T A B L E   O F   C O N T E N T S

```
$WAITLIST.AGC                                            # PP. 1117-1132
$LATITUDE_LONGITUDE_SUBROUTINES.AGC                      # PP. 1133-1139
$PLANETARY_INERTIAL_ORIENTATION.AGC                      # PP. 1140-1148
$MEASUREMENT_INCORPORATION.AGC                           # PP. 1149-1158
$CONIC_SUBROUTINES.AGC                                   # PP. 1159-1204
$INTEGRATION_INITIALIZATION.AGC                          # PP. 1205-1226
$ORBITAL_INTEGRATION.AGC                                 # PP. 1227-1248
$INFLIGHT_ALIGNMENT_ROUTINES.AGC                         # PP. 1249-1258
$POWERED_FLIGHT_SUBROUTINES.AGC                          # PP. 1259-1267
$TIME_OF_FREE_FALL.AGC                                   # PP. 1268-1283
$AGC_BLOCK_TWO_SELF_CHECK.AGC                            # PP. 1284-1293
$PHASE_TABLE_MAINTENANCE.AGC                             # PP. 1294-1302
$RESTARTS_ROUTINE.AGC                                    # PP. 1303-1308
$IMU_MODE_SWITCHING_ROUTINES.AGC                         # PP. 1309-1337
$KEYRUPT_UPRUPT.AGC                                      # PP. 1338-1340
$DISPLAY_INTERFACE_ROUTINES.AGC                          # PP. 1341-1373
$SERVICE_ROUTINES.AGC                                    # PP. 1374-1380
$ALARM_AND_ABORT.AGC                                     # PP. 1381-1385
$UPDATE_PROGRAM.AGC                                      # PP. 1386-1396
$RTB_OP_CODES.AGC                                        # PP. 1397-1402
$T6-RUPT_PROGRAMS.AGC                                    # PP. 1403-1405
$DAP_INTERFACE_SUBROUTINES.AGC                           # PP. 1406-1409
$DAPIDLER_PROGRAM.AGC                                    # PP. 1410-1420
$P-AXIS_RCS_AUTOPILOT.AGC                                # PP. 1421-1441
$Q_R-AXIS_RCS_AUTOPILOT.AGC                              # PP. 1442-1459
$TJET_LAW.AGC                                            # PP. 1460-1469
$KALMAN_FILTER.AGC                                       # PP. 1470-1471
$TRIM_GIMBAL_CNTROL_SYSTEM.AGC                           # PP. 1472-1484
$AOSTASK_AND_AOSJOB.AGC                                  # PP. 1485-1506
$SPS_BACK-UP_RCS_CONTROL.AGC                             # PP. 1507-1510
```

```
# THIS LGC PROGRAM IS INTENDED FOR USE IN THE LM DURING THE MANNED LUNAR LANDING MISSION OR ANY SUBSET THEREOF.
# THE DETAILS OF IMPLEMENTATION ARE SPECIFIED IN REPORT R-567, AS AMENDED.

#                                 GUIDANCE SYSTEM OPERATIONS PLAN
#                         FOR MANNED LM EARTH ORBITAL AND LUNAR MISSIONS
#                                     USING PROGRAM LUMINARY

# THIS PROGRAM AND R-567 HAVE BEEN PREPARED BY THE INSTRUMENTATION LABORATORY, MASSACHUSETTS INSTITUTE OF
# TECHNOLOGY, 75 CAMBRIDGE PARKWAY, CAMBRIDGE, MASSACHUSETTS, UNDER PROJECT 55-238-70, SPONSORED BY THE MANNED
# SPACECRAFT CENTER OF THE NATIONAL AERONAUTICS AND SPACE ADMINISTRATION, CONTRACT NAS 9-4065.

#                           THIS PROGRAM IS REFERRED TO AS LUMINARY 1A
```

```
# TABLE OF LOG CARDS
#       ASSEMBLY AND OPERATION INFORMATION
#       TAGS FOR RELATIVE SETLOC AND BLANK BANK CARDS
#       CONTROLLED CONSTANTS
#       INPUT/OUTPUT CHANNEL BIT DESCRIPTIONS
#       FLAGWORD ASSIGNMENTS
#       SUBROUTINE CALLS

# TABLE OF SUBROUTINE LOG SECTIONS
#       LUMERASE
#               ERASABLE ASSIGNMENTS
#       LNYAIDE
#               INTERRUPT LEAD INS
#               T4RUPT PROGRAM
#               RCS FAILURE MONITOR
#               DOWNLINK LISTS
#               AGS INITIALIZATION
#               FRESH START AND RESTART
#               RESTART TABLES
#               AOTMARK
#               EXTENDED VERBS
#               PINBALL NOUN TABLES
#               LEM GEOMETRY
#               IMU COMPENSATION PACKAGE
#               R63
#               ATTITUDE MANEUVER ROUTINE
#               GIMBAL LOCK AVOIDANCE
#               KALCMANU STEERING
#               SYSTEM TEST STANDARD LEAD INS
#               IMU PERFORMANCE TESTS 2
#               IMU PERFORMANCE TESTS 4
#               PINBALL GAMES BUTTONS AND LIGHTS
#               R60,R62
#               S-BAND ANTENNA FOR LM
#       LEMP20S
#               RADAR LEADIN ROUTINES
#               P20-P25
#       LEMP30S
#               P30,P37
#               P32-P35, P72-P75
#               GENERAL LAMBERT AIMPOINT GUIDANCE
#       KISSING
#               GROUND TRACKING DETERMINATION PROGRAM - P21
#               P34-P35, P74-P75
#               R31
#               P76
#               R30
#               STABLE ORBIT - P38-P39
```

```
#	FLY
#		BURN, BABY, BURN -- MASTER IGNITION ROUTINE
#		P40-P47
#		THE LUNAR LANDING
#		THROTTLE CONTROL ROUTINES
#		LUNAR LANDING GUIDANCE EQUATIONS
#		P70-P71
#		P12
#		ASCENT GUIDANCE
#		SERVICER
#		LANDING ANALOG DISPLAYS
#		FINDCDUW -- GUIDAP INTERFACE
#	LEMP50S
#		P51-P53
#		LUNAR AND SOLAR EPHEMERIDES SUBROUTINES
#	SKIPPER
#		DOWN-TELEMETRY PROGRAM
#		INTER-BANK COMMUNICATION
#		INTERPRETER
#		FIXED-FIXED CONSTANT POOL
#		INTERPRETIVE CONSTANTS
#		SINGLE PRECISION SUBROUTINES
#		EXECUTIVE
#		WAITLIST
#		LATITUDE LONGITUDE SUBROUTINES
#		PLANETARY INERTIAL ORIENTATION
#		MEASUREMENT INCORPORATION
#		CONIC SUBROUTINES
#		INTEGRATION INITIALIZATION
#		ORBITAL INTEGRATION
#		INFLIGHT ALIGNMENT ROUTINES
#		POWERED FLIGHT SUBROUTINES
#		TIME OF FREE FALL
#		AGC BLOCK TWO SELF-CHECK
#		PHASE TABLE MAINTENANCE
#		RESTARTS ROUTINE
#		IMU MODE SWITCHING ROUTINES
#		KEYRUPT, UPRUPT
#		DISPLAY INTERFACE ROUTINES
#		SERVICE ROUTINES
#		ALARM AND ABORT
#		UPDATE PROGRAM
#		RTB OP CODES
#	LMDAP
#		T6-RUPT PROGRAMS
#		DAP INTERFACE SUBROUTINES
#		DAPIDLER PROGRAM
#		P-AXIS RCS AUTOPILOT
#		Q,R-AXIS RCS AUTOPILOT
#		TJET LAW
```

```
#               KALMAN FILTER
#               TRIM GIMBAL CONTROL SYSTEM
#               AOSTASK AND AOSJOB
#               SPS BACK-UP RCS CONTROL
#
#       SYMBOL TABLE LISTING
#       UNREFERENCED SYMBOL LISTING
#       ERASABLE & EQUALS CROSS-REFERENCE TABLE
#       SUMMARY OF SYMBOL TABLE LISTINGS
#       MEMORY TYPE & AVAILABILITY DISPLAY
#       COUNT TABLE
#       PARAGRAPHS GENERATED FOR THIS ASSEMBLY
#       OCTAL LISTING
#       OCCUPIED LOCATIONS TABLE
#       SUBROS CALLED & PROGRAM STATUS
```

```
# VERB LIST FOR LUMINARY

# REGULAR VERBS

# 00     NOT IN USE
# 01     DISPLAY OCTAL COMP 1 IN R1
# 02     DISPLAY OCTAL COMP 2 IN R1
# 03     DISPLAY OCTAL COMP 3 IN R1
# 04     DISPLAY OCTAL COMP 1,2 IN R1,R2
# 05     DISPLAY OCTAL COMP 1,2,3 IN R1,R2,R3
# 06     DISPLAY DECIMAL IN R1 OR R1,R2 OR R1,R2,R3
# 07     DISPLAY DP DECIMAL IN R1,R2 (TEST ONLY)
# 08
# 09
# 10
# 11     MONITOR OCTAL COMP 1 IN R1
# 12     MONITOR OCTAL COMP 2 IN R1
# 13     MONITOR OCTAL COMP 3 IN R1
# 14     MONITOR OCTAL COMP 1,2 IN R1,R2
# 15     MONITOR OCTAL COMP 1,2,3 IN R1,R2,R3
# 16     MONITOR DECIMAL IN R1 OR R1,R2 OR R1,R2,R3
# 17     MONITOR DP DECIMAL IN R1,R2 (TEST ONLY)
# 18
# 19
# 20
# 21     LOAD COMPONENT 1 INTO R1
# 22     LOAD COMPONENT 2 INTO R2
# 23     LOAD COMPONENT 3 INTO R3
# 24     LOAD COMPONENT 1,2, INTO R1,R2
# 25     LOAD COMPONENT 1,2,3 INTO R1,R2,R3
# 26
# 27     DISPLAY FIXED MEMORY
# 28
# 29
# 30     REQUEST EXECUTIVE
# 31     REQUEST WAITLIST
# 32     RECYCLE PROGRAM
# 33     PROCEED WITHOUT DSKY INPUTS
# 34     TERMINATE FUNCTION
# 35     TEST LIGHTS
# 36     REQUEST FRESH START
# 37     CHANGE PROGRAM (MAJOR MODE)
# 38
# 39
```

```
# EXTENDED VERBS

# 40    ZERO CDU-S
# 41    COARSE ALIGN CDU-S
# 42    FINE ALIGN IMU
# 43    LOAD IMU ATT ERROR METERS
# 44    TERMINATE RR CONTINUOUS DESIGNATE (V41N72 OPTION 2)
# 45
# 46
# 47    INITIALIZE AGS (R47)
# 48    REQUEST DAP DATA LOAD ROUTINE (RO3)
# 49    REQUEST CREW DEFINED MANEUVER ROUTINE (R62)
# 50    PLEASE PERFORM
# 51
# 52    MARK X-RETICLE
# 53    MARK Y-RETICLE
# 54    MARK X OR Y-RETICLE
# 55    INCREMENT AGC TIME (DECIMAL)
# 56    TERMINATE TRACKING (P20 + P25)
# 57    PERMIT LANDING RADAR UPDATES
# 58    INHIBIT LANDING RADAR UPDATES
# 59
# 60    COMMAND LR TO POSITON 2.
# 61    DISPLAY DAP FOLLOWING ATTITUDE ERRORS.
# 62    DISPLAY TOTAL ATTITUDE ERRORS WITH RESPECT TO NOUN 22.
# 63    SAMPLE RADAR ONCE PER SECOND (R04).
# 64    REQUEST S-BAND ANTENNA ROUTINE (R05).
# 65    DISABLE U AND V JET FIRINGS DURING DPS BURNS.
# 66    VEHICLES ARE ATTACHED.  MOVE THIS VEHICLE STATE TO OTHER VEHICLE.
# 67    DISPLAY W MATRIX
# 68
# 69    CAUSE RESTART
# 70    UPDATE LIFTOFF TIME
# 71    UNIVERSAL UPDATE-BLOCK ADR
# 72    UNIVERSAL UPDATE-SINGLE ADR
# 73    UPDATE AGC TIME (OCTAL)
# 74    INITIALIZE ERASABLE DUMP VIA DOWNLINK
# 75    ENABLE U AND V JET FIRINGS DURING DPS BURNS.
# 76    MINIMUM IMPUSE COMMAND MODE.
# 77    RATE COMMAND AND ATTITUDE HOLD MODE
# 78    LR SPURIOUS RETURN TEST START (R77)
# 79    LR SPURIOUS RETURN TEST STOP
# 80    UPDATE LEM STATE VECTOR
# 81    UPDATE CSM STATE VECTOR
# 82    REQUEST ORBIT PARAM DISPLAY (R30)
# 83    REQUEST REND  PARAM DISPLAY (R31)
# 84
# 85    DISPLAY RR LOS AZ AND ELEV
# 86
# 87
```

```
# 88
# 89     REQUEST RENDEZVOUS FINAL ATTITUDE ROUTINE (R63)
# 90     REQUEST RENDEZVOUS OUT OF PLANE DISPLAY ROUTINE (R36)
# 91     DISPLAY BANK SUM
# 92     OPERATE IMU PERFORMANCE TEST (P07)
# 93     ENABLE W MATRIX INITIALIZATION
# 94
# 95     NO UPDATE OF EITHER STATE VECTOR (P20 OR P22)
# 96     INTERRUPT INTEGRATION AND GO TO P00
# 97     PERFORM ENGINE FAIL PROCEDURE
# 98
# 99     PLEASE ENABLE ENGINE
```

```
# IN THE FOLLOWING NOUN LIST THE :NO LOAD: RESTRICTION MEANS THE NOUN
# CONTAINS AT LEAST ONE COMPONENT WHICH CANNOT BE LOADED, I.E. OF
# SCALE TYPE L (MIN/SEC), PP (2 INTEGERS) OR TT (LANDING RADAR POSITION).
# IN THIS CASE VERBS 24 AND 25 ARE NOT ALLOWED, BUT VERBS 21, 22 OR 23
# MAY BE USED TO LOAD ANY OF THE NOUN:S COMPONENTS WHICH ARE NOT OF THE
# ABOVE SCALE TYPES.
#
# THE :DEC ONLY: RESTRICTION MEANS ONLY DECIMAL OPERATION IS ALLOWED ON
# EVERY COMPONENT IN THE NOUN. (NOTE THAT :NO LOAD: IMPLIES :DEC ONLY:.)

# NORMAL NOUNS                              COMPONENTS   SCALE AND DECIMAL POINT             RESTRICTIONS
# 00    NOT IN USE
# 01    SPECIFY MACHINE ADDRESS (FRACTIONAL)    3COMP    .XXXXX FOR EACH
# 02    SPECIFY MACHINE ADDRESS (WHOLE)         3COMP    XXXXX. FOR EACH
# 03    SPECIFY MACHINE ADDRESS (DEGREES)       3COMP    XXX.XX DEG FOR EACH
# 04    ANGULAR ERROR/DIFFERENCE                1COMP    XXX.XX DEG
# 05    ANGULAR ERROR/DIFFERENCE                1COMP    XXX.XX DEG
# 06    OPTION CODE                             3COMP    OCTAL ONLY FOR EACH
# LOADING NOUN 07 WILL SET OR RESET SELECTED BITS IN ANY ERASABLE REGISTER
# 07    ECADR OF WORD TO BE MODIFIED            3COMP    OCTAL ONLY FOR EACH
#       ONES FOR BITS TO BE MODIFIED
#       1 TO SET OR 0 TO RESET SELECTED BITS
# 08    ALARM DATA                              3COMP    OCTAL ONLY FOR EACH
# 09    ALARM CODES                             3COMP    OCTAL ONLY FOR EACH
# 10    CHANNEL TO BE SPECIFIED                 1COMP    OCTAL ONLY
# 11    TIG OF CSI                              3COMP    00XXX. HRS                          DEC ONLY
#                                                        000XX. MIN                          MUST LOAD 3 COMPS
#                                                        0XX.XX SEC
# 12    OPTION CODE                             2COMP    OCTAL ONLY FOR EACH
#       (USED BY EXTENDED VERBS ONLY)
# 13    TIG OF CDH                              3COMP    00XXX. HRS                          DEC ONLY
#                                                        000XX. MIN                          MUST LOAD 3 COMPS
#                                                        0XX.XX SEC
# 14    CHECKLIST                               3COMP    XXXXX. FOR EACH
#       (USED BY EXTENDED VERBS ONLY)
#       (NOUN 25 IS PASTED AFTER DISPLAY)
# 15    INCREMENT MACHINE ADDRESS               1COMP    OCTAL ONLY
# 16    TIME OF EVENT                           3COMP    00XXX. HRS                          DEC ONLY
#       (USED BY EXTENDED VERBS ONLY)                    000XX. MIN                          MUST LOAD 3 COMPS
#                                                        0XX.XX SEC
# 17    SPARE
# 18    AUTO MANEUVER BALL ANGLES               3COMP    XXX.XX DEG FOR EACH
# 19    SPARE
# 20    ICDU ANGLES                             3COMP    XXX.XX DEG FOR EACH
# 21    PIPAS                                   3COMP    XXXXX. PULSES FOR EACH
# 22    NEW ICDU ANGLES                         3COMP    XXX.XX DEG FOR EACH
# 23    SPARE
# 24    DELTA TIME FOR AGC CLOCK                3COMP    00XXX. HRS                          DEC ONLY
#                                                        000XX. MIN                          MUST LOAD 3 COMPS
#                                                        0XX.XX SEC
```

```
# 25     CHECKLIST                       3COMP   XXXXX. FOR EACH
#        (USED WITH PLEASE PERFORM ONLY)
# 26     PRIORITY/DELAY, ADRES, BBCON    3COMP   OCTAL ONLY FOR EACH
# 27     SELF TEST ON/OFF SWITCH         1COMP   XXXXX.
# 28     SPARE
# 29     SPARE
# 30     SPARE
# 31     SPARE
# 32     TIME FROM PERIGEE               3COMP   00XXX. HRS                       DEC ONLY
#                                                000XX. MIN                       MUST LOAD 3 COMPS
#                                                0XX.XX SEC
# 33     TIME OF IGNITION                3COMP   00XXX. HRS                       DEC ONLY
#                                                000XX. MIN                       MUST LOAD 3 COMPS
#                                                0XX.XX SEC
# 34     TIME OF EVENT                   3COMP   00XXX. HRS                       DEC ONLY
#                                                000XX. MIN                       MUST LOAD 3 COMPS
#                                                0XX.XX SEC
# 35     TIME FROM EVENT                 3COMP   00XXX. HRS                       DEC ONLY
#                                                000XX. MIN                       MUST LOAD 3 COMPS
#                                                0XX.XX SEC
# 36     TIME OF AGC CLOCK               3COMP   00XXX. HRS                       DEC ONLY
#                                                000XX. MIN                       MUST LOAD 3 COMPS
#                                                0XX.XX SEC
# 37     TIG OF TPI                      3COMP   00XXX. HRS                       DEC ONLY
#                                                000XX. MIN                       MUST LOAD 3 COMPS
#                                                0XX.XX SEC
# 38     TIME OF STATE BEING INTEGRATED  3COMP   00XXX. HRS                       DEC ONLY
#                                                000XX. MIN                       MUST LOAD 3 COMPS
#                                                0XX.XX SEC
# 39     SPARE
```

