutions, displayed in R3 of N53 and N54. Also time-shared with other quantities (such as RRATE2).

SINDELf: Sine of the "transfer angle" between TFRMAG and TFRTERM, scale factor B1. Computed by the same subroutine as that used for COSDELf.

SINTH: See Coordinate Transformations.

T<sub>ent</sub>: Predicted time required to traverse the entry angle PHIE, scale factor B28, units centi-seconds, left in MPAC<sub>dp</sub> when exit from "AUGEKUGL" routine.

T<sub>et</sub>: See Orbital Integration.



$T_{start82}$ : Value of state vector time tag sampled at start of "V82GOFF1", scale factor B28, units centi-seconds. After complete "SR30.1", is replaced by  $(T_{now} - T_{start82})$ , or the "age" of the TFF or mTPER value which has been computed. This value is then used to update TFF or mTPER to give a time origin of present time (regardless of the time entered via N16). TFF and mTPER are then updated once a second in "TICKTEST" so as to maintain their status as time from "now" to the specified point.

TFF: Predicted time of free fall to specified altitude (negative if not yet there), scale factor B28, units centi-seconds. It is set to -MAX if trajectory does not intersect the specified altitude (either completely above or completely below it); it is set 0 for indeterminate computation (near perigee for hyperbola or parabola). Updated as discussed for mTPER (if mTPER cell is not updated, of course), for Average-G off (if on, computed on a periodic basis).

TFFldALF: Value of |semimajor axis| computed in "TFFCONIC", program notation "TFF1/ALF", stored in push-down list location 22D. The scale factor information partially contained in X2: if  $X2 = 0$ , scale factor is B22. Units are meters.

TFFALFA: Value of reciprocal of semimajor axis computed in "TFFCONIC", stored in push-down list location 26D. Scale factor information partially contained in X1: if  $X1 = 0$ , scale factor B-26.

TFFDEN: Value of denominator quantity, stored in push-down list location 0D. If TFFQ1 negative, scale factor is B16; if TFFQ1 positive, the scale factor is B3. In "TFFELL", used as numerator instead.

TFFdRTMU: Value of reciprocal of square root of  $\mu$  of primary body, scale factor B-17, stored in push-down list location 30D. If loaded in "S61.2", may be loaded with modified  $\mu$  (see  $K_{ldrtmu}$ ).

TFFNP: Value of semilatus rectum (p, called LCP, or lower-case p, in comments field) computed in "TFFCONIC", stored in push-down list location 28D. Scale factor information partially contained in X1: if  $X1 = 0$ , scale factor is B38. Units are meters.

TFFNUM: Value of numerator quantity, stored in TFFX cell, scale factor B29 if TFFQ1 negative; scale factor B16 if TFFQ1 positive. In "TFFELL", used as denominator instead.

TFFQ1: Intermediate quantity in time-of-flight calculation (function of present state vector), scale factor B16, stored in push-down list location 14D.

TFFQ2: Intermediate quantity in time-of-flight calculation (function of terminal state vector), scale factor B16, stored in TFFTEM cell briefly.



TFFRTALF: Square root of  $\sqrt{TFFALFA}$ , computed in "TFFCONIC", stored in push-down list location 24D. Scale factor information partially contained in X2: if  $X2 = 0$ , scale factor is B-10.

TFFRZ: Intermediate quantity ( $1/Z$ ) computed in "TFFELL", scale factor B-11, stored in TFFTEM cell.

TFFTEM: Cell used for several purposes during the performance of the time-of-flight calculations; when calculation has a normal exit, contains the value of a quantity used in subsequent calculations, with scale factor information partially contained in X1. If  $X1 = 0$ , scale factor is B59.

TFFVSQ: Value of complement of square of velocity divided by  $\mu$ , computed in "TFFCONIC", scale factor B-20.

TFFX: Argument for power series expansion in time-of-flight calculation, scale factor B0, stored in push-down list location 34D.

TFFZ: Intermediate quantity ( $Z$ ) computed in case "TFFELL" is not entered, scale factor B13, stored in TFFTEM cell.

TFRMAG: Magnitude of  $R_{one}$  computed in "TFFCONIC" and stored in push-down list location 12D as "RMAG1", scale factor B29, units meters. A normalized version, with number of shifts required contained in X1, is stored in push-down list location 32D with program tag "NRMAG". Distinction in type of value employed is not made in programmed equations description.

TFRTERM: Value of terminal radius used as input argument for time-of-flight routines. With scale factor B29, units meters, is stored in push-down list location 18D, tag "RTERM". After being shifted left by -X1 places (to give same scaling as normalized TFRMAG) is stored in NRTERM, push-down list location 16D. Distinction in type of value employed is not made in programmed equations description.

THETAPR: Value of predicted central angle of travel to target from present location, computed in "DELRSP", scale factor B0, units revolutions. Program notation "THETA(1)".

TTE: Time from arrival at altitude specified by  $C_{amsalt}$ , scale factor B28, units centi-seconds, computed in "S61.2" and in "SERVICER" (if O5GSW bit = 0) for N63 display, negative before arrival.

TTE1: Time from arrival at altitude specified by  $C_{amsalt}$ , scale factor B28, units centi-seconds. It is computed in "S61.2", and is measured from the time contained in MM (positive before arrival).

TTE2: Complement of the GET at which vehicle will arrive at  $C_{amsalt}$ , scale factor B28, units centi-seconds, computed in "S61.2" and used in "SERVICER" to update TTE.

UNA<sub>36</sub>: Unit vector computed in R36 perpendicular (i.e.  $\underline{v} * \underline{r}$ ) to the trajectory plane of the CSM (normally the "active" vehicle), scale factor B1.

UNI: See Entry Computations. Computed in "NEWRVN" also.

UNP<sub>36</sub>: Unit vector computed in R36 perpendicular (i.e.  $\underline{v} * \underline{r}$ ) to the trajectory plane of the OWS (normally the "passive" vehicle), scale factor B1.

URH: Value of unit terminal position vector computed in "S61.2", scale factor B2.

URONE: Value of unit  $\underline{R}_{one}$  computed in "NEWRVN" for use in "S61.2", scale factor B1.

UTPIT, UTYAW: See Orbital and Rendezvous Navigation.

$\underline{V}_{gbody}$ : See Burn Control. In "CALCN83" is used to keep cumulative sum of velocity increments in reference coordinates, with DELVIMU being the corresponding information in control coordinates. The notation also "DELVCTL".

$\underline{V}_{gtig}$ : See Burn Control.

$\underline{V}_{init}$ : See Rendezvous Computations. Cell also used as a communication one with "GET.LVC" to define local-vertical coordinate system.

$\underline{V}_{one}$ : Velocity vector used to achieve a constant state vector within display computations, scale factor B7, units meters/centi-second. See  $\underline{R}_{one}$ .

$\underline{V}_{onepr}$ : Value of  $\underline{V}_{one}$  multiplied by TFFdRTMU in "TFFCONIC", scale factor B-10, program notation "VONE".

$\underline{V}_{pass36}$ : Value of OWS velocity vector computed in R36, scale factor B7, units meters/centi-second.

V82FLAGS: Single precision cell used to control performance of R30 (initiated by verb 82) if Average-G is off. Initialized to zero in "V82GOFLP" and set to 1 or 2 at end of "V82GOFF1" to signify that mTPER or TFF respectively should be incremented by "TICKTEST": non-zero setting also used in "V82GOFLP" to initiate V16N44 display, since such a setting signifies that data are available for display.

VBARSQ: Value of square of normalized velocity information used to compute GMAX in "S61.2", scale factor B2, stored in push-down list location OD.

VCV: See Orbital Integration.

VIO: Value of predicted velocity at altitude specified by C<sub>emsalt</sub>, scale factor B7, units meters/centi-second, including an approximate correction for the drag loss in reaching this altitude (the nominal 0.05 g point).

VMAGI: See Boost Computations.

VPRED: Value of predicted velocity at an altitude of 400000 feet above  
Fischer radius computed in "S61.2", scale factor B7, units meters/  
centi-second.

VTTERM: Value of velocity (from vis viva integral) at TFRTERM, scale  
factor B7, units meters/centi-second, stored in push-down list  
location OD (also used as communication cell with "AUGEKUGL").