```
# MIXED NOUNS                                  COMPONENTS   SCALE AND DECIMAL POINT                  RESTRICTIONS
# 40      TIME FROM IGNITION/CUTOFF                 3COMP   XXBXX  MIN/SEC                            NO LOAD, DEC ONLY
#         VG,                                               XXXX.X FT/SEC
#         DELTA V (ACCUMULATED)                             XXXX.X FT.SEC
# 41      TARGET  AZIMUTH                           2COMP   XXX.XX DEG                                (FOR SYSTEM TEST)
#                 ELEVATION                                 XX.XXX DEG
# 42      APOGEE,                                   3COMP   XXXX.X NAUT MI                            DEC ONLY
#         PERIGEE,                                          XXXX.X NAUT MI
#         DELTA V (REQUIRED)                                XXXX.X FT/SEC
# 43      LATITUDE,                                 3COMP   XXX.XX DEG                                DEC ONLY
#         LONGITUDE,                                        XXX.XX DEG
#         ALTITUDE                                          XXXX.X NAUT MI
# 44      APOGEE,                                   3COMP   XXXX.X NAUT MI                            NO LOAD, DEC ONLY
#         PERIGEE,                                          XXXX.X NAUT MI
#         TFF                                               XXBXX  MIN/SEC
# 45      MARKS,                                    3COMP   XXXXX.                                    NO LOAD, DEC ONLY
#         TFI OF NEXT BURN,                                 XXBXX  MIN/SEC
#         MGA                                               XXX.XX DEG
# 46      AUTOPILOT CONFIGURATION                   1COMP   OCTAL ONLY
# 47      LEM WEIGHT,                               2COMP   XXXXX. LBS                                DEC ONLY
#         CSM WEIGHT                                        XXXXX. LBS
# 48      GIMBAL PITCH TRIM,                        2COMP   XXX.XX DEG                                DEC ONLY
#         GIMBAL ROLL TRIM                                  XXX.XX DEG
# 49      DELTA R,                                  3COMP   XXXX.X NAUT MI                            DEC ONLY
#         DELTA V,                                          XXXX.X FT/SEC
#         RADAR DATA SOURCE CODE                            XXXXX.
# 50      SPARE
# 51      S-BAND ANTENNA ANGLES   PITCH             2COMP   XXX.XX DEG                                DEC ONLY
#                                 YAW                       XXX.XX DEG
# 52      CENTRAL ANGLE OF ACTIVE VEHICLE           1COMP   XXX.XX DEG
# 53      SPARE
# 54      RANGE,                                    3COMP   XXX.XX NAUT MI                            DEC ONLY
#         RANGE RATE,                                       XXXX.X FT/SEC
#         THETA                                             XXX.XX DEG
# 55      NO. OF APSIDAL CROSSINGS                  3COMP   XXXXX.                                    DEC ONLY
#         ELEVATION ANGLE                                   XXX.XX DEG
#         CENTRAL ANGLE OF PASSIVE VEHICLE                  XXX.XX DEG
# 56      RR LOS  AZIMUTH                           2COMP   XXX.XX DEG
#                 ELEVATION                                 XXX.XX DEG
# 57      DELTA R                                   1COMP   XXXX.X NAUT MI                            DEC ONLY
# 58      PERIGEE ALT (POST TPI)                    3COMP   XXXX.X NAUT MI                            DEC ONLY
#         DELTA V TPI                                       XXXX.X FT/SEC
#         DELTA V TPF                                       XXXX.X FT/SEC
# 59      DELTA VELOCITY LOS                        3COMP   XXXX.X FT/SEC FOR EA.                     DEC ONLY
# 60      HORIZONTAL VELOCITY                       3COMP   XXXX.X FT/SEC                             DEC ONLY
#         ALTITUDE RATE                                     XXXX.X FT/SEC
#         COMPUTED ALTITUDE                                 XXXXX. FEET
# 61      TIME TO GO IN BRAKING PHASE               3COMP   XXBXX  MIN/SEC                            NO LOAD, DEC ONLY
#         TIME FROM IGNITION                                XXBXX  MIN/SEC
```

```
#       CROSS RANGE DISTANCE                               XXXX.X NAUT MI
# 62    ABSOLUTE VALUE OF VELOCITY              3COMP      XXXX.X FT/SEC                         NO LOAD, DEC ONLY
#       TIME FROM IGNITION                                 XXBXX  MIN/SEC
#       DELTA V (ACCUMULATED)                              XXXX.X FT/SEC
# 63    ABSOLUTE VALUE OF VELOCITY              3COMP      XXXX.X FT/SEC                         DEC ONLY
#       ALTITUDE RATE                                      XXXX.X FT/SEC
#       COMPUTED ALTITUDE                                  XXXXX. FEET
# 64    TIME LEFT FOR REDESIGNATION- LPD ANGLE  3COMP      XXBXX                                 NO LOAD, DEC ONLY
#       ALTITUDE RATE                                      XXXX.X FT/SEC
#       COMPUTED ALTITUDE                                  XXXXX. FEET
# 65    SAMPLED AGC TIME                        3COMP      00XXX. HRS.                           DEC ONLY
#       (FETCHED IN INTERRUPT)                             000XX. MIN                            MUST LOAD 3 COMPS
#                                                          0XX.XX SEC
# 66    LR      RANGE                           2COMP      XXXXX. FEET                           NO LOAD, DEC ONLY
#               POSITION                                   +0000X
# 67    LRVX                                    3COMP      XXXXX. FT/SEC
#       LRVY                                               XXXXX. FT/SEC
#       LRVZ                                               XXXXX. FT/SEC
# 68    SLANT RANGE TO LANDING SITE             3COMP      XXXX.X NAUT MI                        NO LOAD, DEC ONLY
#       TIME TO GO IN BRAKING PHASE                        XXBXX  MIN/SEC
#       LR ALTITUDE - COMPUTED ALTITUDE         XXXXX. FEET
# 69    SPARE
# 70    AOT DETENT CODE/STAR CODE               3COMP      OCTAL ONLY FOR EACH
# 71    AOT DETENT CODE/STAR CODE               3COMP      OCTAL ONLY FOR EACH
# 72    RR      360 - TRUNNION ANGLE            2COMP      XXX.XX DEG
#               SHAFT ANGLE                                XXX.XX DEG
# 73    NEW RR  360 - TRUNNION ANGLE            2COMP      XXX.XX DEG
#               SHAFT ANGLE                                XXX.XX DEG
# 74    TIME FROM IGNITION                      3COMP      XXBXX  MIN/SEC                        NO LOAD, DEC ONLY
#       YAW AFTER VEHICLE RISE                             XXX.XX DEG
#       PITCH AFTER VEHICLE RISE                           XXX.XX DEG
# 75    DELTA ALTITUDE CDH                      3COMP      XXXX.X NAUT MI                        NO LOAD, DEC ONLY
#       DELTA TIME (CDH-CSI OR TPI-CDH)                    XXBXX  MIN/SEC
#       DELTA TIME (TPI-CDH OR TPI-NOMTPI)                 XXBXX  MIN/SEC
# 76    DESIRED HORIZONTAL VELOCITY             3COMP      XXXX.X FT/SEC                         DEC ONLY
#       DESIRED RADIAL VELOCITY                            XXXX.X FT/SEC
#       CROSS-RANGE DISTANCE                               XXXX.X NAUT MI
# 77    TIME TO ENGINE CUTOFF                   2COMP      XXBXX  MIN/SEC                        NO LOAD, DEC ONLY
#       VELOCITY NORMAL TO CSM PLANE                       XXXX.X FT/SEC
# 78    RR      RANGE                           2COMP      XXX.XX NAUT MI                        NO LOAD, DEC ONLY
#               RANGE RATE                                 XXXXX. FT/SEC
# 79    CURSOR ANGLE                            3COMP      XXX.XX DEG                            DEC ONLY
#       SPIRAL ANGLE                                       XXX.XX DEG
#       POSITION CODE                                      XXXXX.
# 80    DATA INDICATOR,                         2COMP      XXXXX.
#       OMEGA                                              XXX.XX DEG
# 81    DELTA V (LV)                            3COMP      XXXX.X FT/SEC FOR EACH                DEC ONLY
```

```
# 82    DELTA V (LV)                          3COMP   XXXX.X FT/SEC FOR EACH                  DEC ONLY
# 83    DELTA V (BODY)                        3COMP   XXXX.X FT/SEC FOR EACH                  DEC ONLY
# 84    DELTA V (OTHER VEHICLE)               3COMP   XXXX.X FT/SEC FOR EACH                  DEC ONLY
# 85    VG (BODY)                             3COMP   XXXX.X FT/SEC FOR EACH                  DEC ONLY
# 86    VG (LV)                               3COMP   XXXX.X FT/SEC FOR EACH                  DEC ONLY
# 87    BACKUP OPTICS LOS       AZIMUTH       2COMP   XXX.XX DEG
#                               ELEVATION             XXX.XX DEG
# 88    HALF UNIT SUN OR PLANET VECTOR        3COMP   .XXXXX FOR EACH                         DEC ONLY
# 89    LANDMARK        LATITUDE              3COMP   XX.XXX DEG                              DEC ONLY
#                       LONGITUDE/2                   XX.XXX DEC
#                       ALTITUDE                      XXX.XX NAUT MI
# 90    Y                                     3COMP   XXX.XX NM                               DEC ONLY
#       Y DOT                                         XXXX.X FPS
#       PSI                                           XXX.XX DEG
# 91    ALTITUDE                              3COMP   XXXXXB. NAUT MI
#       VELOCITY                                      XXXXX.  FT/SEC
#       FLIGHT PATH ANGLE                             XXX.XX  DEG
# 92    SPARE
# 93    DELTA GYRO ANGLES                     3COMP   XX.XXX DEG FOR EACH
# 94    SPARE
# 95    SPARE
# 96    SPARE
# 97    SYSTEM TEST INPUTS                    3COMP   XXXXX. FOR EACH
# 98    SYSTEM TEST RESULTS AND INPUTS        3COMP   XXXXX.
#                                                     .XXXXX
#                                                     XXXXX.
# 99    RMS IN POSITION                       3COMP   XXXXX. FT                               DEC ONLY
#       RMS IN VELOCITY                               XXXX.X FT/SEC
#       RMS IN BIAS                                   XX.XXX RADIANS
```

```
# REGISTERS AND SCALING FOR NORMAL NOUNS

# NOUN            REGISTER                SCALE TYPE
# 00              NOT IN USE
# 01              SPECIFY ADDRESS         B
# 02              SPECIFY ADDRESS         C
# 03              SPECIFY ADDRESS         D
# 04                      DSPTEM1         H
# 05                      DSPTEM1         H
# 06                      OPTION1         A
# 07                      XREG            A
# 08                      ALMCADR         A
# 09                      FAILREG         A
# 10              SPECIFY CHANNEL         A
# 11                      TCSI            K
# 12                      OPTIONX         A
# 13                      TCDH            K
# 14                      DSPTEMX         C
# 15              INCREMENT ADDRESS       A
# 16                      DSPTEMX         K
# 17              SPARE
# 18                      FDAIX           D
# 19              SPARE
# 20                      CDUX            D
# 21                      PIPAX           C
# 22                      THETAD          D
# 23              SPARE
# 24                      DSPTEM2 +1      K
# 25                      DSPTEM1         C
# 26                      DSPTEM1         A
# 27                      SMODE           C
# 28              SPARE
# 29              SPARE
# 30              SPARE
# 31              SPARE
# 32                      -TPER           K
# 33                      TIG             K
# 34                      DSPTEM1         K
# 35                      TTOGO           K
# 36                      TIME2           K
# 37                      TTPI            K
# 38                      TET             K
# 39              SPARE
```

```
# REGISTERS AND SCALING FOR MIXED NOUNS

# NOUN          COMP            REGISTER                SCALE TYPE
# 40            1               TTOGO                   L
#               2               VGDISP                  S
#               3               DVTOTAL                 S
# 41            1               DSPTEM1                 D
#               2               DSPTEM1 +1              E
# 42            1               HAPO                    Q
#               2               HPER                    Q
#               3               VGDISP                  S
# 43            1               LAT                     H
#               2               LONG                    H
#               3               ALT                     Q
# 44            1               HAPOX                   Q
#               2               HPERX                   Q
#               3               TFF                     L
# 45            1               TRKMKCNT                C
#               2               TTOGO                   L
#               3               +MGA                    H
# 46            1               DAPDATR1                A
# 47            1               LEMMASS                 KK
#               2               CSMMASS                 KK
# 48            1               PITTIME                 NN
#               2               ROLLTIME                NN
# 49            1               R22DISP                 Q
#               2               R22DISP +2              S
#               3               WHCHREAD                C
# 50            SPARE
# 51            1               ALPHASB                 H
#               2               BETASB                  H
# 52            1               ACTCENT                 H
# 53            SPARE
# 54            1               RANGE                   JJ
#               2               RRATE                   S
#               3               RTHETA                  H
# 55            1               NN                      C
#               2               ELEV                    H
#               3               CENTANG                 H
# 56            1               RR-AZ                   H
#               2               RR-ELEV                 H
# 57            1               DELTAR                  Q
# 58            1               POSTTPI                 Q
#               2               DELVTPI                 S
#               3               DELVTPF                 S
# 59            1               DVLOS                   S
#               2               DVLOS +2                S
#               3               DVLOS +4                S
# 60            1               VHORIZ                  S
```

```
#                2                HDOTDISP            S
#                3                HCALC               RR
# 61             1                TTFDISP             L
#                2                TTOGO               L
#                3                OUTOFPLN            QQ
# 62             1                ABVEL               S
#                2                TTOGO               L
#                3                DVTOTAL             S
# 63             1                ABVEL               S
#                2                HDOTDISP            S
#                3                HCALC1              RR
# 64             1                FUNNYDSP            PP
#                2                HDOTDISP            S
#                3                HCALC               RR
# 65             1                SAMPTIME            K
#                2                SAMPTIME            K
#                3                SAMPTIME            K
# 66             1                RSTACK +6           W
#                2                CHANNEL 33          TT
# 67             1                RSTACK              X
#                2                RSTACK +2           Y
#                3                RSTACK +4           Z
# 68             1                RANGEDSP            QQ
#                2                TTFDISP             L
#                3                DELTAH              RR
# 69             SPARE
# 70             1                AOTCODE             A
#                2                AOTCODE +1          A
#                3                AOTCODE +2          A
# 71             1                AOTCODE             A
#                2                AOTCODE +1          A
#                3                AOTCODE +2          A
# 72             1                CDUT                WW
#                2                CDUS                D
# 73             1                TANG                WW
#                2                TANG +1             D
# 74             1                TTOGO               L
#                2                YAW                 H
#                3                PITCH               H
# 75             1                DIFFALT             Q
#                2                T1TOT2              L
#                3                T2TOT3              L
# 76             1                ZDOTD               S
#                2                RDOTD               S
#                3                XRANGE              Q
# 77             1                TTOGO               L
#                2                YDOT                S
# 78             1                RSTACK              U
#                2                RSTACK +2           V
# 79             1                CURSOR              D
```

```
#               2               SPIRAL                  D
#               3               POSCODE                 C
# 80            1               DATAGOOD                C
#               2               OMEGAD                  H
# 81            1               DELVLVC                 S
#               2               DELVLVC +2              S
#               3               DELVLVC +4              S
# 82            1               DELVLVC                 S
#               2               DELVLVC +2              S
#               3               DELVLVC +4              S
# 83            1               DELVIMU                 S
#               2               DELVIMU +2              S
#               3               DELVIMU +4              S
# 84            1               DELVOV                  S
#               2               DELVOV +2               S
#               3               DELVOV +4               S
# 85            1               VGBODY                  S
#               2               VGBODY +2               S
#               3               VGBODY +4               S
# 86            1               DELVLVC                 S
#               2               DELVLVC +2              S
#               3               DELVLVC +4              S
# 87            1               AZ                      D
#               2               EL                      D
# 88            1               STARAD                  B
#               2               STARAD +2               B
#               3               STARAD +4               B
# 89            1               LANDLAT                 G
#               2               LANDLONG                G
#               3               LANDALT                 JJ
# 90            1               RANGE                   JJ
#               2               RRATE                   S
#               3               RTHETA                  H
# 91            1               P21ALT                  Q (MEMORY/100 TO DISPLAY TENS N.M.)
#               2               P21VEL                  P
#               3               P21GAM                  H
# 92            SPARE
# 93            1               OGC                     G
#               2               OGC +2                  G
#               3               OGC +4                  G
# 94            SPARE
# 95            SPARE
# 96            SPARE
# 97            1               DSPTEM1                 C
#               2               DSPTEM1 +1              C
#               3               DSPTEM1 +2              C
# 98            1               DSPTEM2                 C
#               2               DSPTEM2 +1              B
#               3               DSPTEM2 +2              C
# 99            1               WWPOS                   XX
```

```
#              2              WWVEL              YY
#              3              WWBIAS             AAA
```

```
# NOUN SCALES AND FORMATS

# -SCALE TYPE-                                 PRECISION
# UNITS                   DECIMAL FORMAT                 --      AGC FORMAT
# ------------            --------------                 --      ----------

# -A-
# OCTAL                   XXXXX                          SP      OCTAL

# -B-                                                                     -14
# FRACTIONAL              .XXXXX                         SP      BIT 1 = 2    UNITS
#                         (MAX .99996)

# -C-
# WHOLE                   XXXXX.                         SP      BIT 1 = 1 UNIT
#                         (MAX 16383.)

# -D-                                                                         15
# CDU DEGREES             XXX.XX DEGREES                 SP      BIT 1 = 360/2   DEGREES
#                         (MAX 359.99)                           (USES 15 BITS FOR MAGNI-
#                                                                 TUDE AND 2-S COMP.)

# -E-                                                                        14
# ELEVATION DEGREES       XX.XXX DEGREES                 SP      BIT 1 = 90/2   DEGREES
#                         (MAX 89.999)

# -F-                                                                         14
# DEGREES (180)           XXX.XX DEGREES                 SP      BIT 1 = 180/2   DEGREES
#                         (MAX 179.99)

# -G-
# DP DEGREES(90)          XX.XXX DEGREES                 DP      BIT 1 OF LOW REGISTER =
#                                                                     28
#                                                                360/2   DEGREES

# -H-
# DP DEGREES (360)        XXX.XX DEGREES                 DP      BIT 1 OF LOW REGISTER =
#                                                                     28
#                         (MAX 359.99)                           360/2   DEGREES
```

```
# -K-
# TIME (HR, MIN, SEC)    00XXX. HR               DP       BIT 1 OF LOW REGISTER =
#                        000XX. MIN                         -2
#                        0XX.XX SEC                       10    SEC
#                        (DECIMAL ONLY.
#                        MAX MIN COMP=59
#                        MAX SEC COMP=59.99
#                        MAX CAPACITY=745 HRS
#                                      39 MINS
#                                      14.55 SECS.
#                        WHEN LOADING, ALL 3
#                        COMPONENTS MUST BE
#                        SUPPLIED.)

# -L-
# TIME (MIN/SEC)         XXBXX MIN/SEC           DP       BIT 1 OF LOW REGISTER =
#                        (B IS A BLANK                      -2
#                        POSITION, DECIMAL                10    SEC
#                        ONLY, DISPLAY OR
#                        MONITOR ONLY.  CANNOT
#                        BE LOADED.
#                        MAX MIN COMP=59
#                        MAX SEC COMP=59
#                        VALUES GREATER THAN
#                        59 MIN 59 SEC
#                        ARE DISPLAYED AS
#                        59 MIN 59 SEC.)

# -M-                                                                -2
# TIME (SEC)             XXX.XX SEC              SP       BIT 1 = 10    SEC
#                        (MAX 163.83)

# -N-
# TIME(SEC) DP           XXX.XX SEC              DP       BIT 1 OF LOW REGISTER =
#                                                           -2
#                                                         10    SEC.

# -P-
# VELOCITY 2             XXXXX. FEET/SEC         DP       BIT 1 OF HIGH REGISTER =
#                        (MAX 41994.)                      -7
#                                                         2    METERS/CENTI-SEC

# -Q-
# POSITION 4             XXXX.X NAUTICAL MILES   DP       BIT 1 OF LOW REGISTER =
#                                                         2 METERS

# -S-
# VELOCITY 3             XXXX.X FT/SEC           DP       BIT 1 OF HIGH REGISTER =
#                                                          -7
#                                                         2    METERS/CENTI-SEC
```

```
# -T-                                                                 -2
# G                        XXX.XX G                 SP      BIT 1 = 10    G
#                          (MAX 163.83)

# -U-
# RENDEZVOUS               XXX.XX NAUT MI           DP      LOW ORDER BIT OF LOW ORDER
# RADAR RANGE                                               WORD = 9.38 FEET

# -V-
# RENDEZVOUS               XXXXX. FEET/SEC          DP      LOW ORDER BIT OF LOW ORDER
# RADAR RANGE RATE                                          WORD = -.6278 FEET/SEC

# -W-
# LANDING RADAR            XXXXX. FEET              DP      LOW ORDER BIT OF LOW ORDER
# ALTITUDE                                                  WORD = 1.079 FEET

# -X-
# LANDING RADAR            XXXXX. FEET/SEC          DP      LOW ORDER BIT OF LOW ORDER
# VELX                                                      WORD = -.6440 FEET/SEC

# -Y-
# LANDING RADAR            XXXXX. FEET/SEC          DP      LOW ORDER BIT OF LOW ORDER
# VELY                                                      WORD = 1.212 FEET/SEC

# -Z-
# LANDING RADAR            XXXXX. FEET/SEC          DP      LOW ORDER BIT OF LOW ORDER
# VELZ                                                      WORD = .8668 FEET/SEC

# -AA-
# INITIAL/FINAL            XXXXX. FEET              DP      LOW ORDER BIT OF LOW ORDER
# ALTITUDE                                                  WORD = 2.345 FEET

# -BB-
# ALTITUDE RATE            XXXXX. FEET/SEC          SP      LOW ORDER BIT = .5
#                          (MAX 08191.)                     FEET/SEC

# -CC-
# FORWARD/LATERAL          XXXXX. FEET/SEC          SP      LOW ORDER BIT = .5571
# VELOCITY                 (MAX 09126.)                     FEET/SEC

# -DD-
# ROTATIONAL HAND          XXXXX. DEG/SEC           SP      FRACTIONAL PART OF PI RAD
# CONTROLLER ANGULAR       (MAX 00044.)                                        4  SEC
# RATES

# -EE-
# OPTICAL TRACKER          XXX.XX DEG.              DP      LOW ORDER BIT OF LOW ORDER
# AZIMUTH ANGLE                                                        15
#                                                           WORD = 360/2    DEGREES
```

```
# -JJ-
# POSITION5                XXX.XX NAUT MI          DP      BIT 1 OF LOW REGISTER =
#                                                          2 METERS

# -KK-                                                                         16
# WEIGHT2                  XXXXX. LBS              SP      FRACTIONAL PART OF 2   KG

# -NN-
# TRIM DEGREES 2           XXX.XX DEG              SP      BIT 1=.01 SEC(TIME)
#                          (MAX 032.76)

# -PP-
# 2 INTEGERS               +XXBYY                  DP      BIT 1 OF HIGH REGISTER =
#                          (B IS A BLANK                    1 UNIT OF XX
#                          POSITION. DECIMAL               BIT 1 OF LOW REGISTER =
#                          ONLY, DISPLAY OR                 1 UNIT OF YY
#                          MONITOR ONLY. CANNOT            (EACH REGISTER MUST
#                          BE LOADED.)                     CONTAIN A POSITIVE INTEGER
#                          (MAX 99B99)                     LESS THAN 100)

# -QQ-
# POSITION7                XXXX.X NAUT MI          DP      BIT 1 OF LOW REGISTER =
#                          (MAX 9058.9)                     -4
#                                                          2   METERS

# -RR-
# COMPUTED ALTITUDE        XXXXX. FEET             DP      BIT 1 OF LOW REGISTER =
#                                                           -4
#                                                          2   METERS

# -SS-
# DP DEGREES               XXXX.X DEGREES          DP      BIT 1 OF HIGH REGISTER =
#                                                          1 DEGREE

# -TT-
# LANDING RADAR            +0000X                  CHANNEL 33, BIT 6 = NOT POSIT. 1
# POSITION                 (DECIMAL ONLY.          CHANNEL 33, BIT 7 = NOT POSIT. 2
#                          DISPLAY OR MONITOR      X = 1 FOR LR POSITION 1
#                          ONLY. CANNOT BE         X = 2 FOR LR POSITION 2
#                          LOADED.)

# -WW-                                                                         15
# 360-CDU DEGREES          XXX.XX DEGREES          SP      BIT 1 = 360 - (360/2   )
#                          (MAX 359.99)                    DEGREES
#                                                          (USES 15 BITS FOR MAGNI-
#                                                           TUDE AND 2-S COMP.)

# -XX-
# POSITION 9               XXXXX. FEET             DP      BIT 1 OF LOW REGISTER =
#                                                           -9
#                                                          2   METERS
```

```
# -YY-
# VELOCITY 4              XXXX.X FEET/SEC         DP      FRACTIONAL PART
#                         (MAX 328.0)                     METERS/CENTI-SEC

# -AAA-
# RADIANS                 XX.XXX RADIANS          DP      BIT 1 OF HIGH REGISTER =
#                         (MAX 31.999)                     -9
#                                                         2   RADIANS.

# THAT-S ALL ON THE NOUNS.
```

```
# ALARM CODES FOR LUMINARY

# *9              *18                                              *60      COLUMN

# CODE          * TYPE                                             SET BY

# 00105        ** AOTMARK SYSTEM IN USE
# 00107           MORE THAN 5 MARK PAIRS                           AOTMARK
# 00111           MARK MISSING                                     AOTMARK
# 00112           MARK OR MARK REJECT NOT BEING ACCEPTED           AOTMARK
# 00113           NO INBITS                                        AOTMARK
# 00114           MARK MADE BUT NOT DESIRED                        AOTMARK
# 00115           NO MARKS IN LAST PAIR TO REJECT                  AOTMARK
# 00206           ZERO ENCODE NOT ALLOWED WITH COARSE ALIGN        IMU MODE SWITCHING
# 00206            + GIMBAL LOC
# 00207           ISS TURNON REQUEST NOT PRESENT FOR 90 SEC        T4RUPT
# 00210           IMU NOT OPERATING                                IMU MODE SWITCH, IMU-2, RD2, P51, P57
# 00211           COARSE ALIGN ERROR                               IMU MODE SWITCH
# 00212           PIPA FAIL BUT PIPA IS NOT BEING USED             IMU MODE SWITCH, T4RPT
# 00213           IMU NOT OPERATING WITH TURN-ON REQUEST           T4RUPT
# 00214           PROGRAM USING IMU WHEN TURNED OFF                T4RUPT
# 00217           BAD RETURN FROM IMUSTALL                         P51, P52, P57
# 00220           IMU NOT ALIGNED - NO REFSMMAT                    R02, R47
# 00401           DESIRED GIMBAL ANGLE YIELDS GIMBAL LOCK          INF ALIGN, IMU-2,
#                                                                  FINDCDUW
# 00402           FINDCDUW NOT CONTROLLING ATTITUDE                FINDCDUW
# 00404           TWO STARS NOT AVAILABLE IN ANY DETENT            R59, LUNAR SURFACE
# 00405           TWO STARS NOT AVAILABLE                          P52
# 00421           W-MATRIX OVERFLOW                                INTEGRV
# 00430        ** ACCELERATION OVERFLOW IN INTEGRATION             ORBITAL INTEGRATION
# 00501         P RADAR ANTENNA OUT OF LIMITS                      R23
# 00502           BAD RADAR GIMBAL ANGLE INPUT                     V41N72
# 00503         P RADAR ANTENNA DESIGNATE FAIL                     R21, NON-P IN V41N72
# 00510           RADAR AUTO DESCRETE NOT PRESENT                  R25
# 00511           LR NOT IN POSITION 2 OR REPOSITIONING            SERVICER
# 00514         P RR GOES OUT OF AUTO MODE WHILE IN USE            P20
# 00515           RR CDU FAIL DISCRETE PRESENT                     R25
# 00520           RADAR RUPT NOT EXPECTED AT THIS TIME             RADAR READ
# 00521           COULD NOT READ RADAR                             P20
# 00522           LANDING RADAR POSITION CHANGE                    RADAR READ
# 00523         P LR ANTENNA DIDN'T ACHIEVE POSITION 2             SERVICER, V60 (NON-P IN V60)
# 00525         P DELTA THETA GREATER THAN 3 DEGREES               R22
# 00526         P RANGE GREATER THAN 400 NAUT. MILES               P20, P22
# 00527         P LOS NOT IN MODE II COVERAGE WHILE ON             R21, R24
#                 LUNAR SURFACE
#                 OR VEHICLE MANEUVER REQUIRED                     R24 (20)
# 00530         P LOS NOT IN MODE2 COVERAGE                        R21
#                 ON LUNAR SURFACE AFTER 600 SECS.
# 00600           IMAGINARY ROOTS ON FIRST ITERATION               P32, P72
# 00601           PERIGEE ALTITUDE CSI LT PMIN1                    P32, P72.
```

```
# 00602          PERIGEE ALTITUDE CDH LT PMIN2                       P32, P72.
# 00603          CSI TO CDH TIME LT TMIN12                           P32, P72, P33, P73
# 00604          CDH TO TPI TIME LT TMIN23                           P32, P72,
#                 OR COMPUTED CDH TIME GREATER THAN INPUT TPI TIME
# 00605          NUMBER OF ITERATIONS EXCEEDS LOOP MAXIMUM           P32, P72
# 00606          DV EXCEEDS MAXIMUM                                  P32, P72
# 00607       ** NO SOLN FROM TIME-THETA OR TIME-RADIUS              TIMETHET, TIMERAD
# 00611          NO TIG FOR GIVEN ELEV ANGLE                         P34, P74
# 00701          ILLEGAL OPTION CODE SELECTED                        P57
# 00777          PIPA FAIL CAUSED THE ISS WARNING                    T4RUPT
# 01102          AGC SELF TEST ERROR                                 SELF CHECK
# 01103       ** UNUSED CCS BRANCH EXECUTED                          ABORT
# 01104        * DELAY ROUTINE BUSY                                  EXEC
# 01105          DOWNLINK TOO FAST                                   T4RUPT
# 01106          UPLINK TOO FAST                                     T4RUPT
# 01107          PHASE TABLE FAILURE.  ASSUME                        RESTART
#                ERASABLE MEMORY IS SUSPECT.                         RESTART
# 01201        * EXECUTIVE OVERFLOW - NO VAC AREAS                   EXEC
# 01202        * EXECUTIVE OVERFLOW - NO CORE SETS                   EXEC
# 01203        * WAITLIST OVERFLOW - TOO MANY TASKS                  WAITLIST
# 01204       ** WAITLIST, VARDELAY, FIXDELAY, OR LONGCALL           WAITLIST ROUTINES
#                  CALLED WITH ZERO OR NEGATIVE DELTA-TIME
# 01206       ** SECOND JOB ATTEMPTS TO GO TO SLEEP                  PINBALL
# 01206           VIA KYBD AND DISPLAY PROGRAM
# 01207        * NO VAC AREAS FOR MARKS                              AOTMARK
# 01210        * TWO PROGRAMS USING DEVICE AT SAME TIME              MODE SWITCHING
# 01211        * ILLEGAL INTERRUPT OF EXTENDED VERB                  AOTMARK
# 01301          ARCSIN-ARCCOS ARGUMENT TOO LARGE                    INTERPRETER
# 01302       ** SQRT CALLED WITH NEGATIVE ARGUMENT                  INTERPRETER
# 01406          BAD RETURN FROM ROOTPSRS                            DESCENT GUIDANCE EQS.
# 01406       ** BAD RETURN FROM ROOTPSRS                            IGNITION ALGORITHM
#
#       NOTE:   1406 IS A POODOO DURING THE IGNITION ALGORITHM
#       AND AN ALARM DURING THE ACTUAL GUIDANCE PHASE.
#
# 01407          VG INCREASING (DELTA-V ACCUMULATED                  S40.8
#                .GT. 90 DEGREES AWAY FROM DESIRED THRUST            S40.8
#                VECTOR.)                                            S40.8
# 01410          UNINTENTIONAL OVERFLOW IN GUIDANCE                  DESCENT GUIDANCE EQS.
# 01412          DESCENT IGNALG NOT CONVERGING                       P63
# 01501       ** KEYBOARD AND DISPLAY ALARM DURING                   PINBALL
# 01501           INTERNAL USE (NVSUB). ABORT
# 01502       ** ILLEGAL FLASHING DISPLAY                            GOPLAY
# 01520          V37 REQUEST NOT PERMITTED AT THIS TIME              V37
# 01600          OVERFLOW IN DRIFT TEST                              IMU 4
# 01601          BAD IMU TORQUE                                      OPT PRE ALIGN CALIB
# 01601                                                              IMU 4 (LEM)
# 01703          IGNITION TIME SLIPPED                               MIDTOAVE
# 01706          INCORRECT PROGRAM REQUESTED FOR VEHICLE
#                CONFIGURATION                                       P40, P42
```

```
# 02000       * DAP STILL IN PROGRESS AT NEXT TIMES RUPT         DAP
# 02001         JET FAILURES HAVE DISABLED Y-Z TRANS.            DAP
# 02002         JET FAILURES HAVE DISABLED X TRANSLATION         DAP
# 02003         JET FAILURES HAVE DISABLED P-ROTATION            DAP
# 02004         JET FAILURES HAVE DISABLED U-V ROTATION          DAP
# 03777         ICDU FAIL CAUSED THE ISS WARNING                 T4RUPT
# 04777         ICDU, PIPA FAILS CAUSED THE ISS WARNING          T4RUPT
# 07777         IMU FAIL CAUSED THE ISS WARNING                  T4RUPT
# 10777         IMU, PIPA FAILS CAUSED THE ISS WARNING           T4RUPT
# 13777         IMU, ICDU FAILS CAUSED THE ISS WARNING           T4RUPT
# 14777         IMU, ICDU, PIPA FAILS CAUSED THE ISS WARNING     T4RUPT
#
#             * INDICATES AN ABORT CODE THAT RESULTS IN A SOFTWARE RESTART.
#
#            ** INDICATES A MORE SERIOUS ABORT CODE THAT RESULTS IN THE
#               PROGRAM GOING TO ROO.
#
#             P INDICATES A PRIORITY ALARM.
#
#               ALL OTHERS ARE NON-ABORTIVE
```

```
# CHECKLIST CODES FOR LUMINARY

# *9             *17              *26                                     *9       COLUMN

# R1CODE                         ACTION TO BE EFFECTED                   PROGRAM

# 00013          KEY IN           NORMAL OR GYRO TORQUE COARSE ALIGN      P52
# 00014          PROCEED          DO IMU FINE ALIGN ROUTINE               P51, P63, P57
# 00014          ENTER            DO LANDING SITE DETERMINATION (N89DISP) P57OPTION2
# 00015          PERFORM          CELESTIAL BODY ACQUISITION              R51, P51
# 00062          SWITCH           AGC POWER DOWN                          P06
# 00201          SWITCH           RR MODE TO AUTOMATIC                    P20, P22, R04
# 00203          SWITCH           GUID CONTROL TO GNC, MODE TO AUTO...    P12, P42, P71
#                                 ALSO THR CONT TO AUTO                   P40, P63, P70
# 00205          PERFORM          MANUAL ACQUISITION OF RR                R23
# 00500          SWITCH           LR ANTENNA TO POSITION 1                P63

#                                 SWITCH DENOTES CHANGE POSITION OF A CONSOLE SWITCH
#                                 PERFORM DENOTES START OR END OF A TASK
#                                 KEY IN DENOTES KEY IN OF DATA THRU THE DSKY
```

```
# OPTION CODES FOR LUMINARY

# THE SPECIFIED OPTION CODES WIL BE FLASHED IN COMPONENT R1 IN
# CONJUNCTION WITH V04N06 OR V04N12 (FOR EXTENDED VERBS) TO REQUEST THE
# ASTRONAUT TO LOAD INTO COMPONENT R2 THE OPTION HE DESIRES.

# *9             *17                             *52                             *11             *25     COLUMN

# OPTION
# CODE           PURPOSE                         INPUT FOR COMPONENT 2           PROGRAM(S)      APPLICABILITY

# 00001          SPECIFY IMU ORIENTATION         1=PREF 2=NOM 3=REFSMMAT         P52             ALL
#                                                4=LAND SITE
# 00002          SPECIFY VEHICLE                 1=THIS 2=OTHER                  P21,R30         ALL
# 00003          SPECIFY TRACKING ATTITUDE       1=PREFERRED 2=OTHER             R63             ALL
# 00004          SPECIFY RADAR                   1=RR 2=LR                       R04             SUNDANCE + LUMINARY
# 00005          SPECIFY SOR PHASE               1=FIRST 2=SECOND                P38             COLOSSUS + LUMINARY
# 00006          SPECIFY RR COARSE ALIGN OPTION  1=LOCKON 2=CONTINUOUS DESIG.    V41N72          SUNDANCE + LUMINARY
# 00010          SPECIFY ALIGNMENT MODE          0=ANY TIME 1=REFSMMAT +G        P57             LUMINARY
#                                                2=TWO BODIES 3=ONE BODY + G
# 00012          SPECIFY CSM ORBIT OPTION        1=NO ORBIT CHANGE 2=CHANGE      P22             LUMINARY
#                                                ORBIT TO PASS OVER LM
```

```
# TAGS FOR RELATIVE SETLOC AND BLANK BANK CARDS

#               COUNT   BANKSUM

# MODULE 1 CONTAINS BANKS 0 THROUGH 5

                BLOCK   02
RADARFF         EQUALS
FFTAG1          EQUALS
FFTAG2          EQUALS
FFTAG3          EQUALS
FFTAG4          EQUALS
FFTAG7          EQUALS
FFTAG8          EQUALS
FFTAG9          EQUALS
FFTAG10         EQUALS
FFTAG11         EQUALS
FFTAG12         EQUALS
FFTAG13         EQUALS
                BNKSUM  02

                BLOCK   03
FFTAG5          EQUALS
FFTAG6          EQUALS
                BNKSUM  03

                BANK    00
DLAYJOB         EQUALS
                BNKSUM  00

                BANK    01
RESTART         EQUALS
LOADDAP1        EQUALS
                BNKSUM  01

                BANK    04
R02             EQUALS
VERB37          EQUALS
PINBALL4        EQUALS
CONICS1         EQUALS
KEYRUPT         EQUALS
R36LM           EQUALS
UPDATE2         EQUALS
E/PROG          EQUALS
AOTMARK2        EQUALS
                BNKSUM  04
```

```
                BANK    05
FRANDRES        EQUALS
DOWNTELM        EQUALS
ABORTS1         EQUALS
EPHEM1          EQUALS
ASENT3          EQUALS
                BNKSUM  05

# MODULE 2 CONTAINS BANKS 6 THROUGH 13

                BANK    06
IMUCOMP         EQUALS
T4RUP           EQUALS
RCSMONT         EQUALS
MIDDGIM         EQUALS
EARTHLOC        EQUALS
                BNKSUM  06

                BANK    07
AOTMARK1        EQUALS
MODESW          EQUALS
ASENT2          EQUALS
                BNKSUM  07

                BANK    10
RTBCODES        EQUALS
DISPLAYS        EQUALS
PHASETAB        EQUALS
FLESHLOC        EQUALS
SLCTMU          EQUALS
                BNKSUM  10

                BANK    11
ORBITAL         EQUALS
F2DPS*11        EQUALS
INTVEL          EQUALS
                BNKSUM  11

                BANK    12
CONICS          EQUALS
ORBITAL1        EQUALS
```

```
INTPRET2        EQUALS
                BNKSUM  12

                BANK    13
LATLONG         EQUALS
INTINIT         EQUALS
LEMGEOM         EQUALS
P76LOC          EQUALS
ORBITAL2        EQUALS
ABTFLGS         EQUALS
                BNKSUM  13
```

```
# MODULE 3 CONTAINS BANKS 14 THROUGH 21

                BANK    14
P50S1           EQUALS
STARTAB         EQUALS
ASENT4          EQUALS
                BNKSUM  14

                BANK    15
P50S            EQUALS
EPHEM           EQUALS
                BNKSUM  15

                BANK    16
DAPS1           EQUALS
                BNKSUM  16

                BANK    17
DAPS2           EQUALS
P40S3           EQUALS
                BNKSUM  17

                BANK    20
DAPS3           EQUALS
LOADDAP         EQUALS
RODTRAP         EQUALS
                BNKSUM  20

                BANK    21
DAPS4           EQUALS
R10             EQUALS
R11             EQUALS
                BNKSUM  21
```

```
# MODULE 4 CONTAINS BANKS 22 THROUGH 27

                BANK    22
KALCMON1        EQUALS
KALCMON2        EQUALS
R30LOC          EQUALS
RENDEZ          EQUALS
SERV2           EQUALS
LANDCNST        EQUALS
                BNKSUM  22

                BANK    23
POWFLITE        EQUALS
POWFLIT1        EQUALS
INFLIGHT        EQUALS
APOPERI         EQUALS
R61             EQUALS
R62             EQUALS
INTPRET1        EQUALS
MEASINC         EQUALS
MEASINC1        EQUALS
EXTVB1          EQUALS
P12A            EQUALS
NORMLIZ         EQUALS
ASENT7          EQUALS
                BNKSUM  23

                BANK    24
PLANTIN         EQUALS
P20S            EQUALS
                BNKSUM  24

                BANK    25
P20S1           EQUALS
P20S2           EQUALS
RADARUPT        EQUALS
RRLEADIN        EQUALS
R29S1           EQUALS
PLANTIN3        EQUALS
                BNKSUM  25

                BANK    26
```

```
P20S3           EQUALS
BAWLANGS        EQUALS
MANUVER         EQUALS
MANUVER1        EQUALS
PLANTIN1        EQUALS
PLANTIN2        EQUALS
                BNKSUM  26

                BANK    27
TOF-FF          EQUALS
TOF-FF1         EQUALS
P40S1           EQUALS
VECPT           EQUALS
ASENT1          EQUALS
SERV3           EQUALS
                BNKSUM  27
```

```
# MODULE 5 CONTAINS BANKS 30 THROUGH 35

                BANK     30
LOWSUPER        EQUALS
P12             EQUALS
ASENT           EQUALS
FCDUW           EQUALS
FLOGSUB         EQUALS
VB67A           EQUALS
ASENT5          EQUALS
                BNKSUM   30

                BANK     31
FTHROT          EQUALS
F2DPS*31        EQUALS
VB67            EQUALS
                BNKSUM   31

                BANK     32
P20S4           EQUALS
F2DPS*32        EQUALS
ABORTS          EQUALS
LRS22           EQUALS
P66LOC          EQUALS
R47             EQUALS
SERV            EQUALS
                BNKSUM   32

                BANK     33
SERVICES        EQUALS
R29/SERV        EQUALS
ASENT6          EQUALS
                BNKSUM   33

                BANK     34
STBLEORB        EQUALS
P30S1           EQUALS
CSI/CDH1        EQUALS
ASCFILT         EQUALS
R12STUFF        EQUALS
SERV4           EQUALS
                BNKSUM   34
```

```
                BANK    35
CSI/CDH         EQUALS
P30S            EQUALS
GLM             EQUALS
P40S2           EQUALS
                BNKSUM  35
```

```
# MODULE 6 CONTAINS BANKS 36 THROUGH 43

                BANK    36
P40S            EQUALS
                BNKSUM  36

                BANK    37
P05P06          EQUALS
IMU2            EQUALS
IMU4            EQUALS
R31             EQUALS
IMUSUPER        EQUALS
SERV1           EQUALS
                BNKSUM  37

                BANK    40
PINBALL1        EQUALS
SELFSUPR        EQUALS
PINSUPER        EQUALS
R31LOC          EQUALS
                BNKSUM  40

                BANK    41
PINBALL2        EQUALS
                BNKSUM  41

                BANK    42
SBAND           EQUALS
PINBALL3        EQUALS
                BNKSUM  42

                BANK    43
EXTVERBS        EQUALS
SELFCHEC        EQUALS
                BNKSUM  43
```

```
HI6ZEROS        EQUALS  ZEROVECS                # ZERO VECTOR ALWAYS IN HIGH MEMORY
LO6ZEROS        EQUALS  ZEROVEC                 # ZERO VECTOR ALWAYS IN LOW MEMORY
HIDPHALF        EQUALS  UNITX
LODPHALF        EQUALS  XUNIT
HIDP1/4         EQUALS  DP1/4TH
LODP1/4         EQUALS  D1/4                    # 2DEC .25
HIUNITX         EQUALS  UNITX
HIUNITY         EQUALS  UNITY
HIUNITZ         EQUALS  UNITZ
LOUNITX         EQUALS  XUNIT                   # 2DEC .5
LOUNITY         EQUALS  YUNIT                   # 2DEC 0
LOUNITZ         EQUALS  ZUNIT                   # 2DEC 0

DELRSPL         EQUALS  SPLRET                  # COL PGM, ALSO CALLED BY R30 IN LUMINARY.

# ROPE-SPECIFIC ASSIGNS OBVIATING NEED TO CHECK COMPUTER FLAG IN DETERMINING INTEGRATION AREA ENTRIES.

ATOPTHIS        EQUALS  ATOPLEM
ATOPOTH         EQUALS  ATOPCSM
OTHPREC         EQUALS  CSMPREC
MOONTHIS        EQUALS  LMOONFLG
MOONOTH         EQUALS  CMOONFLG
MOVATHIS        EQUALS  MOVEALEM
RMM             =       LODPMAX
RME             =       LODPMAX1
THISPREC        EQUALS  LEMPREC
THISAXIS        =       UNITZ
NB1NB2          EQUALS  THISAXIS                # FOR R31
ERASID          EQUALS  BITS2-10                # DOWNLINK ERASABLE DUMP ID
DELAYNUM        EQUALS  TWO
```

```
# DPS AND APS ENGINE PARAMETERS

                SETLOC  P40S
                BANK
                COUNT*  $$/P40

# *** THE ORDER OF THE FOLLOWING SIX CONSTANTS MUST NOT BE CHANGED ***

FDPS            2DEC    4.3670 B-7              # 9817.5 LBS FORCE IN NEWTONS
MDOTDPS         2DEC    0.1480 B-3              # 32.62 LBS/SEC IN KGS/CS
DTDECAY         2DEC    -38
FAPS            2DEC    1.5569 B-7              # 3500 LBS FORCE IN NEWTONS
MDOTAPS         2DEC    0.05135 B-3             # 11.32 LBS/SEC IN KGS/CS
ATDECAY         2DEC    -10

# ********************************************************************

FRCS4           2DEC    0.17792 B-7             # 400 LBS FORCE IN NEWTONS
FRCS2           2DEC    0.08896 B-7             # 200 LBS FORCE IN NEWTONS

                SETLOC  P40S1
                BANK
                COUNT*  $$/P40

# *** APS IMPULSE DATA FOR P42 ****************************************

K1VAL           2DEC    124.55 B-23             # 2800 LB-SEC
K2VAL           2DEC    31.138 B-24             # 700 LB-SEC
K3VAL           2DEC    1.5569 B-10             # FAPS (3500 LBS THRUST)

# ********************************************************************

S40.136         2DEC    .4671 B-9               # .4671 M NEWTONS (DPS)
S40.136_        2DEC    .4671 B+1               # S40.136 SHIFTED LEFT 10.

                SETLOC  ASENT1
                BANK
                COUNT*  $$/P70

(1/DV)A         2DEC    15.20 B-7               # 2 SECONDS WORTH OF INITIAL ASCENT
```

```
                                                    # STAGE ACCELERATION -- INVERTED (M/CS)
                                                    # 1) PREDICATED ON A LIFTOFF MASS OF
                                                    #    4869.9 KG (SNA-8-D-027 7/11/68)
                                                    # 2) PREDICATED ON A CONTRIBUTION TO VEH-
                                                    #    ICLE ACCELERATION FROM RCS THRUSTERS
                                                    #    EQUIV. TO 1 JET ON CONTINUOUSLY.

K(1/DV)         2DEC    436.70 B-9                  # DPS ENGINE THRUST IN NEWTONS / 100 CS.

(AT)A           2DEC    3.2883 E-4 B9               # INITIAL ASC. STG. ACCELERATION ** M/CS.
                                                    # ASSUMPTIONS SAME AS FOR (1/DV)A.
(TBUP)A         2DEC    91902 B-17                  # ESTIMATED BURN-UP TIME OF THE ASCENT STG.
                                                    # ASSUMPTIONS SAME AS FOR (1/DV)A WITH THE
                                                    # ADDITIONAL ASSUMPTION THAT NET MASS-FLOW
                                                    # RATE = 5.299 KG/SEC = 5.135 (APS) +
                                                    # .164 (1 RCS JET).
                SETLOC  ASENT
                BANK
                COUNT*  $$/ASENT
AT/RCS          2DEC    .0000785 B+10               # 4 JETS IN A DRY LEM

                SETLOC  SERVICES
                BANK
                COUNT*  $$/SERV

# *** THE ORDER OF THE FOLLOWING TWO CONSTANTS MUST NOT BE CHANGED *******

APSVEX          DEC     -3030 E-2 B-5               # 9942 FT/SEC IN M/CS.
DPSVEX          DEC*    -2.95588868 E+1 B-05*       # VE (DPS) +2.95588868E+ 3

# **************************************************************************

                SETLOC  F2DPS*31
                BANK
                COUNT*  $$/F2DPS

TRIMACCL        2DEC*   +3.50132708 E-5 B+08*       # A (T) +3.50132708E- 1
```

```
# THROTTLING AND THRUST DETECTION PARAMETERS

                SETLOC  P40S
                BANK
                COUNT*  $$/P40

THRESH1         DEC     24
THRESH3         DEC     12
HIRTHROT        =       BIT13

                SETLOC  FFTAG5
                BANK
                COUNT*  $$/P40

THRESH2         DEC     308

                SETLOC  FTHROT
                BANK
                COUNT*  $$/THROT

FMAXODD         DEC     +3841                   # FSAT          +4.81454413 E+4
FMAXPOS         DEC     +3467                   # FMAX          +4.34546769 E+4
THROTLAG        DEC     +20                     # TAU (TH)      +1.99999999 E-1
SCALEFAC        2DEC*   +7.97959872 E+2 B-16*   # BITPERF       +7.97959872 E-2

                SETLOC  F2DPS*32
                BANK
                COUNT*  $$/F2DPS

DPSTHRSH        DEC     36                      # (THRESH1 + THRESH3 FOR P63)
```

```
# LM HARDWARE-RELATED PARAMETERS

                SETLOC  RADARUPT
                BANK
                COUNT*  $$/RRUPT

LVELBIAS        DEC     -12288                  # LANDING RADAR BIAS FOR 153.6 KC.
RDOTBIAS        2DEC    17000                   # BIAS COUNT FOR RR RANGE RATE.

                SETLOC  LRS22
                BANK
                COUNT*  $$/LRS22

RDOTCONV        2DEC    -.0019135344 B7         # CONVERTS RR RDOT READING TO M/CS AT 2(7)
RANGCONV        2DEC    2.859024 B-3            # CONVERTS RR RANGE READING TO M. AT 2(-29

                SETLOC  SERVICES
                BANK
                COUNT*  $$/SERV

HBEAMANT        2DEC    -.4687018041            # RANGE BEAM IN LR ANTENNA COORDINATES.
                2DEC    0
                2DEC    -.1741224271

HSCAL           2DEC    -.3288792               # SCALES 1.079 FT/BIT TO 2(22)M.

# ***** THE SEQUENCE OF THE FOLLOWING CONSTANTS MUST BE PRESERVED *********

VZSCAL          2DEC    +.5410829105            # SCALES .8668 FT/SEC/BIT TO 2(18) M/CS.
VYSCAL          2DEC    +.7565672446            # SCALES 1.212 FT/SEC/BIT TO 2(18) M/CS.
VXSCAL          2DEC    -.4020043770            # SCALES -.644 FT/SEC/BIT TO 2(18) M/CS.

# ************************************************************************

KPIP            DEC     .0512                   # SCALES DELV TO UNITS OF 2(5) M/CS.
KPIP1           2DEC    .0128                   # SCALES DELV TO UNITS OF 2(7) M/CS.
KPIP2           2DEC    .0064                   # SCALES DELV TO UNITS OF 2(8) M/CS.
```

```
ALTCONV         2DEC    1.399078846 B-4         # CONVERTS M*2(-24) TO BIT UNITS *2(-28).
ARCONV1         2DEC    656.167979 B-10         # CONV. ALTRATE COMP. TO BIT UNITS<

                SETLOC  R10
                BANK
                COUNT*  $$/R10

ARCONV          OCT     24402                   # 656.1679798B-10 CONV ALTRATE TO BIT UNIT
ARTOA           DEC     .1066098 B-1            # .25/2.345 B-1 4X/SEC CYCLE RATE.
ARTOA2          DEC     .0021322 B8             # (.5)/(2.345)(100)
VELCONV         OCT     22316                   # 588.914 B-10 CONV VEL. TO BIT UNITS.
KPIP1(5)        DEC     .0512                   # SCALES DELV TO M/CS*2(-5).
MAXVBITS        OCT     00547                   # MAX. DISPLAYED VELOCITY 199.9989 FT/SEC.

                SETLOC  DAPS3
                BANK
                COUNT*  $$/DAPAO

TORKJET1        DEC     .03757                  # 550 / .2 SCALED AT (+16) 64 / 180
```

```
# PARAMETERS RELATING TO MASS, INERTIA, AND VEHICLE DIMENSIONS

                SETLOC  FRANDRES
                BANK
                COUNT*  $$/START

FULLAPS         DEC     5050 B-16           # NOMINAL FULL ASCENT MASS -- 2(16) KG.

                SETLOC  LOADDAP1
                BANK
                COUNT*  $$/R03

MINLMD          DEC     -2850 B-16          # MIN. DESCENT STAGE MASS -- 2(16) KG.
MINMINLM        DEC     -2200 B-16          # MIN ASCENT STAGE MASS -- 2(16) KG.
MINCSM          =       BIT11               # MIN CSM MASS (OK FOR 1/ACCS) = 9050 LB

                SETLOC  DAPS3
                BANK
                COUNT*  $$/DAPAD

LOASCENT        DEC     2200 B-16           # MIN ASCENT LEM MASS -- 2(16) KG.
HIDESCNT        DEC     15300 B-16          # MAX DESCENT LEM MASS -- 2(16) KG.
LODESCNT        DEC     1750 B-16           # MIN DESCENT STAGE (ALONE) -- 2(16) KG.
```

```
# PHYSICAL CONSTANTS ( TIME - INVARIANT )

                SETLOC  IMU2
                BANK
                COUNT*  $$/P07

OMEG/MS         2DEC    .24339048

                SETLOC  R30LOC
                BANK
                COUNT*  $$/R30

# *** THE ORDER OF THE FOLLOWING TWO CONSTANTS MUST BE PRESERVED ***********

1/RTMUM         2DEC*   .45162595 E-4 B14*
1/RTMUE         2DEC*   .50087529 E-5 B17*

# ************************************************************************

                SETLOC  P40S1
                BANK
                COUNT*  $$/S40.9

EARTHMU         2DEC*   -3.986032 E10 B-36*     # M(3)/CS(2)

                SETLOC  ASENT1
                BANK
                COUNT*  $$/P12

MUM(-37)        2DEC*   4.9027780 E8 B-37*
MOONRATE        2DEC*   .26616994890062991 E-7 B+19*    # RAD/CS.

                SETLOC  SERVICES
                BANK
                COUNT*  $$/SERV

# *** THE ORDER OF THE FOLLOWING TWO CONSTANTS MUST BE PRESERVED ***********

-MUDT           2DEC*   -7.9720645 E+12 B-44*
-MUDT1          2DEC*   -9.8055560 E+10 B-44*

# ************************************************************************

-MUDTMUN        2DEC*   -9.8055560 E+10 B-38*
RESQ            2DEC*   40.6809913 E12 B-58*
```

```
20J             2DEC    3.24692010 E-2
2J              2DEC    3.24692010 E-3

                SETLOC  P50S1
                BANK
                COUNT*  $$/LOSAM

RSUBEM          2DEC    384402000 B-29
RSUBM           2DEC    1738090 B-29
RSUBE           2DEC    6378166 B-29
ROE             2DEC    .00257125

                SETLOC  CONICS1
                BANK
                COUNT*  $$/LT-LG

ERAD            2DEC    6373338 B-29            # PAD RADIUS
504RM           2DEC    1738090 B-29            # METERS B-29 (EQUATORIAL MOON RADIUS)

                SETLOC  CONICS1
                BANK
                COUNT*  $$/CONIC

# *** THE ORDER OF THE FOLLOWING CONSTANTS MUST BE PRESERVED **************

MUTABLE         2DEC*   3.986032 E10 B-36*      # MUE
                2DEC*   .25087606 E-10 B+34*    # 1/MUE
                2DEC*   1.99650495 E5 B-18*     # SQRT(MUE)
                2DEC*   .50087529 E-5 B+17*     # 1/SQRT(MUE)
                2DEC*   4.902778 E8 B-30*       # MUM
                2DEC*   .203966 E-8 B+28*       # 1/MUM
                2DEC*   2.21422176 E4 B-15*     # SQRT(MUM)
                2DEC*   .45162595 E-4 B+14*     # 1/SQRT(MUM)

# ***************************************************************************
```

```
                SETLOC  INTINIT
                BANK
                COUNT*  $$/INTIN

OMEGMOON        2DEC*   2.66169947 E-8 B+23*

                SETLOC  ORBITAL2
                BANK
                COUNT*  $$/ORBIT

# *** THE ORDER OF THE FOLLOWING CONSTANTS MUST NOT BE CHANGED ************

                2DEC*   1.32715445 E16 B-54*
MUM             2DEC*   4.9027780 E8 B-30*
MUEARTH         2DEC*   3.986032 E10 B-36*
                2DEC    0
J4REQ/J3        2DEC*   .4991607391 E7 B-26*
                2DEC    -176236.02 B-25
2J3RE/J2        2DEC*   -.1355426363 E5 B-27*
                2DEC*   .3067493316 E18 B-60*
J2REQSQ         2DEC*   1.75501139 E21 B-72*
3J22R2MU        2DEC*   9.20479048 E16 B-58*

# *************************************************************************

                SETLOC  TOF-FF1
                BANK
                COUNT*  $$/TFF

1/RTMU          2DEC*   .5005750271 E-5 B17*    # MODIFIED EARTH MU

                SETLOC  SBAND
                BANK
                COUNT*  $$/R05

REMDIST         2DEC    384402000 B-29          # MEAN DISTANCE BETWEEN EARTH AND MOON.
```

```
# PHYSICAL CONSTANTS (TIME - VARIANT)

                SETLOC  STARTAB
                BANK
                COUNT*  $$/STARS

                2DEC    +.8342971408 B-1        # STAR 37       X
                2DEC    -.2392481515 B-1        # STAR 37       Y
                2DEC    -.4966976975 B-1        # STAR 37       Z

                2DEC    +.8139832631 B-1        # STAR 36       X
                2DEC    -.5557243189 B-1        # STAR 36       Y
                2DEC    +.1691204557 B-1        # STAR 36       Z

                2DEC    +.4541086270 B-1        # STAR 35       X
                2DEC    -.5392368197 B-1        # STAR 35       Y
                2DEC    +.7092312789 B-1        # STAR 35       Z

                2DEC    +.3201817378 B-1        # STAR 34       X
                2DEC    -.4436021946 B-1        # STAR 34       Y
                2DEC    -.8370786986 B-1        # STAR 34       Z

                2DEC    +.5520184464 B-1        # STAR 33       X
                2DEC    -.7933187400 B-1        # STAR 33       Y
                2DEC    -.2567508745 B-1        # STAR 33       Z

                2DEC    +.4537196908 B-1        # STAR 32       X
                2DEC    -.8779508801 B-1        # STAR 32       Y
                2DEC    +.1527766153 B-1        # STAR 32       Z

                2DEC    +.2069525789 B-1        # STAR 31       X
                2DEC    -.8719885748 B-1        # STAR 31       Y
                2DEC    -.4436288486 B-1        # STAR 31       Z

                2DEC    +.1217293692 B-1        # STAR 30       X
                2DEC    -.7702732847 B-1        # STAR 30       Y
```

```
                2DEC    +.6259880410 B-1        # STAR 30       Z

                2DEC    -.1124304773 B-1        # STAR 29       X
                2DEC    -.9694934200 B-1        # STAR 29       Y
                2DEC    +.2178116072 B-1        # STAR 29       Z

                2DEC    -.1146237858 B-1        # STAR 28       X
                2DEC    -.3399692557 B-1        # STAR 28       Y
                2DEC    -.9334250333 B-1        # STAR 28       Z

                2DEC    -.3516499609 B-1        # STAR 27       X
                2DEC    -.8240752703 B-1        # STAR 27       Y
                2DEC    -.4441196390 B-1        # STAR 27       Z

                2DEC    -.5326876930 B-1        # STAR 26       X
                2DEC    -.7160644554 B-1        # STAR 26       Y
                2DEC    +.4511047742 B-1        # STAR 26       Z

                2DEC    -.7861763936 B-1        # STAR 25       X
                2DEC    -.5217996305 B-1        # STAR 25       Y
                2DEC    +.3311371675 B-1        # STAR 25       Z

                2DEC    -.6898393233 B-1        # STAR 24       X
                2DEC    -.4182330640 B-1        # STAR 24       Y
                2DEC    -.5909338474 B-1        # STAR 24       Z

                2DEC    -.5812035376 B-1        # STAR 23       X
                2DEC    -.2909171294 B-1        # STAR 23       Y
                2DEC    +.7599800468 B-1        # STAR 23       Z

                2DEC    -.9170097662 B-1        # STAR 22       X
                2DEC    -.3502146628 B-1        # STAR 22       Y
                2DEC    -.1908999176 B-1        # STAR 22       Z
```

```
                2DEC    -.4523440203 B-1        # STAR 21       X
                2DEC    -.0493710140 B-1        # STAR 21       Y
                2DEC    -.8904759346 B-1        # STAR 21       Z

                2DEC    -.9525211695 B-1        # STAR 20       X
                2DEC    -.0593434796 B-1        # STAR 20       Y
                2DEC    -.2986331746 B-1        # STAR 20       Z

                2DEC    -.9656605484 B-1        # STAR 19       X
                2DEC    +.0525933156 B-1        # STAR 19       Y
                2DEC    +.2544280809 B-1        # STAR 19       Z

                2DEC    -.8608205219 B-1        # STAR 18       X
                2DEC    +.4636213989 B-1        # STAR 18       Y
                2DEC    +.2098647835 B-1        # STAR 18       Z

                2DEC    -.7742591356 B-1        # STAR 17       X
                2DEC    +.6152504197 B-1        # STAR 17       Y
                2DEC    -.1482892839 B-1        # STAR 17       Z

                2DEC    -.4657947941 B-1        # STAR 16       X
                2DEC    +.4774785033 B-1        # STAR 16       Y
                2DEC    +.7450164351 B-1        # STAR 16       Z

                2DEC    -.3612508532 B-1        # STAR 15       X
                2DEC    +.5747270840 B-1        # STAR 15       Y
                2DEC    -.7342932655 B-1        # STAR 15       Z

                2DEC    -.4118589524 B-1        # STAR 14       X
                2DEC    +.9065485360 B-1        # STAR 14       Y
                2DEC    +.0924226975 B-1        # STAR 14       Z

                2DEC    -.1820751783 B-1        # STAR 13       X
```

```
                2DEC    +.9404899869 B-1        # STAR 13       Y
                2DEC    -.2869271926 B-1        # STAR 13       Z

                2DEC    -.0614937230 B-1        # STAR 12       X
                2DEC    +.6031563286 B-1        # STAR 12       Y
                2DEC    -.7952489957 B-1        # STAR 12       Z

                2DEC    +.1371725575 B-1        # STAR 11       X
                2DEC    +.6813721061 B-1        # STAR 11       Y
                2DEC    +.7189685267 B-1        # STAR 11       Z

                2DEC    +.2011399589 B-1        # STAR 10       X
                2DEC    +.9690337941 B-1        # STAR 10       Y
                2DEC    -.1432348512 B-1        # STAR 10       Z

                2DEC    +.3507315038 B-1        # STAR 9        X
                2DEC    +.8926333307 B-1        # STAR 9        Y
                2DEC    +.2831839492 B-1        # STAR 9        Z

                2DEC    +.4105636020 B-1        # STAR 8        X
                2DEC    +.4988110001 B-1        # STAR 8        Y
                2DEC    +.7632988371 B-1        # STAR 8        Z

                2DEC    +.7032235469 B-1        # STAR 7        X
                2DEC    +.7075846047 B-1        # STAR 7        Y
                2DEC    +.0692868685 B-1        # STAR 7        Z

                2DEC    +.5450107404 B-1        # STAR 6        X
                2DEC    +.5314955466 B-1        # STAR 6        Y
                2DEC    -.6484410356 B-1        # STAR 6        Z

                2DEC    +.0130968840 B-1        # STAR 5        X
                2DEC    +.0078062795 B-1        # STAR 5        Y
```

```
                2DEC     +.9998837600 B-1          # STAR 5         Z

                2DEC     +.4917678276 B-1          # STAR 4         X
                2DEC     +.2204887125 B-1          # STAR 4         Y
                2DEC     -.8423473935 B-1          # STAR 4         Z

                2DEC     +.4775639450 B-1          # STAR 3         X
                2DEC     +.1166004340 B-1          # STAR 3         Y
                2DEC     +.8708254803 B-1          # STAR 3         Z

                2DEC     +.9342640400 B-1          # STAR 2         X
                2DEC     +.1735073142 B-1          # STAR 2         Y
                2DEC     -.3115219339 B-1          # STAR 2         Z

                2DEC     +.8748658918 B-1          # STAR 1         X
                2DEC     +.0260879174 B-1          # STAR 1         Y
                2DEC     +.4836621670 B-1          # STAR 1         Z

CATLOG          DEC      6970

# ********************************************************************************

                SETLOC   EPHEM1
                BANK
                COUNT*   $$/EPHEM

KONMAT          2DEC     1.0 B-1                   # ********************
                2DEC     0                         #                    *
                2DEC     0                         #                    *
                2DEC     0                         #                    *
                2DEC     .91745 B-1                # K1 COS(OBL)        *
                2DEC     -.03571 B-1               # K2 SIN(OBL)SIN(IM) *
                2DEC     0                         #                    *
                2DEC     .39784 B-1                # K3 SIN(OBL)        *
```

```
                2DEC    .082354 B-1             # K4 COS(OBL)SIN(IM) *
CSTODAY         2DEC    8640000 B-33            #                    * NOTE:           *
RCB-13          OCT     00002                   #                    * TABLES CONTAIN *
                OCT     00000                   #                    * CONSTANTS FOR  *
RATESP          2DEC    .03660098 B+4           # LOMR               * 1969 - 1970    *
                2DEC    .00273779 B+4           # LOSR
                2DEC    -.00014719 B+4          # LONR
                2DEC    .815282336              # LOMO
                2DEC    .274674910              # LOSO
                2DEC    .986209499              # LONO
VAL67           2DEC*   .01726666666 B+1*       # AMOD
                2DEC    .530784445              # AARG
                2DEC    .036291712 B+1          # 1/27
                2DEC    .003505277 B+1          # BMOD
                2DEC    .585365625              # BARG
                2DEC    .03125 B+1              # 1/32
                2DEC    .005325277 B+1          # CMOD
                2DEC    -.01106341036           # CARG
                2DEC    .002737925 B+1          # 1/365

# ********************************************************************************

                SETLOC  PLANTIN2
                BANK
                COUNT*  $$/LUROT

COSI            2DEC    .99964173 B-1           # COS (5521.5 SEC.) B-1
SINI            2DEC    .02676579 B-1           # SIN (5521.5 SEC.) B-1
NODDOT          2DEC    -.457335121 E-2         # REV/CSEC B+28 = -1.07047011 E-8 RAD/SEC
FDOT            2DEC    .570863327              # REV/CSEC B+27 =  2.67240410 E-6 RAD/SEC
```

```
BDOT            2DEC    -3.07500686 E-8        # REV/CSEC B+28 = -7.19757301 E-14 RAD/SEC
NODIO           2DEC    .986209434             # REVS B-D      = 6.19653663041 RAD
FSUBO           2DEC    .829090536             # REVS B-D      = 5.20932947829 RAD
BSUBO           2DEC    .0651201393            # REVS B-D      = 0.40916190299 RAD
WEARTH          2DEC    .973561595             # REV/CSEC B+23 = 7.29211494 E-5 RAD/SEC
```

```
# ***   CHANNEL DESCRIPTIONSF WORDS ARE ALLOCATED IN ERASABLE ASSIGNMENTS ***

#       CHANNEL 1       IDENTICAL TO COMPUTER REGISTER L (0001)

#       CHANNEL 2       IDENTICAL TO COMPUTER REGISTER Q (0002)

#       CHANNEL 3       HISCALAR; INPUT CHANNEL; MOST SIGNIFICANT 14 BITS FROM 33 STAGE BINARY COUNTER. SCALE
#                       FACTOR IS B23 IN CSEC, SO MAX VALUE ABOUT 23.3 HOURS AND LEAST SIGNIFICANT BIT 5.12 SECS.

#       CHANNEL 4       LOSCALAR; INPUT CHANNEL; NEXT MOST SIGNIFICANT 14 BITS FROM THE 33 STAGE BINARY COUNTER
#                       ASSOCIATED WITH CHANNEL 3. SCALE FACTOR IS B9 IN  CSEC. SO MAX VAL IS 5.12 SEC AND LEAST
#                       SIGNIFICANT BIT IS 1/3200 SEC. SCALE FACTOR OF D.P. WORD WITH CHANNEL 3 IS B23 CSEC.

#       CHANNEL 5       PYJETS; OUTPUT CHANNEL; PITCH RCS JET CONTROL.   (REACTION CONTROL SYSTEM) USES BITS 1-8.

#       CHANNEL 6       ROLLJETS; OUTPUT CHANNEL; ROLL RCS JET CONTROL.   (REACTION CONTROL SYSTEM) USES BIT 1-8.

#       CHANNEL 7       SUPERBNK; OUTPUT CHANNEL; NOT RESET BY RESTART;   FIXED EXTENSION BITS USED TO SELECT THE
#                       APPROPRIATE FIXED MEMORY BANK IF FBANK IS 30 OCTAL OR MORE. USES BITS 5-7.

#       CHANNEL 10      OUTO; OUTPUT CHANNEL; REGISTER USED TO TRANSMIT    LATCHING-RELAY DRIVING INFORMATION FOR
#                       THE DISPLAY SYSTEM.  BITS 15-12 ARE SET TO THE ROW NUMBER (1-14 OCTAL) OF THE RELAY TO BE
#                       CHANGED AND BITS 11-1 CONTAIN THE REQUIRED SETTINGS FOR THE RELAYS IN THE ROW.

#       CHANNEL 11      DSALMOUT; OUTPUT CHANNEL; REGISTER WHOSE BITS ARE USED FOR ENGINE ON-OFF CONTROL AND TO
#                       DRIVE INDIVIDUAL INDICATORS OF THE DISPLAY SYSTEM. BITS 1-7 ARE A RELAYS.
#
#         BIT 1         ISS WARNING
#         BIT 2         LIGHT COMPUTER ACTIVITY LAMP
#         BIT 3         LIGHT UPLINK ACTIVITY LAMP
#         BIT 4         LIGHT TEMP CAUTION LAMP
#         BIT 5         LIGHT KEYBOARD RELEASE LAMP
#         BIT 6         FLASH VERB AND NOUN LAMPS
#         BIT 7         LIGHT OPERATOR ERROR LAMP
```

```
#          BIT 8          SPARE
#          BIT 9          TEST CONNECTOR OUTBIT
#          BIT 10         CAUTION RESET
#          BIT 11         SPARE
#          BIT 12         SPARE
#          BIT 13         ENGINE ON
#          BIT 14         ENGINE OFF
#          BIT 15         SPARE

#        CHANNEL 12       CHAN12; OUTPUT CHANNEL; BITS USED TO DRIVE NAVIGATION AND SPAECRAFT HARDWARE
#
#          BIT 1          ZERO RR CDU; CDU'S GIVE RRADAR INFORMATION FOR LM
#          BIT 2          ENABLE CDU RADAR ERROR COUNTERS
#          BIT 3          NOT USED
#          BIT 4          COARSE ALIGN ENABLE OF IMU
#          BIT 5          ZERO IMU CDU'S
#          BIT 6          ENABLE IMU ERROR COUNTER, CDU ERROR COUNTER.
#          BIT 7          SPARE
#          BIT 8          DISPLAY INERTIAL DATA
#          BIT 9          -PITCH GIMBAL TRIM (BELL MOTION) DESCENT ENGINE
#          BIT 10         +PITCH GIMBAL TRIM (BELL MOTION) DESCENT ENGINE
#          BIT 11         -ROLL  GIMBAL TRIM (BELL MOTION) DESCENT ENGINE
#          BIT 12         +ROLL  GIMBAL TRIM (BELL MOTION) DESCENT ENGINE
#          BIT 13         LR POSITION 2 COMMAND
#          BIT 14         ENABLE RENDESVOUS RADAR LOCK-ON;AUTO ANGLE TRACK'G
#          BIT 15         ISS TURN ON DELAY COMPLETE
```

```
#       CHANNEL 13      CHAN13; OUTPUT CHANNEL.
#
#         BIT 1         RADAR C         PROPER SETTING OF THE A,B,C MATRIX
#         BIT 2         RADAR B         SELECTS CERTAIN RADAR
#         BIT 3         RADAR A         PARAMETERS TO BE READ.
#         BIT 4         RADAR ACTIVITY
#         BIT 5         NOT USED (CONNECTS AN ALTERNATE INPUT TO UPLINK)
#         BIT 6         BLOCK INPUTS TO UPLINK CELL
#         BIT 7         DOWNLINK TELEMETRY WORD ORDER CODE BIT
#         BIT 8         RHC COUNTER ENABLE (READ HAND CONTROLLER ANGLES)
#         BIT 9         START RHC READ INTO COUNTERS IF BIT 8 SET
#         BIT 10        TEST ALARMS, TEST DSKY LIGHTS
#         BIT 11        ENABLE STANDBY
#         BIT 12        RESET TRAP 31-A         ALWAYS APPEAR TO BE SET TO 0
#         BIT 13        RESET TRAP 31-B         ALWAYS APPEAR TO BE SET TO 0
#         BIT 14        RESET TRAP 32           ALWAYS APPEAR TO BE SET TO 0
#         BIT 15        ENABLE T6 RUPT

#       CHANNEL 14      CHAN14; OUTPUT CHANNEL; USED TO CONTROL COMPUTER  COUNTER CELLS (CDU,GYRO,SPACECRAFT FUNC.
#
#         BIT 1         OUTLINK ACTIVITY (NOT USED)
#         BIT 2         ALTITUDE RATE OR ALTITIDE SELECTOR
#         BIT 3         ALTITUDE METER ACTIVITY
#         BIT 4         THRUST DRIVE ACTIVITY FOR DESCENT ENGINE
#         BIT 5         SPARE
#         BIT 6         GYRO ENABLE POWER FOR PULSES
#         BIT 7         GYRO SELECT B           PAIR OF BITS IDENTIFIES AXIS OF -
#         BIT 8         GYRO SELECT A           GYRO SYSTEM TO BE TORQUED.
#         BIT 9         GYRO TORQUING COMMAND IN NEGATIVE DIRECTION
```

```
#          BIT 10        GYRO ACTIVITY
#          BIT 11        DRIVE CDU S
#          BIT 12        DRIVE CDU T
#          BIT 13        DRIVE CDU Z
#          BIT 14        DRIVE CDU Y
#          BIT 15        DRIVE CDU X

#        CHANNEL 15      MNKEYIN; INPUT CHANNEL;KEY CODE INPUT FROM KEYBOARD OF DSKY, SENSED BY PROGRAM WHEN
#                        PROGRAM INTERRUPT #5 IS RECEIVED. USES BITS 5-1

#        CHANNEL 16      NAVKEYIN; INPUT CHANNEL; OPTICS MARK INFORMATION AND NAVIGA ION PANEL DSKY (CM) OR THRUST
#                        CONTROL (LM) SENSED BY PROGRAM WHEN PROGRAM INTER-RUPT #6 IS RECEIVED. USES BITS 3-7 ONLY.
#
#          BIT 1         NOT ASSIGNED
#          BIT 2         NOT ASSIGNED
#          BIT 3         OPTICS X-AXIS MARK SIGNAL FOR ALIGN OPTICAL TSCOPE
#          BIT 4         OPTICS Y-AXIS MARK SIGNAL FOR AOT
#          BIT 5         OPTICS MARK REJECT SIGNAL
#          BIT 6         DESCENT+ ; CREW DESIRED SLOWING RATE OF DESCENT
#          BIT 7         DESCENT- ; CREW DESIRED SPEEDING UP RATE OF D'CENT

# NOTE: ALL BITS IN CHANNELS 30-33 ARE INVERTED AS SENSED BY THE  PROGRAM, SO THAT A VALUE OF ZERO MEANS
# THAT THE INDICATED SIGNAL IS PRESENT.

#        CHANNEL 30      INPUT CHANNEL
#
#          BIT 1         ABORT WITH DESCENT STAGE
#          BIT 2            UNUSED
#          BIT 3         ENGINE ARMED SIGNAL
#          BIT 4         ABORT WITH ASCENT ENGINE STAGE
#          BIT 5         AUTO THROTTLE; COMPUTER CONTROL OF DESCENT ENGINE
```

```
#          BIT 6          DISPLAY INERTIAL DATA
#          BIT 7          RR CDU FAIL
#          BIT 8          SPARE
#          BIT 9          IMU OPERATE WITH NO MALFUNCTION
#          BIT 10         LM COMPUTER (NOT AGS) HAS CONTROL OF LM
#          BIT 11         IMU CAGE COMMAND TO DRIVE IMU GIMBAL ANGLES TO 0.
#          BIT 12         IMU CDU FAIL (MALFUNCTION OF IMU CDU,S)
#          BIT 13         IMU FAIL (MALFUNCTION OF IMU STABILIZATION LOOPS)
#          BIT 14         ISS TURN ON REQUESTED
#          BIT 15         TEMPERATURE OF STABLE MEMBER WITHIN DESIGN LIMITS

#        CHANNEL 31       INPUT CHANNEL; BITS ASSOCIATED WITH THE ATTITUDE  CONTROLLER, TRANSLATIONAL CONTROLLER,
#                         AND SPACECRAFT ATTITUDE CONTROL; USED BY RCS DAP
#
#          BIT 1          ROTATION (BY RHC) COMMANDED IN POSITIVE PITCH DIRECTION; MUST BE IN MINIMUM IMPULSE MODE.
#                         ALSO POSITIVE ELEVATION CHANGE FOR LANDING POINT  DESIGNATOR
#          BIT 2          AS BIT 1 EXCEPT NEGATIVE PITCH AND ELEVATION
#          BIT 3          ROTATION (BY RHC) COMMANDED IN POSITIVE YAW DIRECTION; MUST BE IN MINIMUM IMPULSE MODE.
#          BIT 4          AS BIT 3 EXCEPT NEGATIVE YAW
#          BIT 5          ROTATION (BY RHC) COMMANDED IN POSITIVE ROLL DIRECTION; MUST BE IN MINIMUM IMPULSE MODE.
#                         ALSO POSITIVE AZIMUTH CHANGE FOR LANDING POINT DESIGNATOR
#          BIT 6          AS BIT 5 EXCEPT NEGATIVE ROLL AND AZIMUTH
#          BIT 7          TRANSLATION IN +X DIRECTION COMMANDED BY THC
#          BIT 8          TRANSLATION IN -X DIRECTION COMMANDED BY THC
#          BIT 9          TRANSLATION IN +Y DIRECTION COMMANDED BY THC
#          BIT 10         TRANSLATION IN -Y DIRECTION COMMANDED BY THC
#          BIT 11         TRANSLATION IN +Z DIRECTION COMMANDED BY THC
#          BIT 12         TRANSLATION IN -Z DIRECTION COMMANDED BY THC
```

```
#          BIT 13        ATTITUDE HOLD MODE ON SCS MODE CONTROL SWITCH
#          BIT 14        AUTO STABILIZATION OF ATTITUDE ON SCS MODE SWITCH
#          BIT 15        ATTITUDE CONTROL OUT OF DETENT (RHC NOT IN NEUTRAL

#        CHANNEL 32         INPUT CHANNEL.
#
#          BIT 1            THRUSTERS 2 & 4 DISABLED BY CREW
#          BIT 2            THRUSTERS 5 & 8 DISABLED BY CREW
#          BIT 3            THRUSTERS 1 & 3 DISABLED BY CREW
#          BIT 4            THRUSTERS 6 & 7 DISABLED BY CREW
#          BIT 5            THRUSTERS 14 & 16 DISABLED BY CREW
#          BIT 6            THRUSTERS 13 & 15 DISABLED BY CREW
#          BIT 7            THRUSTERS 9 & 12 DISABLED BY CREW
#          BIT 8            THRUSTERS 10 & 11 DISABLED BY CREW
#          BIT 9            DESCENT ENGINE GIMBALS DISABLED BY CREW
#          BIT 10           APPARENT DESCENT ENGINE GIMBAL FAILURE
#          BIT 14        INDICATES PROCEED KEY IS DEPRESSED

#        CHANNEL 33      CHAN33; INPUT CHANNEL; FOR HARDWARE STATUS AND COMMAND INFORMATION. BITS 15-11 ARE FLIP-
#                        FLOP BITS RESET BY A CHANNEL "WRITE" COMMAND THAT ARE RESET BY A RESTART & BY T4RUPT LOOP.
#
#          BIT 1         SPARE
#          BIT 2         RR AUTO-POWER ON
#          BIT 3         RR RANGE LOW SCALE
#          BIT 4         RR DATA GOOD
#          BIT 5         LR RANGE DATA GOOD
#          BIT 6         LR POS1
#          BIT 7         LR POS2
```

```
#          BIT 8          LR VEL DATA GOOD
#          BIT 9          LR RANGE LOW SCALE
#          BIT 10         BLOCK UPLINK INPUT
#          BIT 11         UPLINK TOO FAST
#          BIT 12         DOWNLINK TOO FAST
#          BIT 13         PIPA FAIL
#          BIT 14         WARNING OF REPEATED ALARMS: RESTART,COUNTER FAIL, VOLTAGE FAIL,AND SCALAR DOUBLE.
#          BIT 15         LGC OSCILLATOR STOPPED

#        CHANNEL 34      DNT M1; OUTPUT CHANNEL; DOWNLINK 1  FIRST OF TWO WORDS SERIALIZATION.
#        CHANNEL 35      DNT M2; OUTPUT CHANNEL DOWNLINK 2 SOCOND OF TWO   WORDS SERIALIZATION.
```

```
# FLAGWORDS 0-11        ARE DOWNLINKED AND CAN BE SET AND CLEARED BY UP-FLAG AND DOWN-FLAG INSTRUCTIONS IN THE
#                       INTERPRETER.  THESE WERE PREVIOUSLY LISTED UNDER "INTERPRETIVE SWITCH BIT ASSIGNMENTS" IN
#                       THE ERASABLE LOG SECTION.  FLAGWORDS 12 & 13 WERE PREVIOUSLY RADMODES AND DAPBOOLS AND
#                       ARE STILL DOWNLINKED UNDER THOSE NAMES.

#               ALPHABETICAL LIST OF FLAGWORDS

#
# FLAGWORD      DEC. NUMBER     BIT AND FLAG            BIT NAME

# ACCOKFLG      207             BIT  3 FLAG 13          ACCSOKAY
# ACC4-2FL      199             BIT 11 FLAG 13          ACC4OR2X
# ACMODFLG      032             BIT 13 FLAG  2          ACMODBIT
# ALTSCALE      186             BIT  9 FLAG 12          ALTSCBIT
# ANTENFLG      183             BIT 12 FLAG 12          ANTENBIT
# AORBSFLG      205             BIT  5 FLAG 13          AORBSYST
# AORBTFLG      200             BIT 10 FLAG 13          AORBTRAN
# APSESW        130             BIT  5 FLAG  8          APSESBIT
# APSFLAG       152             BIT 13 FLAG 10          APSFLBIT
# ASTNFLAG      108             BIT 12 FLAG  7          ASTNBIT
# ATTFLAG       104             BIT  1 FLAG  6          ATTFLBIT
# AUTOMODE      193             BIT  2 FLAG 12          AUTOMBIT
# AUTR1FLG      209             BIT  1 FLAG 13          AUTRATE1
# AUTR2FLG      208             BIT  2 FLAG 13          AUTRATE2
# AUXFLAG       103             BIT  2 FLAG  6          AUXFLBIT
# AVEGFLAG      115             BIT  5 FLAG  7          AVEGFBIT
# AVEMIDSW      149             BIT  1 FLAG  9          AVEMDBIT
# AVFLAG        040             BIT  5 FLAG  2          AVFLBIT
# CALCMAN2      043             BIT  2 FLAG  2          CALC2BIT
# CALCMAN3      042             BIT  3 FLAG  2          CALC3BIT
# CDESFLAG      180             BIT 15 FLAG 12          CDESBIT
# CMOONFLG      123             BIT 12 FLAG  8          CMOONBIT
# COGAFLAG      131             BIT  4 FLAG  8          COGAFBIT
# CSMDKFLG      197             BIT 13 FLAG 13          CSMDOCKD
# CULTFLAG      053             BIT  7 FLAG  3          CULTBIT
# DAPBOOLS                      FLGWRD13
# DBSELFLG      206             BIT  4 FLAG 13          DBSELECT
# DESIGFLG      185             BIT 10 FLAG 12          DESIGBIT
# DIDFLAG       016             BIT 14 FLAG             DIDFLBIT
# DIMOFLAG      059             BIT  1 FLAG  3          DIMOBIT
# DMENFLG       081             BIT  9 FLAG  5          DMENFBIT
# DRIFTDFL      202             BIT  8 FLAG 13          DRIFTBIT
# DRIFTFLG      030             BIT 15 FLAG  2          DRFTBIT
# DSKYFLAG      075             BIT 15 FLAG  5          DSKYFBIT
```

```
# D6OR9FLG      058             BIT  2 FLAG  3          D6OR9BIT
# ENGONFLG      083             BIT  7 FLAG  5          ENGONBIT
# ERADFLAG      017             BIT 13 FLAG  1          ERADFBIT
# ETPIFLAG      038             BIT  7 FLAG  2          ETPIBIT         EQUIVALENT FLAG NAME: DPTNSW
# FINALFLG      039             BIT  6 FLAG  2          FINALBIT
# FLAGWRD0      (000-014)       (STATE +0)
# FLAGWRD1      (015-029)       (STATE +1)
# FLAGWRD2      (030-044)       (STATE +2)
# FLAGWRD3      (045-059)       (STATE +3)
# FLAGWRD4      (060-074)       (STATE +4)
# FLAGWRD5      (075-089)       (STATE +5)
# FLAGWRD6      (090-104)       (STATE +6)
# FLAGWRD7      (105-119)       (STATE +7)
# FLAGWRD8      (120-134)       (STATE +8D)
# FLAGWRD9      (135-149)       (STATE +9D)
# FLAP          142             BIT  8 FLAG  9          FLAPBIT
# FLGWRD10      (150-164)       (STATE +10D)
# FLGWRD11      (165-179)       (STATE +11D)
# FLGWRD12      (180-194)       (STATE +12D)
# FLGWRD13      (195-209)       (STATE +13D)
# FLPC          138             BIT 12 FLAG  9          FLPCBIT
# FLPI          139             BIT 11 FLAG  9          FLPIBIT
# FLRCS         149             BIT 10 FLAG  9          FLRCSBIT
# FLUNDISP      125             BIT 10 FLAG  8          FLUNDBIT
# FLVR          136             BIT 14 FLAG  9          FLVRBIT
# FREEFLAG      012             BIT  3 FLAG  0          FREEFBIT
# FSPASFLG      005             BIT 10 FLAG  0          FSPASBIT
# GLOKFAIL      046             BIT 14 FLAG  3          GLOKFBIT
# GMBDRVSW      095             BIT 10 FLAG  6          GMBDRBIT
# GUESSW        028             BIT  2 FLAG  1          GUESSBIT
# HFLSHFLG      179             BIT  1 FLAG 11          HFLSHBIT
# IDLEFLAG      113             BIT  7 FLAG  7          IDLEFBIT
# IGNFLAG       107             BIT 13 FLAG  7          IGNFLBIT
# IMPULSW       036             BIT  9 FLAG  2          IMPULBIT
# IMUSE         007             BIT  8 FLAG  0          IMUSEBIT
# INFINFLG      128             BIT  7 FLAG  8          INFINBIT
# INITALGN      133             BIT  2 FLAG  8          INITABIT
# INTFLAG       151             BIT 14 FLAG 10          INTFLBIT
# INTYPFLG      056             BIT  4 FLAG  3          INTYPBIT
# ITSWICH       105             BIT 15 FLAG  7          ITSWBIT
# JSWITCH       001             BIT 14 FLAG  0          JSWCHBIT
# LETABORT      141             BIT  9 FLAG  9          LETABBIT
# LMOONFLG      124             BIT 11 FLAG  8          LMOONBIT
# LOKONSW       010             BIT  5 FLAG  0          LOKONBIT
# LOSCMFLG      033             BIT 12 FLAG  2          LOSCMBIT
# LRALTFLG      190             BIT  5 FLAG 12          LRALTBIT
# LRBYPASS      165             BIT 15 FLAG 11          LRBYBIT
# LRINH         172             BIT  8 FLAG 11          LRINHBIT
# LRPOSFLG      189             BIT  6 FLAG 12          LRPOSBIT
# LRVELFLG      187             BIT  8 FLAG 12          LRVELBIT
# PAGE63
# LUNAFLAG      048             BIT 12 FLAG  3          LUNABIT
# MANUFLAG      106             BIT 14 FLAG  7          MANUFBIT
# MGLVFLAG      088             BIT  2 FLAG  5          MGLVFBIT
# MIDAVFLG      148             BIT  2 FLAG  9          MIDAVBIT
# MIDFLAG       002             BIT 13 FLAG  0          MIDFLBIT
# MID1FLAG      147             BIT  3 FLAG  9          MID1BIT
# MKOVFLAG      072             BIT  3 FLAG  4          MKOVBIT
# MOONFLAG      003             BIT 12 FLAG  0          MOONBIT
```

```
# MRKIDFLG      060             BIT 15 FLAG  4          MRKIDBIT
# MRKNVFLG      066             BIT  9 FLAG  4          MRKNVBIT
# MRUPTFLG      070             BIT  5 FLAG  4          MRUPTBIT
# MUNFLAG       097             BIT  8 FLAG  6          MUNFLBIT
# MWAITFLG      064             BIT 11 FLAG  4          MWAITBIT
# NEEDLFLG      011             BIT  4 FLAG  0          NEEDLBIT
# NEWIFLG       122             BIT 13 FLAG  8          NEWIBIT
# NJETSFLG      015             BIT 15 FLAG             NJETSBIT
# NODOFLAG      044             BIT  1 FLAG  2          NODOBIT
# NOLRREAD      170             BIT 10 FLAG 11          NOLRRBIT
# NORMSW        110             BIT 10 FLAG  7          NORMSBIT
# NORRMON       086             BIT  4 FLAG  5          NORRMBIT
# NOR29FLG      049             BIT 11 FLAG  3          NR29FBIT
# NOTHROTL      078             BIT 12 FLAG  5          NOTHRBIT
# NOUPFLAG      024             BIT  6 FLAG  1          NOUPFBIT
# NRMNVFLG      067             BIT  8 FLAG  4          NRMNVBIT
# NRMIDFLG      062             BIT 13 FLAG  4          NRMIDBIT
# NRUPTFLG      071             BIT  4 FLAG  4          NRUPTBIT
# NTARGFLG      102             BIT  3 FLAG  6          NTARGBIT
# NWAITFLG      065             BIT 10 FLAG  4          NWAITBIT
# OLDESFLG      014             BIT  1 FLAG  0          OLDESBIT
# OPTNSW        038             BIT  7 FLAG  2          OPTNBIT         EQUIVALENT FLAG NAME: ETPIFLAG
# ORBWFLAG      054             BIT  6 FLAG  3          ORBWFBIT
# ORDERSW       129             BIT  6 FLAG  8          ORDERBIT
# OURRCFLG      198             BIT 12 FLAG 13          OURRCBIT
# PDSPFLAG      063             BIT 12 FLAG  4          PDSPFBIT
# PFRATFLG      041             BIT  4 FLAG  2          PFRATBIT
# PINBRFLG      069             BIT  6 FLAG  4          PINBRBIT
# PRECIFLG      052             BIT  8 FLAG  3          PRECIBIT
# PRIODFLG      061             BIT 14 FLAG  1          PRIODBIT
# PRONVFLG      068             BIT  7 FLAG  4          PRONVBIT
# PSTHIGAT      169             BIT 11 FLAG 11          PSTHIBIT
# PULSEFLG      195             BIT 15 FLAG 13          PULSES
# P21FLAG       004             BIT 11 FLAG  0          P21FLBIT
# P25FLAG       006             BIT  9 FLAG  0          P25FLBIT
# P39/79SW      126             BIT  9 FLAG  8          P39SWBIT
# QUITFLAG      145             BIT 5 FLAG 9            QUITBIT
# RADMODES                      FLGWRD12
# RASFLAG                       FLGWRD10
# RCDUFAIL      188             BIT  7 FLAG 12          RCDUFBIT
# RCDUOFLG      182             BIT 13 FLAG 12          RCDUOBIT
# READLR        174             BIT  6 FLAG 11          READLBIT
```

```
# READRFLG      051             BIT  9 FLAG  3          READRBIT        EQUIVALENT FLAG NAME FOR R04FLAG
# READVEL       175             BIT  5 FLAG 11          READVBIT
# REDFLAG       099             BIT  6 FLAG  6          REDFLBIT
# REFSMFLG      047             BIT 13 FLAG  3          REFSMBIT
# REINTFLG      158             BIT  7 FLAG 10          REINTBIT
# REMODFLG      181             BIT 14 FLAG 12          REMODBIT
# RENDWFLG      089             BIT  1 FLAG  5          RENDWBIT
# REPOSMON      184             BIT 11 FLAG 12          REPOSBIT
# RHCSCFLG      203             BIT  7 FLAG 13          RHCSCALE
# RNDVZFLG      008             BIT  7 FLAG  0          RNDVZBIT
# RNGEDATA      176             BIT  4 FLAG 11          RNGEDBIT
# RNGSCFLG      080             BIT 10 FLAG  5          RNGSCBIT
# RODFLAG       018             BIT 12 FLAG  1          RODFLBIT
# ROTFLAG       144             BIT  6 FLAG  9          ROTFLBIT
# RPQFLAG       120             BIT 15 FLAG  8          RPQFLBIT
# RRDATAFL      191             BIT  4 FLAG 12          RRDATABT
# RRNBSW        009             BIT  6 FLAG  0          RRNBBIT
# RRRSFLAG      192             BIT  3 FLAG 12          RRRSBIT
# RVSW          111             BIT  9 FLAG  7          RVSWBIT
# R04FLAG       051             BIT  9 FLAG  3          R04FLBIT        EQUIVALENT FLAG NAME:  READRFLG
# R10FLAG       013             BIT  2 FLAG  0          R10FLBIT
# R61FLAG       020             BIT 10 FLAG  1          R61FLBIT
# R77FLAG       079             BIT 11 FLAG  5          R77FLBIT
# SCALBAD       177             BIT  3 FLAG 11          SCABBIT
# SLOPESW       027             BIT  3 FLAG  1          SLOPEBIT
# SNUFFER       077             BIT 13 FLAG  5          SNUFFBIT
# SOLNSW        087             BIT  3 FLAG  5          SOLNSBIT
# SRCHOPTN      031             BIT 14 FLAG  2          SRCHOBIT
# STATEFLG      055             BIT  5 FLAG  3          STATEBIT
# STEERSW       034             BIT 11 FLAG  2          STEERBIT
# SURFFLAG      127             BIT  8 FLAG  8          SURFFBIT
# SWANDISP      109             BIT 11 FLAG  7          SWANDBIT
# S32.1F1       090             BIT 15 FLAG  6          S32BIT1
# S32.1F2       091             BIT 14 FLAG  6          S32BIT2
# S32.1F3A      092             BIT 13 FLAG  6          S32BIT3A
# S32.1F3B      093             BIT 12 FLAG  6          S32BIT3B
# TFFSW         119             BIT  1 FLAG  7          TFFSWBIT
# TRACKFLG      025             BIT  5 FLAG  1          TRACKBIT
# TURNONFL      194             BIT  1 FLAG 12          TURNONBT
# ULLAGFLG      204             BIT  6 FLAG 13          ULLAGER
# UPDATFLG      023             BIT  7 FLAG  1          UPDATBIT
# UPLOCKFL      116             BIT  4 FLAG 7           UPLOCBIT
# USEQRFLG      196             BIT 14 FLAG 13          USEQRJTS
# VEHUPFLG      022             BIT  8 FLAG  1          VEHUPBIT
# VELDATA       173             BIT  7 FLAG 11          VELDABIT
# VERIFLAG      117             BIT  3 FLAG  7          VERIFBIT
# VFLAG         050             BIT 10 FLAG  3          VFLAGBIT
# VFLSHFLG      178             BIT  2 FLAG 11          VFLSHBIT
# VINTFLAG      057             BIT  3 FLAG  3          VINTFBIT
# VXINH         168             BIT 12 FLAG 11          VXINHBIT
```

```
# V37FLAG        114             BIT  6 FLAG  7          V37FLBIT
# V67FLAG        112             BIT  8 FLAG  7          V67FLBIT
# V82EMFLG       118             BIT  2 FLAG  7          V82EMBIT
# XDELVFLG       037             BIT  8 FLAG  2          XDELVBIT
# XDSPFLAG       074             BIT  1 FLAG  4          XDSPBIT
# XORFLG         171             BIT  9 FLAG 11          XORFLBIT
# XOVINFLG       201             BIT  9 FLAG 13          XOVINHIB
# 3AXISFLG       084             BIT  6 FLAG  5          3AXISBIT
# 360SW          134             BIT  1 FLAG  8          360SWBIT

# ASSIGNMENT AND DESCRIPTION OF FLAGWORDS

FLAGWRD0         =       STATE +0                # (000-014)

                                                 #       (SET)                   (RESET)

# BIT 15 FLAG 0 (S)
                 =       000D
                 =       BIT15

# BIT 14 FLAG 0 (S)
JSWITCH          =       001D                    #       INTEGRATION OF W        INTEGRATION OF STATE
JSWCHBIT         =       BIT14                   #       MATRIX                  VECTOR

# BIT 13 FLAG 0 (S)
MIDFLAG          =       002D                    #       INTEGRATION WITH        INTEGRATION WITHOUT
                                                 #       SECONDARY BODY AND      SOLAR PERTURBATIONS
MIDFLBIT         =       BIT13                   #       SOLAR PERTURBATIONS

# BIT 12 FLAG 0 (L)
MOONFLAG         =       003D                    #       MOON IS SPHERE OF       EARTH IS SPHERE OF
MOONBIT          =       BIT12                   #       INFLUENCE               INFLUENCE

# BIT 11 FLAG 0
P21FLAG          =       004D                    #       USE BASE VECTORS        1ST PASS -- CALC-
P21FLBIT         =       BIT11                   #       ALREADY CALCULATED      ULATE BASE VECTORS

# BIT 10 FLAG 0
FSPASFLG         =       005D                    #       FIRST PASS THROUGH      NOT FIRST PASS THRU
FSPASBIT         =       BIT10                   #       REPOSITION ROUTINE      REPOSITION ROUTINE
```

```
# BIT 9 FLAG 0  (S)
P25FLAG         =       006D                    #       P25 OPERATING           P25 NOT OPERATING
P25FLBIT        =       BIT9

# BIT 8 FLAG 0  (S)
IMUSE           =       007D                    #       IMU IN USE              IMU NOT IN USE
IMUSEBIT        =       BIT8

# BIT 7 FLAG 0  (S)
RNDVZFLG        =       008D                    #       P20 RUNNING (RADAR      P20 NOT RUNNING
RNDVZBIT        =       BIT7                    #       IN USE)

# BIT 6 FLAG 0  (S)
RRNBSW          =       009D                    #       RADAR TARGET IN         RADAR TARGET IN
RRNBBIT         =       BIT6                    #       NB COORDINATES          SM COORDINATES

# BIT 5 FLAG 0  (S)
LOKONSW         =       010D                    #       RADAR LOCK-ON           RADAR LOCK-ON NOT
LOKONBIT        =       BIT5                    #       DESIRED                 DESIRED

# BIT 4 FLAG 0  (S)
NEEDLFLG        =       011D                    #       TOTAL ATTITUDE          A/P FOLLOWING
NEEDLBIT        =       BIT4                    #       ERROR DISPLAYED         ERROR DISPLAYED

# BIT 3 FLAG 0
FREEFLAG        =       012D                    # (USED BY P51-53 TEMP IN MANY DIFFERENT
                                                # ROUTINES & BY LUNAR + SOLAR EPHEMERIDES)
FREEFBIT        =       BIT3

# BIT 2 FLAG 0
R10FLAG         =       013D                    #       R10 OUTPUTS DATA TO     BESIDES OUTPUT WHEN
R10FLBIT        =       BIT2                    #       ALTITUDE & ALTITUDE     SET, R10 ALSO OUTPUT
                                                #       RATE METERS ONLY        TO FORWARD & LATERAL
                                                #                               VELOCITY CROSSPOINTER

# BIT 1 FLAG 0  (L)
OLDESFLG        =       014D                    #       R29 GYRO CMD LOOP       R29 GYRO CMD LOOP
OLDESBIT        =       BIT1                    #       REQUESTED               NOT REQUESTED


FLAGWRD1        =       STATE +1                # (015-029)
```

```
                                                 #       (SET)                   (RESET)

# BIT 15 FLAG 1 (S)
NJETSFLG        =       015D                     #       TWO JET RCS BURN        FOUR JET RCS BURN
NJETSBIT        =       BIT15

# BIT 14 FLAG 1 (L)
DIDFLAG         =       016D                     #       INERTIAL DATA IS        PERFORM DATA DISPLAY
DIDFLBIT        =       BIT14                    #       AVAILABLE               INITIALIZATION FUNCS

# BIT 13 FLAG 1 (S)
ERADFLAG        =       017D                     #       COMPUTE REARTH          USE CONSTANT REARTH
ERADFBIT        =       BIT13                    #       FISCHER ELLIPSOID       PAD RADIUS

# BIT 12 FLAG 1
RODFLAG         =       018D                     #       IF IN P66, NORMAL       IF IN P66, RE-INIT-
RODFLBIT        =       BIT12                    #       OPERATION CONTINUES.    IALIZATION IS PER-
                                                 #       RESTART CLEARS FLAG     FORMED AND FLAG IS

# BIT 11 FLAG 1
                =       019D
                =       BIT11

# BIT 10 FLAG 1 (L)
R61FLAG         =       020D                     #       RUN R61 LEM             RUN R65 LEM
R61FLBIT        =       BIT10

# BIT 9 FLAG 1
                =       021D
                =       BIT9

# BIT 8 FLAG 1  (S)
VEHUPFLG        =       022D                     #       CSM STATE-VECTOR        LEM STATE VECTOR
VEHUPBIT        =       BIT8                     #       BEING UPDATED           BEING UPDATED

# BIT 7 FLAG 1  (S)
UPDATFLG        =       023D                     #       UPDATING BY MARKS       UPDATING BY MARKS
UPDATBIT        =       BIT7                     #       ALLOWED                 NOT ALLOWED

# BIT 6 FLAG 1  (S)
NOUPFLAG        =       024D                     #       NEITHER CSM             EITHER STATE
                                                 #       NOR LM STATE VECTOR     VECTOR MAY BE
NOUPFBIT        =       BIT6                     #       MAY BE UPDATED          UPDATED
```

```
# BIT 5 FLAG 1  (S)
TRACKFLG        =       025D                    #       TRACKING ALLOWED        TRACKING NOT ALLOWED
TRACKBIT        =       BIT5

# BIT 4 FLAG 1
                =       026D
                =       BIT4

# BIT 3 FLAG 1  (S)
SLOPESW         =       027D                    #       ITERATE WITH BIAS       ITERATE WITH REGULAR
                                                #       METHOD IN ITERATOR      FALSI METHOD IN
SLOPEBIT        =       BIT3                    #                               ITERATOR

# BIT 2 FLAG 1  (S)
GUESSSW         =       028D                    #       NO STARTING VALUE       STARTING VALUE FOR
GUESSBIT        =       BIT2                    #       FOR ITERATION           ITERATION EXISTS

# BIT 1 FLAG 1
                =       029D
                =       BIT1                    # OH 2009-05-15 SCAN DOES NOT HAVE THIS LINE

FLAGWRD2        =       STATE +2                # (030-044)

                                                #       (SET)                   (RESET)

# BIT 15 FLAG 2 (S)
DRIFTFLG        =       030D                    #       T3RUPT CALLS GYRO       T3RUPT DOES NO GYRO
DRFTBIT         =       BIT15                   #       COMPENSATION            COMPENSATION

# BIT 14 FLAG 2 (S)
SRCHOPTN        =       031D                    #       RADAR IN AUTOMATIC      RADAR NOT IN AUTO-
SRCHOBIT        =       BIT14                   #       SEARCH OPTION (R24)     MATIC SEARCH OPTION

# BIT 13 FLAG 2 (S)
ACMODFLG        =       032D                    #       MANUAL ACQUISITION      AUTO ACQUISITION
ACMODBIT        =       BIT13                   #       BY RENDEZVOUS RADAR     BY RENDEZVOUS RADAR

# BIT 12 FLAG 2 (S)
LOSCMFLG        =       033D                    #       LINE OF SIGHT BEING     LINE OF SIGHT NOT
                                                #       COMPUTED (R21)          BEING COMPUTED
LOSCMBIT        =       BIT12
```

```
# BIT 11 FLAG 2 (S)
STEERSW         =       034D                    #       SUFFICIENT THRUST       INSUFFICIENT THRUST
STEERBIT        =       BIT11                   #       IS PRESENT              IS PRESENT

# BIT 10 FLAG 2 (S)
                =       035D                    # OH 2009-05-15 THESE TWO LINE DON'T APPEAR IN SCAN
                =       BIT10

# BIT 9 FLAG 2 (S)
IMPULSW         =       036D                    #       MINIMUM IMPULSE         STEERING BURN (NO
                                                #       BURN (CUTOFF TIME       CUTOFF TIME YET
IMPULBIT        =       BIT9                    #       SPECIFIED)              AVAILABLE)

# BIT 8 FLAG 2 (S)
XDELVFLG        =       037D                    #       EXTERNAL DELTAV VG      LAMBERT (AIMPOINT)
XDELVBIT        =       BIT8                    #       COMPUTATION             VG COMPUTATION

# BIT 7 FLAG 2 (S)
ETPIFLAG        =       038D                    #       ELEVATION ANGLE         TPI TIME SUPPLIED
                                                #       SUPPLIED FOR            FOR P34,74 TO COMPUTE
ETPIBIT         =       BIT7                    #       P34,74                  ELEVATION

# BIT 7 FLAG 2 (L)
OPTNSW          =       ETPIFLAG                #       SOI PHASE OF P38/78     SOR PHASE OF P38/78
OPTNBIT         =       BIT7

# BIT 6 FLAG 2 (S)
FINALFLG        =       039D                    #       LAST PASS THROUGH       INTERIM PASS THROUGH
                                                #       RENDEZVOUS PROGRAM      RENDEZVOUS PROGRAM
FINALBIT        =       BIT6                    #       COMPUTATIONS            COMPUTATIONS

# BIT 5 FLAG 2 (S)
AVFLAG          =       040D                    #       LEM IS ACTIVE           CSM IS ACTIVE
AVFLBIT         =       BIT5                    #       VEHICLE                 VEHICLE

# BIT 4 FLAG 2 (S)
PFRATFLG        =       041D                    #       PREFERRED ATTITUDE      PREFERRED ATTITUDE
PFRATBIT        =       BIT4                    #       COMPUTED                NOT COMPUTED

# BIT 3 FLAG 2 (S)
```

```
CALCMAN3        =       042D                    #       NO FINAL ROLL           FINAL ROLL IS
CALC3BIT        =       BIT3                    #                               NECESSARY

# BIT 2 FLAG 2 (S)
CALCMAN2        =       043D                    #       PERFORM MANEUVER        BYPASS STARTING
CALC2BIT        =       BIT2                    #       STARTING PROCEDURE      PROCEDURE

# BIT 1 FLAG 2 (S)
NODOFLAG        =       044D                    #       V37 NOT PERMITTED       V37 PERMITTED
NODOBIT         =       BIT1

FLAGWRD3        =       STATE +3                # (045-059)

                                                #       (SET)                   (RESET)


# BIT 15 FLAG 3
                =       045D                    #
                =       BIT15                   # OH 2009-05-15 THIS LINE IS NOT IN SCANS

# BIT 14 FLAG 3 (S)
GLOKFAIL        =       046D                    #       GIMBAL LOCK HAS         NOT IN GIMBAL LOCK
GLOKFBIT        =       BIT14                   #       OCCURRED

# BIT 13 FLAG 3 *** PROTECTED FROM FRESH START ***
REFSMFLG        =       047D                    #       REFSMMAT GOOD           REFSMMAT NO GOOD
REFSMBIT        =       BIT13

# BIT 12 FLAG 3 (S)
LUNAFLAG        =       048D                    #       LUNAR LAT-LONG          EARTH LAT-LONG
LUNABIT         =       BIT12

# BIT 11 FLAG 3 (L)
NOR29FLG        =       049D                    #       R29 NOT ALLOWED         R29 ALLOWED (RR DES-
NR29FBIT        =       BIT11                   #                               IGNATED POWERED FLT)

# BIT 10 FLAG 3 (S)
VFLAG           =       050D                    #       LESS THAN TWO STARS     TWO STARS IN FIELD
VFLAGBIT        =       BIT10                   #       IN FIELD OF VIEW        OF VIEW

# BIT 9 FLAG 3  (S)
RO4FLAG         =       051D                    #       ALARM 521               ALARM 521 ALLOWED
                                                #       SUPPRESSED
```

```
R04FLBIT        =       BIT9

# BIT 9 FLAG 3  (L)
READRFLG        =       R04FLAG                 #       READING RR DATA         NOT READING RR DATA
READRBIT        =       BIT9                    #       PURSUANT TO R29         PURSUANT TO R29

# BIT 8 FLAG 3  (S)
PRECIFLG        =       052D                    #       NORMAL INTEGRATION      ENGAGES 4-TIME STEP
                                                #       IN P00                  (P00) LOGIC IN INTE-
PRECIBIT        =       BIT8                    #                               GRATION

# BIT 7 FLAG 3  (S)
CULTFLAG        =       053D                    #       STAR OCCULTED           STAR NOT OCCULTED
CULTBIT         =       BIT7

# BIT 6 FLAG 3  (S)
ORBWFLAG        =       054D                    #       W MATRIX VALID FOR      W MATRIX INVALID FOR
ORBWFBIT        =       BIT6                    #       ORBITAL NAVIGATION      ORBITAL NAVIGATION

# BIT 5 FLAG 3  (S)
STATEFLG        =       055D                    #       PERMANENT STATE         PERMANENT STATE
STATEBIT        =       BIT5                    #       VECTOR UPDATED          VECTOR NOT UPDATED

# BIT 4 FLAG 3  (S)
INTYPFLG        =       056D                    #       CONIC INTEGRATION       ENCKE INTEGRATION
INTYPBIT        =       BIT4

# BIT 3 FLAG 3  (S)
VINTFLAG        =       057D                    #       CSM STATE VECTOR        LEM STATE VECTOR
VINTFBIT        =       BIT3                    #       BEING INTEGRATED        BEING INTEGRATED

# BIT 2 FLAG 3 (S)
D6OR9FLG        =       058D                    #       DIMENSION OF W IS 9     DIMENSION OF W IS 6
D6OR9BIT        =       BIT2                    #       FOR INTEGRATION         FOR INTEGRATION

# BIT 1 FLAG 3  (S)
DIMOFLAG        =       059D                    #       W MATRIX IS TO BE       W MATRIX IS NOT TO
DIMOBIT         =       BIT1                    #       USED                    USED

FLAGWRD4        =       STATE +4                # (060-074)
```

```
                                                        #       (SET)                   (RESET)

        # BIT 15 FLAG 4 (S)
        MRKIDFLG        =       060D                    #       MARK DISPLAY IN         NO MARK DISPLAY IN
        MRKIDBIT        =       BIT15                   #       ENDIDLE                 ENDIDLE

        # BIT 14 FLAG 4 (S)
        PRIODFLG        =       061D                    #       PRIORITY DISPLAY IN     NO PRIORITY DISPLAY
        PRIODBIT        =       BIT14                   #       ENDIDLE                 IN ENDIDLE

        # BIT 13 FLAG 4 (S)
        NRMIDFLG        =       062D                    #       NORMAL DISPLAY IN       NO NORMAL DISPLAY
        NRMIDBIT        =       BIT13                   #       ENDIDLE                 IN ENDIDLE

        # BIT 12 FLAG 4 (S)
        PDSPFLAG        =       063D                    #       P20 SETS SO AS TO       LEAVE AS NORMAL DISP
                                                        #       TURN A NORMAL DIS-
        PDSPFBIT        =       BIT12                   #       PLAY INTO A PRIORITY
                                                        #       DISPLAY IN R60

        # BIT 11 FLAG 4 (S)
        MWAITFLG        =       064D                    #       HIGHER PRIORITY         NO HIGHER PRIORITY
                                                        #       DISPLAY OPERATING       DISPLAY OPERATING
        MWAITBIT        =       BIT11                   #       WHEN MARK               WHEN MARK DISPLAY
                                                        #       DISPLAY INITIATED       INITIATED

        # BIT 10 FLAG 4 (S)
        NWAITFLG        =       065D                    #       HIGHER PRIORITY         NO HIGHER PRIORITY
                                                        #       DISPLAY OPERATING       DISPLAY OPERATING
        NWAITBIT        =       BIT10                   #       WHEN NORMAL             WHEN NORMAL DISPLAY
                                                        #       DISPLAY INITIATED       INITIATED

        # BIT 9 FLAG 4  (S)
        MRKNVFLG        =       066D                    #       ASTRONAUT USING         ASTRONAUT NOT USING
                                                        #       KEYBOARD WHEN MARK      KEYBOARD WHEN MARK
        MRKNVBIT        =       BIT9                    #       DISPLAY INITIATED       DISPLAY INITIATED

        # BIT 8 FLAG 4  (S)
        NRMNVFLG        =       067D                    #       ASTRONAUT USING         ASTRONAUT NOT USING
                                                        #       KEYBOARD WHEN           KEYBOARD WHEN
        NRMNVBIT        =       BIT8                    #       NORMAL DISPLAY          NORMAL DISPLAY
                                                        #       INITIATED               INITIATED

        # BIT 7 FLAG 4  (S)
        PRONVFLG        =       068D                    #       ASTRONAUT USING         ASTRONAUT NOT USING
```

```
                                                            #       KEYBOARD WHEN           KEYBOARD WHEN
PRONVBIT        =       BIT7                                #       PRIORITY DISPLAY        PRIORITY DISPLAY
                                                            #       INITIATED               INITIATED

# BIT 6 FLAG 4  (S)
PINBRFLG        =       069D                                #       ASTRONAUT HAS           ASTRONAUT HAS NOT
                                                            #       INTERFERED WITH         INTERFERED WITH
PINBRBIT        =       BIT6                                #       EXISTING DISPLAY        EXISTING DISPLAY

# BIT 5 FLAG 4  (S)
MRUPTFLG        =       070D                                #       MARK DISPLAY            MARK DISPLAY NOT
                                                            #       INTERRUPTED BY          INTERRUPTED BY
MRUPTBIT        =       BIT5                                #       PRIORITY DISPLAY        PRIORITY DISPLAY

# BIT 4 FLAG 4  (S)
NRUPTFLG        =       071D                                #       NORMAL DISPLAY          NORMAL DISPLAY NOT
                                                            #       INTERRUPTED BY          INTERRUPTED BY
NRUPTBIT        =       BIT4                                #       PRIORITY OR MARK        PRIORITY OR MARK
                                                            #       DISPLAY                 DISPLAY

# BIT 3 FLAG 4  (S)
MKOVFLAG        =       072D                                #       MARK DISPLAY OVER       NO MARK DISPLAY OVER
MKOVBIT         =       BIT3                                #       NORMAL                  NORMAL

# BIT 2 FLAG 4
                =       073D
                =       BIT2                                # OH 2009-05-15 NOT IN SCAN.

# BIT 1 FLAG 4  (S)
XDSPFLAG        =       074D                                #       MARK DISPLAY NOT        NO SPECIAL MARK
XDSPBIT         =       BIT1                                #       TO BE INTERRUPTED       INFORMATION

FLAGWRD5        =       STATE +5                            # (075-089)

                                                            #       (SET)                   (RESET)

# BIT 15 FLAG 5 (S)
DSKYFLAG        =       075D                                #       DISPLAYS SENT TO        NO DISPLAYS TO DSKY
DSKYFBIT        =       BIT15                               #       DSKY

# BIT 14 FLAG 5
                =       076D
                =       BIT14
```

```
# BIT 13 FLAG 5 (S,L)
SNUFFER         =       077D                    #       U,V JETS DISABLED       U,V JETS ENABLED
                                                #       DURING DPS              DURING DPS
SNUFFBIT        =       BIT13                   #       BURNS (V65)             BURNS (V75)

# BIT 12 FLAG 5 (S)
NOTHROTL        =       078D                    #       INHIBIT FULL            PERMIT FULL THROTTLE
NOTHRBIT        =       BIT12                   #       THROTTLE

# BIT 11 FLAG 5 (S,L)
R77FLAG         =       079D                    #       R77 IS ON,              R77 IS NOT ON.
                                                #       SUPPRESS ALL RADAR
                                                #       ALARMS AND TRACKER
R77FLBIT        =       BIT11                   #       FAILS

# BIT 10 FLAG 5 (S)
RNGSCFLG        =       080D                    #       SCALE CHANGE HAS        NO SCALE CHANGE HAS
                                                #       OCCURRED DURING         OCCURRED DURING
RNGSCBIT        =       BIT10                   #       RR READING              RR READING

# BIT 9 FLAG 5  (S)
DMENFLG         =       081D                    #       DIMENSION OF W IS 9     DIMENSION OF W IS 6
DMENFBIT        =       BIT9                    #       FOR INCORPORATION       FOR INCORPORATION

# BIT 8 FLAG 5  (S)
                =       082D
                =       BIT8

# BIT 7 FLAG 5  (S)
ENGONFLG        =       083D                    #       ENGINE TURNED ON        ENGINE TURNED OFF
ENGONBIT        =       BIT7                    #

# BIT 6 FLAG 5  (S)
3AXISFLG        =       084D                    #       MANEUVER SPECIFIED      MANEUVER SPECIFIED
                                                #       BY THREE AXES           BY ONE AXIS; R60
3AXISBIT        =       BIT6                    #                               CALLS VECPOINT.

# BIT 5 FLAG 5
                =       085D
                =       BIT5                    # OH 2009-05-15 NOT IN SCAN

# BIT 4 FLAG 5  (S)
```

```
NORRMON          =       086D                  #       BYPASS RR GIMBAL        PERFORM
NORRMBIT         =       BIT4                  #       MONITOR                 RR GIMBAL MONITOR

# BIT 3 FLAG 5  (S)
SOLNSW           =       087D                  #       LAMBERT DOES NOT        LAMBERT CONVERGES OR
                                               #       CONVERGE, OR TIME-RAD   TIME-RADIUS NON-
SOLNSBIT         =       BIT3                  #       NEARLY CIRCULAR         CIRCULAR

# BIT 2 FLAG 5  (S)
MGLVFLAG         =       088D                  #       LOCAL VERTICAL          MIDDLE GIMBAL ANGLE
                                               #       COORDINATES             COMPUTED
MGLVFBIT         =       BIT2                  #       COMPUTED

# BIT 1 FLAG 5  (S)
RENDWFLG         =       089D                  #       W MATRIX VALID          W MATRIX INVALID
                                               #       FOR RENDEZVOUS          FOR RENDEZVOUS
RENDWBIT         =       BIT1                  #       NAVIGATION              NAVIGATION


FLAGWRD6         =       STATE +6              # (090-104)

                                               #       (SET)                   (RESET)

# BIT 15 FLAG 6 (S)
S32.1F1          =       090D                  #       DELTA V AT CSI TIME     DVT1 LESS THAN MAX
S32BIT1          =       BIT15                 #       ONE EXCEEDS MAX

# BIT 14 FLAG 6 (S)
S32.1F2          =       091D                  #       FIRST PASS OF           REITERATION OF
S32BIT2          =       BIT14                 #       NEWTON ITERATION        NEWTON

# BIT 13 FLAG 6 (S)
S32.1F3A         =       092D                  # BIT 13 AND BIT 12 FUNCTION AS AN ORDERED
S32BIT3A         =       BIT13                 # PAIR (13,12) INDICATING THE POSSIBLE OC-
                                               # CURRENCE OF 2 NEWTON ITERATIONS FOR S32.1
                                               # IN THE PROGRAM IN THE FOLLOWING ORDER:
# BIT 12 FLAG 6 (S)                            # (0,1) (I.E. BIT 13 RESET, BIT 12 SET)
S32.1F3B         =       093D                  #      = FIRST NEWTON ITERATION BEING DONE
S32BIT3B         =       BIT12                 # (0,0)= FIRST PASS OF SECOND NEWTON ITERATION
                                               # (1,1)= 50 FT/SEC STAGE OF SECOND NEWTON ITERATION
                                               # (1,0)= REMAINDER OF SECOND NEWTON ITERATION
# BIT 11 FLAG 6 (S)
                 =       094D                  #
                 =       BIT11                 #
```

```
     # BIT 10 FLAG 6 (S)
     GMBDRVSW        =       095D                    #       TRIMGIMB OVER           TRIMGIMB NOT OVER
     GMBDRBIT        =       BIT10                   #

     # BIT 9 FLAG 6
                     =       096D                    #
                     =       BIT9                    #

     # BIT 8 FLAG 6  (S)
     MUNFLAG         =       097D                    #       SERVICER CALLS          SERVICER CALLS
     MUNFLBIT        =       BIT8                    #       MUNRVG                  CALCRVG

     # BIT 7 FLAG 6  (L)
                     =       098D                    #
                     =       BIT7                    #

     # BIT 6 FLAG 6  (L)
     REDFLAG         =       099D                    #       LANDING SITE            LANDING SITE
                                                     #       REDESIGNATION           REDESIGNATION NOT
     REDFLBIT        =       BIT6                    #       PERMITTED               PERMITTED

     # BIT 5 FLAG 6
                     =       100D                    #
                     =       BIT5                    # OH 2009-05-15 NOT IN SCAN

     # BIT 4 FLAG 6
                     =       101D                    #
                     =       BIT4                    # OH 2009-05-15 NOT IN SCAN

     # BIT 3 FLAG 6  (S)
     NTARGFLG        =       102D                    #       ASTRONAUT DID           ASTRONAUT DID NOT
                                                     #       OVERWRITE DELTA         OVERWRITE DELTA
     NTARGBIT        =       BIT3                    #       VELOCITY AT TPI         VELOCITY
                                                     #       OR TPM (P34,35)

     # BIT 2 FLAG 6
     AUXFLAG         =       103D                    #       PROVIDING IDLEFLAG      SERVICER WILL SKIP
     AUXFLBIT        =       BIT2                    #       IS NOT SET, SERV-       DVMON ON ITS NEXT
                                                     #       ICER WILL EXERCISE      PASS EVEN IF THE
                                                     #       DVMON ON ITS NEXT       IDLEFLAG IS NOT SET.
                                                     #       PASS.                   IT WILL THEN SET
                                                     #                               AUXFLAG.

     # BIT 1 FLAG 6  (L)
     ATTFLAG         =       104D                    #       LEM ATTITUDE EXISTS     NO LEM ATTITUDE
                                                     #       IN MOON-FIXED           AVAILABLE IN MOON-
```

```
ATTFLBIT        =       BIT1                    #       COORDINATES             FIXED COORDINATES

FLAGWRD7        =       STATE +7                # (105-119)

                                                #       (SET)                   (RESET)

# BIT 15 FLAG 7 (S)
ITSWICH         =       105D                    #       R34;TPI TIME TO BE      TPI HAS BEEN
ITSWBIT         =       BIT15                   #       COMPUTED                COMPUTED

# BIT 14 FLAG 7 (S)
MANUFLAG        =       106D                    #       ATTITUDE MANEUVER       NO ATTITUDE MANEUVER
                                                #       GOING DURING RR         DURING RR SEARCH
MANUFBIT        =       BIT14                   #       SEARCH

# BIT 13 FLAG 7 (S)
IGNFLAG         =       107D                    #       TIG HAS ARRIVED         TIG HAS NOT ARRIVED
IGNFLBIT        =       BIT13                   #

# BIT 12 FLAG 7 (S)
ASTNFLAG        =       108D                    #       ASTRONAUT HAS           ASTRONAUT HAS NOT
ASTNBIT         =       BIT12                   #       OKAYED IGNITION         OKAYED IGNITION

# BIT 11 FLAG 7 (L)
SWANDISP        =       109D                    #       LANDING ANALOG          LANDING ANALOG
SWANDBIT        =       BIT11                   #       DISPLAYS ENABLED        DISPLAYS SUPPRESSED

# BIT 10 FLAG 7 (S)
NORMSW          =       110D                    #       UNIT NORMAL INPUT       LAMBERT COMPUTES ITS
NORMSBIT        =       BIT10                   #       TO LAMBERT              OWN UNIT NORMAL

# BIT 9 FLAG 7  (S)
RVSW            =       111D                    #       DO NOT COMPUTE          COMPUTE FINAL STATE
                                                #       FINAL STATE VECTOR      VECTOR IN TIME-THETA
RVSWBIT         =       BIT9                    #       IN TIME-DELTA

# BIT 8 FLAG 7  (S)
V67FLAG         =       112D                    #       ASTRONAUT OVERWRITE     ASTRONAUT DOES NOT
                                                #       W-MATRIX INITIAL        OVERWRITE W-MATRIX
V67FLBIT        =       BIT8                    #       VALUES                  INITIAL VALUES
```

```
# BIT 7 FLAG 7  (S)
IDLEFLAG        =       113D                    #       NO DV MONITOR           CONNECT DV MONITOR
IDLEFBIT        =       BIT7                    #

# BIT 6 FLAG 7  (S)
V37FLAG         =       114D                    #       AVERAGEG (SERVICER)     AVERAGEG (SERVICER)
V37FLBIT        =       BIT6                    #       RUNNING                 OFF

# BIT 5 FLAG 7  (S)
AVEGFLAG        =       115D                    #       AVERAGEG (SERVICER)     AVERAGEG (SERVICER)
AVEGFBIT        =       BIT5                    #       DESIRED                 NOT DESIRED

# BIT 4 FLAG 7  (S)
UPLOCKFL        =       116D                    #       K-KBAR-K FAIL           NO K-KBAR-K FAIL
UPLOCBIT        =       BIT4                    #

# BIT 3 FLAG 7  (S)
VERIFLAG        =       117D                    # CHANGED WHEN V33E OCCURS AT END OF P27
VERIFBIT        =       BIT3                    #

# BIT 2 FLAG 7  (L,C)
V82EMFLG        =       118D                    #       MOON VICINITY           EARTH VICINITY
V82EMBIT        =       BIT2                    #

# BIT 1 FLAG 7  (S)
TFFSW           =       119D                    #       CALCULATE TPERIGEE      CALCULATE TFF
TFFSWBIT        =       BIT1                    #


FLAGWRD8        =       STATE +8D               # (120-134)

                                                #       (SET)                   (RESET)


# BIT 15 FLAG 8 (S)
RPQFLAG         =       120D                    #       RPQ NOT COMPUTED        RPQ COMPUTED
                                                #       (RPQ = VECTOR BE-
RPQFLBIT        =       BIT15                   #       TWEEN SECONDARY BODY
                                                #       AND PRIMARY BODY)

# BIT 14 FLAG 8
                =       121D                    #
                =       BIT14                   #
```

```
# BIT 13 FLAG 8 (S)
NEWIFLG         =       122D                    #       FIRST PASS THROUGH      SUCCEEDING ITERATION
NEWIBIT         =       BIT13                   #       INTEGRATION             OF INTEGRATION

# BIT 12 FLAG 8 *** PROTECTED FROM FRESH START ***
CMOONFLG        =       123D                    #       PERMANENT CSM STATE     PERMANENT CSM STATE
CMOONBIT        =       BIT12                   #       IN LUNAR SPHERE         IN EARTH SPHERE

# BIT 11 FLAG 8 *** PROTECTED FROM FRESH START ***
LMOONFLG        =       124D                    #       PERMANENT LM STATE      PERMANENT LM STATE
LMOONBIT        =       BIT11                   #       IN LUNAR SPHERE         IN EARTH SPHERE

# BIT 10 FLAG 8 (L)
FLUNDISP        =       125D                    #       CURRENT GUIDANCE        CURRENT GUIDANCE
FLUNDBIT        =       BIT10                   #       DISPLAYS INHIBITED      DISPLAYS PERMITTED

# BIT 9 FLAG 8  (L)
P39/79SW        =       126D                    #       P39/79 OPERATING        P38/78 OPERATING
P39SWBIT        =       BIT9                    #

# BIT 8 FLAG 8  *** PROTECTED FROM FRESH START ***
SURFFLAG        =       127D                    #       LM ON LUNAR SURFACE     LM NOT ON LUNAR
SURFFBIT        =       BIT8                    #                               SURFACE

# BIT 7 FLAG 8  (S)
INFINFLG        =       128D                    #       NO CONIC SOLUTION       CONIC SOLUTION
                                                #       (CLOSURE THROUGH        EXISTS
INFINBIT        =       BIT7                    #       INFINITY REQUIRED)

# BIT 6 FLAG 8  (S)
ORDERSW         =       129D                    #       ITERATOR USES 2ND       ITERATOR USES 1ST
ORDERBIT        =       BIT6                    #       ORDER MINIMUM MODE      ORDER STANDARD MODE

# BIT 5 FLAG 8  (S)
APSESW          =       130D                    #       RDESIRED OUTSIDE        RDESIRED INSIDE
                                                #       PERICENTER-APOCENTER    PERICENTER-APOCENTER
APSESBIT        =       BIT5                    #       RANGE IN TIME-RADIUS    RANGE IN TIME-RADIUS

# BIT 4 FLAG 8  (S)
COGAFLAG        =       131D                    #       NO CONIC SOLUTION --    CONIC SOLUTION
                                                #       TOO CLOSE TO RECTI-     EXISTS (COGA DOES NOT
```

```
COGAFBIT        =       BIT4                    #       LINEAR (COGA OVERFLWS)  OVERFLOW)

# BIT 3 FLAG 8  (S)
                =       132D                    #
                =       BIT3                    # OH 2009-05-15 LINE NOT IN SCAN

# BIT 2 FLAG 8  (L)
INITALGN        =       133D                    #       INITIAL PASS THRU       SECOND PASS THRU P57
INITABIT        =       BIT2                    #       P57                     (CHECK RESET-MILLARD)

# BIT 1 FLAG 8  (S)
360SW           =       134D                    #       TRANSFER ANGLE NEAR     TRANSFER ANGLE NOT
360SWBIT        =       BIT1                    #       360 DEGREES             NEAR 360 DEGREES

FLAGWRD9        =       STATE +9D               # (135-149)

                                                #       (SET)                   (RESET)

# BIT 15 FLAG 9
                =       135D                    #
                =       BIT15                   #

# BIT 14 FLAG 9 (L)
FLVR            =       136D                    #       VERTICAL RISE           NON-VERTICAL RISE
FLVRBIT         =       BIT14                   #       (ASCENT GUIDANCE)

# BIT 13 FLAG 9
                =       137D                    #
                =       BIT13                   # OH 2009-05-15 LINE NOT IN SCAN

# BIT 12 FLAG 9 (L)
FLPC            =       138D                    #       NO POSITION CONTROL     POSITION CONTROL
FLPCBIT         =       BIT12                   #       (ASCENT GUIDANCE)

# BIT 11 FLAG 9 (L)
FLPI            =       139D                    #       PRE-IGNITION PHASE      REGULAR GUIDANCE
FLPIBIT         =       BIT11                   #       (ASCENT GUIDANCE)

# BIT 10 FLAG 9 (L)
FLRCS           =       140D                    #       RCS INJECTION MODE      MAIN ENGINE MODE
FLRCSBIT        =       BIT10                   #       (ASCENT GUIDANCE)

# BIT 9 FLAG 9  (L)
```

```
LETABORT        =       141D                    #       ABORT PROGRAMS          ABORT PROGRAMS
LETABBIT        =       BIT9                    #       ARE ENABLED             ARE NOT ENABLED

# BIT 8 FLAG 9  (L)
FLAP            =       142D                    #       APS CONTINUED ABORT     APS ABORT IS NOT A
                                                #       AFTER DPS STAGING       CONTINUATION
FLAPBIT         =       BIT8                    #       (ASCENT GUIDANCE)

# BIT 7 FLAG 9  (L)
                =       143D
                =       BIT7                    # OH 2009-05-15 LINE NOT IN SCAN


# BIT 6 FLAG 9  (L)
ROTFLAG         =       144D                    #       P70 AND P71 WILL        P70 AND P71 WILL NOT
ROTFLBIT        =       BIT6                    #       FORCE VEHICLE           FORCE VEHICLE
                                                #       ROTATION IN THE         ROTATION IN THE
                                                #       PREFERRED DIRECTION     PREFERRED DIRECTION

# BIT 5 FLAG 9  (S)
QUITFLAG        =       145D                    #       DISCONTINUE INTEGR.     CONTINUE INTEGRATION
QUITBIT         =       BIT5                    #

# BIT 4 FLAG 9
                =       146D                    #
                =       BIT4                    #

# BIT 3 FLAG 9  (L)
MID1FLAG        =       147D                    #       INTEGRAT TO TDEC        INTEGRATE TO THE
MID1FBIT        =       BIT3                    #                               THEN-PRESENT TIME

# BIT 2 FLAG 9  (L)
MIDAVFLG        =       148D                    #       INTEGRATION ENTERED     INTEGRATION WAS
                                                #       FROM ONE OF MIDTOAV     NOT ENTERED VIA
MIDAVBIT        =       BIT2                    #       PORTALS                 MIDTOAV

# BIT 1 FLAG 9  (S)
AVEMIDSW        =       149D                    #       AVETOMID CALLING        NO AVETOMID W INTEGR
                                                #       FOR W.MATRIX INTEGR     ALLOW SET UP RM, VN
AVEMDBIT        =       BIT1                    #       DON'T WRITE OVER RN,    PIPTIME
                                                #       VN,PIPTIME


RASFLAG         EQUALS  FLGWRD10                # WAS ONLY AN INSTALL-ERASTALL FLAG
```

```
FLGWRD10        =       STATE +10D              # (150-164)

                                                #       (SET)                   (RESET)

# BIT 15 FLAG 10 (S)
                =       150D                    #
                =       BIT15                   # OH 2009-05-15 LINE NOT IN SCAN
# BIT 14 FLAG 10 (L,C)
INTFLAG         =       151D                    #       INTEGRATION IN          INTEGRATION NOT IN
INTFLBIT        =       BIT14                   #       PROGRESS                PROGRESS

# BIT 13 FLAG 10 (S,L)
APSFLAG         =       152D                    #       ASCENT STAGE            DESCENT STAGE
APSFLBIT        =       BIT13                   #        *** PROTECTED FROM FRESH START ***

# BIT 12 FLAG 10
                =       153D                    #
                =       BIT12                   # OH 2009-05-15 LINE NOT IN SCAN

# BIT 11 FLAG 10
                =       154D                    #
                =       BIT11                   # OH 2009-05-15 LINE NOT IN SCAN

# BIT 10 FLAG 10
                =       155D                    #
                =       BIT10                   # OH 2009-05-15 LINE NOT IN SCAN

# BIT 9 FLAG 10
                =       156D                    #
                =       BIT9                    # OH 2009-05-15 LINE NOT IN SCAN

# BIT 8 FLAG 10
                =       157D                    #
                =       BIT8                    # OH 2009-05-15 LINE NOT IN SCAN

# BIT 7 FLAG 10 (L,C)
REINTFLG        =       158D                    #       INTEGRATION ROUTINE     INTEGRATION ROUTINE
REINTBIT        =       BIT7                    #       TO BE RESTARTED         NOT TO BE RESTARTED

# BIT 6 FLAG 10
                =       159D                    #
                =       BIT6                    # OH 2009-05-15 LINE NOT IN SCAN

# BIT 5 FLAG 10
                =       160D                    #
                =       BIT5                    # OH 2009-05-15 LINE NOT IN SCAN
```

```
# BIT 4 FLAG 10
                =       161D                    #
                =       BIT4                    # OH 2009-05-15 LINE NOT IN SCAN

# BIT 3 FLAG 10
                =       162D                    #
                =       BIT3                    # OH 2009-05-15 LINE NOT IN SCAN

# BIT 2 FLAG 10
                =       163D                    #
                =       BIT2                    # OH 2009-05-15 LINE NOT IN SCAN

# BIT 1 FLAG 10
                =       164D                    #
                =       BIT1                    # OH 2009-05-15 LINE NOT IN SCAN


FLGWRD11        =       STATE +11D              # (165-179)

                                                #       (SET)                   (RESET)

# BIT 15 FLAG 11 (L)(R12)
LRBYPASS        =       165D                    #       BYPASS ALL LANDING      DO NOT BYPASS LR
LRBYBIT         =       BIT15                   #       RADAR UPDATES           UPDATES

# BIT 14 FLAG 11
                =       166D                    #
                =       BIT14                   #

# BIT 13 FLAG 11
                =       167D                    #
                =       BIT13                   #

# BIT 12 FLAG 11 (L)(R12)
VXINH           =       168D                    #       IF Z VELOCITY DATA      UPDATE X AXIS
                                                #       UNREASONABLE,           VELOCITY
VXINHBIT        =       BIT12                   #       BYPASS X VELOCITY
                                                #       UPDATE ON NEXT PASS

# BIT 11 FLAG 11 (L)(R12)
PSTHIGAT        =       169D                    #       PAST HIGATE             PREHIGATE
PSTHIBIT        =       BIT11                   #

# BIT 10 FLAG 11 (L)(R12)
```

```
NOLRREAD        =       170D                    #       LANDING RADAR           LR NOT REPOSITIONING
                                                #       REPOSITIONING;
NOLRRBIT        =       BIT10                   #       BYPASS UPDATE

# BIT 9 FLAG 11 (L)(R12)
XORFLG          =       171D                    #       BELOW LIMIT             ABOVE LIMIT DO
                                                #       INHIBIT X AXIS          NOT INHIBIT
XORFLBIT        =       BIT9                    #       OVERRIDE

# BIT 8 FLAG 11
LRINH           =       172D                    #       LANDING RADAR UP-       LR UPDATES INHIBITED
LRINHBIT        =       BIT8                    #       DATES PERMITTED         BY ASTRONAUT
                                                #       BY ASTRONAUT

# BIT 7 FLAG 11 (L)(R12)
VELDATA         =       173D                    #       LR VELOCITY             LR VELOCITY MEASURE
VELDABIT        =       BIT7                    #       MEASUREMENT MADE        NOT MADE

# BIT 6 FLAG 11 (L)(R12)
READLR          =       174D                    #       OK TO READ LR           DO NOT READ LR RANGE
READLBIT        =       BIT6                    #       RANGE DATA              DATA

# BIT 5 FLAG 11 (L)(R12)
READVEL         =       175D                    #       OK TO READ LR           DO NOT READ LR
READVBIT        =       BIT5                    #       VELOCITY DATA           VELOCITY DATA

# BIT 4 FLAG 11 (L)(R12)
RNGEDATA        =       176D                    #       LR ALTITUDE             LR ALTITUDE MEASURE
RNGEDBIT        =       BIT4                    #       MEASUREMENT MADE        NOT MADE

# BIT 3 FLAG 11
SCALBAD         =       177D                    #       LR LOW SCALE DISP-      LS SCALE DISCRETE
SCABBIT         =       BIT3                    #       CRETE NOT PRESENT       APPEARS OK
                                                #       WHEN IT SHOULD

# BIT 2 FLAG 11 (L)(R12)
VFLSHFLG        =       178D                    #       LR VELOCITY FAIL        LR VEL FAIL LAMP
                                                #       LAMP SHOULD BE          SHOULDN'T FLASH
VFLSHBIT        =       BIT2                    #       FLASHING

# BIT 1 FLAG 11 (L)(R12)
```

```
HFLSHFLG        =       179D                    #       LR ALTITUDE FAIL        LR ALTITUDE FAIL
HFLSHBIT        =       BIT1                    #       LAMP SHOULD BE          LAMP SHOULD NOT BE
                                                #       FLASHING                FLASHING

RADMODES        EQUALS  FLGWRD12                # RADAR FLAG WORD

FLGWRD12        =       STATE +12D              # (180-194)             WAS RADMODES

                                                #       (SET)                   (RESET)

# BIT 15 FLAG 12
CDESFLAG        =       180D                    #       CONTINUOUS DESIG-       LGC CHECKS FOR LOCK-
CDESBIT         =       BIT15                   #       NATE, LGC COMMANDS      ON WHEN ANTENNA
                                                #       RR REGARDLESS OF        BEING DESIGNATED
                                                #       LOCK-ON

# BIT 14 FLAG 12
REMODFLG        =       181D                    #       CHANGE IN ANTENNA       NO REMODE REQUESTED
REMODBIT        =       BIT14                   #       MODE BEEN REQUESTED     OR OCCURRING
                                                #       I.E., REMODE

# BIT 13 FLAG 12
RCDUOFLG        =       182D                    #       RR CDU'S BEING          RR CDU'S NOT BEING
RCDUOBIT        =       BIT13                   #       ZEROED                  ZEROED

# BIT 12 FLAG 12
ANTENFLG        =       183D                    #       RR ANTENNA MODE IS      RR ANTENNA IN MODE 1
ANTENBIT        =       BIT12                   #       MODE 2

# BIT 11 FLAG 12
REPOSMON        =       184D                    #       REPOSITION MONITOR.     NO REPOSITION TAKING
REPOSBIT        =       BIT11                   #       RR REPOSITION IS        PLACE
                                                #       TAKING PLACE

# BIT 10 FLAG 12
DESIGFLG        =       185D                    #       RR DESIGNATE            RR DESIGNATE NOT
DESIGBIT        =       BIT10                   #       REQUESTED OR IN         REQUESTED OR IN
                                                #       PROGRESS                PROGRESS

# BIT 9 FLAG 12
ALTSCALE        =       186D                    #       LR ALTITUDE READING     LR ALTITUDE READING
ALTSCBIT        =       BIT9                    #       IS ON HIGH SCALE        IS ON LOW SCALE
```

```
# BIT 8 FLAG 12
LRVELFLG        =       187D                    #       LR VELOCITY DATA        NO LR VELOCITY DATA
LRVELBIT        =       BIT8                    #       FAIL                    FAIL

# BIT 7 FLAG 12
RCDUFAIL        =       188D                    #       RR CDU FAIL HAS         RR CDU FAIL OCCURRED
RCDUFBIT        =       BIT7                    #       NOT OCCURRED

# BIT 6 FLAG 12
LRPOSFLG        =       189D                    #       LANDING RADAR           LR POSITION 1
LRPOSBIT        =       BIT6                    #       POSITION 2

# BIT 5 FLAG 12
LRALTFLG        =       190D                    #       LR ALTITUDE DATA        NO LR ALTITUDE DATA
LRALTBIT        =       BIT5                    #       FAIL.  COULD NOT BE     FAIL
                                                #       READ SUCCESSFULLY.

# BIT 4 FLAG 12
RRDATAFL        =       191D                    #       RR DATA FAIL.           NO RR DATA FAIL
RRDATABT        =       BIT4                    #       DATA COULD NOT BE
                                                #       READ SUCCESSFULLY

# BIT 3 FLAG 12
RRRSFLAG        =       192D                    #       RR RANGE READING        RR RANGE READING ON
RRRSBIT         =       BIT3                    #       ON THE HIGH SCALE       THE LOW SCALE

# BIT 2 FLAG 12
AUTOMODE        =       193D                    #       RR NOT IN AUTO MODE.    RR IN AUTO MODE
AUTOMBIT        =       BIT2                    #       AUTO MODE DISCRETE
                                                #       IS NOT PRESENT

# BIT 1 FLAG 12
TURNONFL        =       194D                    #       RR TURN-ON SEQUENCE     NO RR TURN-ON
TURNONBT        =       BIT1                    #       IN PROGRESS.  (ZERO     SEQUENCE IN PROGRESS
                                                #       CDU'S, FIX ANTENNA
                                                #       MODE)

DAPBOOLS        EQUALS  FLGWRD13                # DIGITAL AUTOPILOT FLAGWORD
```

```
FLGWRD13        =       STATE +13D              # (195-209)     WAS DAPBOOLS

                                                #       (SET)                   (RESET)

# BIT 15 FLAG 13
PULSEFLG        =       195D                    #       MINIMUM IMPUSE          NOT IN MINIMUM
PULSES          =       BIT15                   #       COMMAND MODE IN         IMPULSE COMMAND MODE
                                                #       "ATT HOLD" (V76)        (V77)

# BIT 14 FLAG 13
USEQRFLG        =       196D                    #       GIMBAL UNUSABLE.        TRIM GIMBAL MAY BE
USEQRJTS        =       BIT14                   #       USE JETS ONLY.          USED.

# BIT 13 FLAG 13
CSMDKFLG        =       197D                    #       CSM DOCKED.  USE        CSM NOT DOCKED TO LM
CSMDOCKD        =       BIT13                   #       BACKUP DAP

# BIT 12 FLAG 13
OURRCFLG        =       198D                    #       CURRENT DAP PASS        CURRENT DAP PASS IS
OURRCBIT        =       BIT12                   #       IS RATE COMMAND         NOT RATE COMMAND

# BIT 11 FLAG 13
ACC4-2FL        =       199D                    #       4 JET X-AXIS TRANS-     2 JET X-AXIS TRANS-
ACC4OR2X        =       BIT11                   #       LATION REQUESTED        LATION REQUESTED

# BIT 10 FLAG 13
AORBTFLG        =       200D                    #       B SYSTEM FOR X-         A SYSTEM FOR X-
AORBTRAN        =       BIT10                   #       TRANSLATION             TRANSLATION PREFER'D

# BIT 9 FLAG 13
XOVINFLG        =       201D                    #       X-AXIS OVERRIDE         X-AXIS OVERRIDE OKAY
XOVINHIB        =       BIT9                    #       LOCKED OUT

# BIT 8 FLAG 13
DRIFTDFL        =       202D                    #       ASSUME 0 OFFSET         USE OFFSET ACCELERA-
DRIFTBIT        =       BIT8                    #       DRIFTING FLIGHT         ION ESTIMATE

# BIT 7 FLAG 13
RHCSCFLG        =       203D                    #       NORMAL RHC SCALING      FINE RHC SCALING
RHCSCALE        =       BIT7                    #       REQUESTED               REQUESTED
```

```
# BIT 6 FLAG 13
ULLAGFLG        =       204D                    #       ULLAGE REQUEST BY       NO INTERNAL ULLAGE
ULLAGER         =       BIT6                    #       MISSION PROGRAM         REQUEST

# BIT 5 FLAG 13
AORBSFLG        =       205D                    #       P-AXIS COUPLES 7.15     P-AXIS COUPLES 4.12
AORBSYST        =       BIT5                    #       AND 8.16 PREFERRED      AND 3.11 PREFERRED

# BIT 4 FLAG 13
DBSELFLG        =       206D                    #       MAX DB SELECTED         MIN DB SELECTED BY
DBSELECT        =       BIT4                    #       BY CREW (5 DEG)         CREW (0.3 DEG)

# BIT 3 FLAG 13
ACCOKFLG        =       207D                    #       CONTROL AUTHORITY       RESTART OR FRESH ST.
ACCSOKAY        =       BIT3                    #       VALUES FROM 1/ACCS      SINCE LAST 1/ACCS;
                                                #       USABLE                  OUTPUTS SUSPECT.

# BIT 2 FLAG 13
AUTR2FLG        =       208D                    # THESE FLAGS ARE USED TOGETHER TO INDICATE
AUTRATE2        =       BIT2                    # ASTRONAUT-CHOSEN KALCMANU MANEUVER RATES
                                                # (0,0)=(BIT2,BIT1)=    0.2 DEG/SEC
# BIT 1 FLAG 13                                 # (0,1)=                0.5 DEG/SEC
AUTR1FLG        =       209D                    # (1,0)=                2.0 DEG/SEC
AUTRATE1        =       BIT1                    # (1,1)=               10.0 DEG/SEC
```

```
# CONVENTIONS AND NOTATIONS UTILIZED FOR ERASABLE ASSIGNMENTS.

#       EQUALS  IS USED IN TWO WAYS.  IT IS OFTEN USED TO CHAIN A GROUP
#               OF ASSIGNMENTS SO THAT THE GROUP MAY BE MOVED WITH THE
#               CHANGING OF ONLY ONE CARD.  EXAMPLE:
#
#                       X       EQUALS  START
#                       Y       EQUALS  X       +SIZE.X
#                       Z       EQUALS  Y       +SIZE.Y
#
#               (X, Y, AND Z ARE CONSECUTIVE AND BEGIN AT START.
#               SIZE.X AND SIZE.Y ARE THE RESPECTIVE SIZES OF X AND Y.
#               USUALLY NUMERIC, IE. 1, 2, 6, 18D, ETC.)

#       EQUALS  OFTEN IMPLIES THE SHARING OF REGISTERS (DIFFERENT NAMES
#               AND DIFFERENT DATA).  EXAMPLE:
#
#                       X       EQUALS  Y

#       =       MEANS THAT MULTIPLE NAMES HAVE BEEN GIVEN TO THE SAME DATA.
#               (THIS IS LOGICAL EQUIVALENCE, NOT SHARING.)  EXAMPLE:
#
#                       X       =       Y

#       THE SIZE AND UTILIZATION OF AN ERASABLE ARE OFTEN INCLUDED IN
#       THE COMMENTS IN THE FOLLOWING FORM:  M(SIZE)N.
#
#               M       REFERS TO THE MOBILITY OF THE ASSIGNMENT.
#                       B       MEANS THAT THE SYMBOL IS REFERENCED BY BASIC
#                               INSTRUCTIONS AND THUS IS E-BANK SENSITIVE.
#                       I       MEANS THAT THE SYMBOL IS REFERENCED ONLY BY
#                               INTERPRETIVE INSTRUCTIONS, AND IS THUS E-BANK
#                               INSENSITIVE AND MAY APPEAR IN ANY E-BANK.
#
#               SIZE    IS THE NUMBER OF REGISTERS INCLUDED BY THE SYMBOL.
#
#               N       INDICATES THE NATURE OF PERMANENCE OF THE CONTENTS.
#                       PL      MEANS THAT THE CONTENTS ARE PAD LOADED.
#                       DSP     MEANS THAT THE REGISTER IS USED FOR A DISPLAY.
#                       PRM     MEANS THAT THE REGISTER IS PERMANENT.  IE., IT
#                               IS USED DURING THE ENTIRE MISSION FOR ONE
#                               PURPOSE AND CANNOT BE SHARED.
#                       TMP     MEANS THAT THE REGISTER IS USED TEMPORARILY OR
#                               IS A SCRATCH REGISTER FOR THE ROUTINE TO WHICH
#                               IT IS ASSIGNED.  THAT IS, IT NEED NOT BE SET
#                               PRIOR TO INVOCATION OF THE ROUTINE NOR DOES IT
#                               CONTAIN USEFUL OUTPUT TO ANOTHER ROUTINE.  THUS
```

```
#                                IT MAY BE SHARED WITHANY OTHER ROUTINE WHICH
#                                IS NOT ACTIVE IN PARALLEL
#                        IN      MEANS INPUT TO THE ROUTINE AND IT IS PROBABLY
#                                TEMPORARY FOR A HIGHER-LEVEL ROUTINE/PROGRAM.
#                        OUT     MEANS OUTPUT FROM THE ROUTINE, PROBABLY
#                                TEMPORARY FOR A HIGHER-LEVEL ROUTINE/PROGRAM.
```

```
# SPECIAL REGISTERS.

A               EQUALS  0
L               EQUALS  1               # L AND Q ARE BOTH CHANNELS AND REGISTERS
Q               EQUALS  2
EBANK           EQUALS  3
FBANK           EQUALS  4
Z               EQUALS  5               # ADJACENT TO FBANK AND BBANK FOR DXCH Z
BBANK           EQUALS  6               # (DTCB) AND DXCH FBANK (DTCF).
                                        # REGISTER 7 IS A ZERO-SOURCE, USED BY ZL.

ARUPT           EQUALS  10              # INTERRUPT STORAGE
LRUPT           EQUALS  11
QRUPT           EQUALS  12
SAMPTIME        EQUALS  13              # SAMPLED TIME 1 & 2.
ZRUPT           EQUALS  15              # (13 AND 14 ARE SPARES.)
BANKRUPT        EQUALS  16              # USUALLY HOLDS FBANK OR BBANK.
BRUPT           EQUALS  17              # RESUME ADDRESS AS WELL.

CYR             EQUALS  20
SR              EQUALS  21
CYL             EQUALS  22
EDOP            EQUALS  23              # EDITS INTERPRETIVE OPERATION CODE PAIRS.

TIME2           EQUALS  24
TIME1           EQUALS  25
TIME3           EQUALS  26
TIME4           EQUALS  27
TIME5           EQUALS  30
TIME6           EQUALS  31
CDUX            EQUALS  32
CDUY            EQUALS  33
CDUZ            EQUALS  34
CDUT            EQUALS  35              # REND RADAR TRUNNION CDU
CDUS            EQUALS  36              # REND RADAR SHAFT CDU
PIPAX           EQUALS  37
PIPAY           EQUALS  40
PIPAZ           EQUALS  41
Q-RHCCTR        EQUALS  42              # RHC COUNTER REGISTERS
P-RHCCTR        EQUALS  43
R-RHCCTR        EQUALS  44
INLINK          EQUALS  45
RNRAD           EQUALS  46
GYROCMD         EQUALS  47
CDUXCMD         EQUALS  50
CDUYCMD         EQUALS  51
CDUZCMD         EQUALS  52
CDUTCMD         EQUALS  53
CDUSCMD         EQUALS  54
```

```
THRUST          EQUALS  55
LEMONM          EQUALS  56
OUTLINK         EQUALS  57
ALTM            EQUALS  60

# INTERPRETIVE REGISTERS ADDRESSED RELATIVE TO VAC AREA.

LVSQUARE        EQUALS  34D             # SQUARE OF VECTOR INPUT TO ABVAL AND UNIT
LV              EQUALS  36D             # LENGTH OF VECTOR INPUT TO UNIT.
X1              EQUALS  38D             # INTERPRETIVE SPECIAL REGISTER RELATIVE
X2              EQUALS  39D             # TO THE WORK AREA.
S1              EQUALS  40D
S2              EQUALS  41D
QPRET           EQUALS  42D

# INPUT/OUTPUT CHANNELS

# *** CHANNEL ZERO IS TO BE USED IN AN INDEXED OPERATION ONLY. ***
LCHAN           EQUALS  L
QCHAN           EQUALS  Q
HISCALAR        EQUALS  3
LOSCALAR        EQUALS  4
CHAN5           EQUALS  5
CHAN6           EQUALS  6
SUPERBNK        EQUALS  7               # SUPER-BANK.
OUT0            EQUALS  10
DSALMOUT        EQUALS  11
CHAN12          EQUALS  12
CHAN13          EQUALS  13
CHAN14          EQUALS  14
MNKEYIN         EQUALS  15
NAVKEYIN        EQUALS  16
CHAN30          EQUALS  30
CHAN31          EQUALS  31
CHAN32          EQUALS  32
CHAN33          EQUALS  33
DNTM1           EQUALS  34
DNTM2           EQUALS  35

# END OF CHANNEL ASSIGNMENTS
```

```
# INTERPRETIVE SWITCH BIT ASSIGNMENTS

#       ** FLAGWORDS AND BITS NOW ASSIGNED AND DEFINED IN THEIR OWN LOG SECTION. **
```

```
# GENERAL ERASABLE ASSIGNMENTS

                SETLOC  61
# INTERRUPT TEMPORARY STORAGE POOL.     (11D)

# (ITEMP1 THROUGH RUPTREG4)

# ANY OF THESE MAY BE USED AS TEMPORARIES DURING INTERRUPT OR WITH INTERRUPT INHIBITED. THE ITEMP SERIES
# IS USED DURING CALLS TO THE EXECUTIVE AND WAITLIST -- THE RUPTREGS ARE NOT.

ITEMP1          ERASE
WAITEXIT        EQUALS  ITEMP1
EXECTEM1        EQUALS  ITEMP1

ITEMP2          ERASE
WAITBANK        EQUALS  ITEMP2
EXECTEM2        EQUALS  ITEMP2

ITEMP3          ERASE
RUPTSTOR        EQUALS  ITEMP3
WAITADR         EQUALS  ITEMP3
NEWPRIO         EQUALS  ITEMP3

ITEMP4          ERASE
LOCCTR          EQUALS  ITEMP4
WAITTEMP        EQUALS  ITEMP4

ITEMP5          ERASE
NEWLOC          EQUALS  ITEMP5

ITEMP6          ERASE
NEWLOC+1        EQUALS  ITEMP6          # DP ADDRESS.

                SETLOC  67
NEWJOB          ERASE                   # MUST BE AT LOC 67 DUE TO WIRING.

RUPTREG1        ERASE
RUPTREG2        ERASE
RUPTREG3        ERASE
RUPTREG4        ERASE
KEYTEMP1        EQUALS  RUPTREG4
DSRUPTEM        EQUALS  RUPTREG4

# FLAGWORD RESERVATIONS.                (16D)

STATE           ERASE   +15D            # +15D FLAGWORD REGISTERS

# P25 RADAR STORAGE.  (MAY BE UNSHARED IN E7)  (TEMP OVERLAY)  (2D)  OVERLAYS FLGWRD 14 & 15
```

```
LASTYCMD        EQUALS  STATE +14D      # B(1)PRM       THESE ARE CALLED BY T4RUPT
LASTXCMD        EQUALS  LASTYCMD +1     # B(1)PRM       THEY MUST BE CONTIGUOUS, Y FIRST

# EXEC TEMPORARIES WHICH MAY BE USED BETWEEN CCS NEWJOBS (32D) (INTB15+ THROUGH RUPTMXTM)
INTB15+         ERASE                   # REFLECTS 15TH BIT OF INDEXABLE ADDRESSES
DSEXIT          =       INTB15+         # RETURN FOR DSPIN
EXITEM          =       INTB15+         # RETURN FOR SCALE FACTOR ROUTINE SELECT
BLANKRET        =       INTB15+         # RETURN FOR 2BLANK

INTBIT15        ERASE                   # SIMILAR TO ABOVE.
WRDRET          =       INTBIT15        # RETURN FOR 5BLANK
WDRET           =       INTBIT15        # RETURN FOR DSPWD
DECRET          =       INTBIT15        # RETURN FOR PUTCOM(DEC LOAD)
21/22REG        =       INTBIT15        # TEMP FOR CHARIN

# THE REGISTERS BETWEEN ADDRWD AND PRIORITY MUST STAY IN THE FOLLOWING ORDER FOR INTERPRETIVE TRACE.

ADDRWD          ERASE                   # 12 BIT INTERPRETIVE OPERAND SUB-ADDRESS.
POLISH          ERASE                   # HOLDS CADR MADE FROM POLISH ADDRESS.
UPDATRET        =       POLISH          # RETURN FOR UPDATNN, UPDATVB
CHAR            =       POLISH          # TEMP FOR CHARIN
ERCNT           =       POLISH          # COUNTER FOR ERROR LIGHT RESET
DECOUNT         =       POLISH          # COUNTER FOR SCALING AND DISPLAY (DEC)

FIXLOC          ERASE                   # WORK AREA ADDRESS.
OVFIND          ERASE                   # SET NON-ZERO ON OVERFLOW.

VBUF            ERASE   +5              # TEMPORARY STORAGE USED FOR VECTORS.
SGNON           =       VBUF            # TEMP FOR +,- ON
NOUNTEM         =       VBUF            # COUNTER FOR MIXNOUN FETCH
DISTEM          =       VBUF            # COUNTER FOR OCTAL DISPLAY VERB
DECTEM          =       VBUF            # COUNTER FOR FETCH (DEC DISPLAY VERBS)

SGNOFF          =       VBUF +1         # TEMP FOR +,- ON
NVTEMP          =       VBUF +1         # TEMP FOR NVSUB
SFTEMP1         =       VBUF +1         # STORAGE FOR SF CONST HI PART (=SFTEMP2-1)
HITEMIN         =       VBUF +1         # TEMP FOR LOAD OF HRS,MIN,SEC
                                        # MUST = LOTEMIN-1.

CODE            =       VBUF +2         # FOR DSPIN
SFTEMP2         =       VBUF +2         # STORAGE FOR SF CONST LO PART (=SFTEMP1+1)
LOTEMIN         =       VBUF +2         # TEMP FOR LOAD OF HRS,MIN,SEC
                                        # MUST = HITEMIN+1

MIXTEMP         =       VBUF +3         # FOR MIXNOUN DATA
SIGNRET         =       VBUF +3         # RETURN FOR +,- ON

# ALSO MIXTEMP+1 = VBUF+4, MIXTEMP+2 = VBUF+5

BUF             ERASE   +2              # TEMPORARY SCALAR STORAGE.
```

```
BUF2            ERASE   +1
INDEXLOC        EQUALS  BUF             # CONTAINS ADDRESS OF SPECIFIED INDEX.
SWWORD          EQUALS  BUF             # ADDRESS OF SWITCH WORD.
SWBIT           EQUALS  BUF +1          # SWITCH BIT WITHIN THE SWITCH WORD
MPTEMP          ERASE                   # TEMPORARY USED IN MULTIPLY AND SHIFT
DMPNTEMP        =       MPTEMP          # DMPSUB TEMPORARY
DOTINC          ERASE                   # COMPONENT INCREMENT FOR DOT SUBROUTINE
DVSIGN          EQUALS  DOTINC          # DETERMINES SIGN OF DDV RESULT
ESCAPE          EQUALS  DOTINC          # USED IN ARCSIN/ARCCOS.
ENTRET          =       DOTINC          # EXIT FROM ENTER

DOTRET          ERASE                   # RETURN FROM DOT SUBROUTINE
DVNORMCT        EQUALS  DOTRET          # DIVIDENT NORMALIZATION COUNT IN DDV.
ESCAPE2         EQUALS  DOTRET          # ALTERNATE ARCSIN/ARCCOS SWITCH
WDCNT           =       DOTRET          # CHAR COUNTER FOR DSPWD
INREL           =       DOTRET          # INPUT BUFFER SELECTION ( X,Y,Z, REG)

MATINC          ERASE                   # VECTOR INCREMENT IN MXV AND VXM
MAXDVSW         EQUALS  MATINC          # +0 IF DP QUOTIENT IS NEAR ONE -- ELSE -1.
POLYCNT         EQUALS  MATINC          # POLYNOMIAL LOOP COUNTER
DSPMMTEM        =       MATINC          # DSPCOUNT SAVE FOR DSPMM
MIXBR           =       MATINC          # INDICATOR FOR MIXED OR NORMAL NOUN

TEM1            ERASE                   # EXEC TEMP
POLYRET         =       TEM1
DSREL           =       TEM1            # REL ADDRESS FOR DSPIN

TEM2            ERASE                   # EXEC TEMP
DSMAG           =       TEM2            # MAGNITUDE STORE FOR DSPIN
IDADDTEM        =       TEM2            # MIXNOUN INDIRECT ADDRESS (GARBLED)

TEM3            ERASE                   # EXEC TEMP
COUNT           =       TEM3            # FOR DSPIN

TEM4            ERASE                   # EXEC TEMP
LSTPTR          =       TEM4            # LIST POINTER FOR GRABUSY
RELRET          =       TEM4            # RETURN FOR RELDSP
FREERET         =       TEM4            # RETURN FOR FREEDSP
DSPWDRET        =       TEM4            # RETURN FOR DSPSIGN
SEPSCRET        =       TEM4            # RETURN FOR SEPSEC
SEPMNRET        =       TEM4            # RETURN FOR SEPMIN

TEM5            ERASE                   # EXEC TEMP
NOUNADD         =       TEM5            # TEMP STORAGE FOR NOUN ADDRESS

NNADTEM         ERASE                   # TEMP FOR NOUN ADDRESS TABLE ENTRY
NNTYPTEM        ERASE                   # TEMP FOR NOUN TYPE TABLE ENTRY
IDAD1TEM        ERASE                   # TEMP FOR INDIR ADDRESS TABLE ENTRY (MIXNN)
                                        # MUST = IDAD2TEM-1, = IDAD3TEM-2
IDAD2TEM        ERASE                   # TEMP FOR INDIR ADDRESS TABLE ENTRY (MIXNN)
```

```
                                        # MUST = IDAD1TEM+1, = IDAD3TEM-1.
IDAD3TEM        ERASE                   # TEMP FOR INDIR ADDRESS TABLE ENTRY (MIXNN)
                                        # MUST = IDAD1TEM+2, = IDAD2TEM+1.
RUTMXTEM        ERASE                   # TEMP FOR SF ROUT TABLE ENTRY (MIXNN ONLY)

# AX*SR*T STORAGE.                      (6D)
DEXDEX          EQUALS  TEM2            # B(1)TMP
DEX1            EQUALS  TEM3            # B(1)TMP
DEX2            EQUALS  TEM4            # B(1)TMP
RTNSAVER        EQUALS  TEM5            # B(1)TMP
TERM1TMP        EQUALS  MPAC +3         # B(2)TMP

DEXI            =       DEX1

# THE FOLLOWING 10 REGISTERS ARE USED FOR TEMPORARY STORAGE OF THE DERIVATIVE COEFFICIENT TABLE OF
# SUBROUTINE ROOTPSRS.  THEY MUST REMAIN WITHOUT INTERFERENCE WITH ITS SUBROUTINES WHICH ARE POWRSERS (POLY).
# DMPSUB, DMPNSUB, SHORTMP, DDV/BDDV, ABS, AND USPRCADR.

DERCOF-8        =       MPAC -12        # ROOTPSRS DER COF N-4 HI ORDER
DERCOF-7        =       MPAC -11        # ROOTPSRS DER COF N-4 LO ORDER
DERCOF-6        =       MPAC -10        # ROOTPSRS DER COF N-3 HI ORDER
DERCOF-5        =       MPAC -7         # ROOTPSRS DER COF N-3 LO ORDER
DERCOF-4        =       MPAC -6         # ROOTPSRS DER COF N-2 HI ORDER
DERCOF-3        =       MPAC -5         # ROOTPSRS DER COF N-2 LO ORDER
DERCOF-2        =       MPAC -4         # ROOTPSRS DER COF N-1 HI ORDER
DERCOF-1        =       MPAC -3         # ROOTPSRS DER COF N-1 LO ORDER
DERCOFN         =       MPAC -2         # ROOTPSRS DER COF N, HI ORDER
DERCOF+1        =       MPAC -1         # ROOTPSRS DER COF N, LO ORDER

PWRPTR          =       POLISH          # ROOTPSRS POWER TABLE POINTER
DXCRIT          =       VBUF +2         # ROOTPSRS CRITERION FOR ENDING ITERS HI
DXCRIT+1        =       VBUF +3         # ROOTPSRS CRITERION FOR ENDING ITERS LOW
ROOTPS          =       VBUF +4         # ROOTPSRS ROOT HI ORDER
ROOTPS+1        =       VBUF +5         # ROOTPSRS ROOT LO ORDER
RETROOT         =       BUF +2          # ROOTPSRS RETURN ADDRESS OF USER
PWRCNT          =       MATINC          # ROOTPSRS DER TABLE LOOP COUNTER
DERPTR          =       TEM1            # ROOTPSRS DER TABLE POINTER
```

```
# DYNAMICALLY ALLOCATED CORE SETS FOR JOBS      (84D)

MPAC            ERASE   +6              # MULTI-PURPOSE ACCUMULATOR.
MODE            ERASE                   # +1 FOR TP, +0 FOR DP, OR -1 FOR VECTOR.
LOC             ERASE                   # LOCATION ASSOCIATED WITH JOB.
BANKSET         ERASE                   # USUALLY CONTAINS BBANK SETTING.
PUSHLOC         ERASE                   # WORD OF PACKED INTERPRETIVE PARAMETERS.
PRIORITY        ERASE                   # PRIORITY OF PRESENT JOB AND WORK AREA.

                ERASE   +83D            # EIGHT SETS OF 12 REGISTERS EACH

# INCORP STORAGE:  R22 (N29)    (SHARES WITH FOLLOWING SECTION)         (4D)

R22DISP         EQUALS  TIME2SAV        # I(4) N49 DISPLAY OF DELTA R AND DELTA V

# STANDBY VERB ERASABLES.  REDOCTR BEFORE THETADS.      (14D)

TIME2SAV        ERASE   +1
SCALSAVE        ERASE   +1
REDOCTR         ERASE                   # CONTAINS NUMBER OF RESTARTS
THETAD          ERASE   +2
CPHI            =       THETAD          # O     DESIRED GIMBAL ANGLES
CTHETA          =       THETAD +1       # I     FOR
CPSI            =       THETAD +2       # M     MANEUVER
DELV            ERASE   +5
DELVX           =       DELV
DELVY           =       DELV +2
DELVZ           =       DELV +4


# DOWNLINK STORAGE.                     (28D)

DNLSTADR        EQUALS  DNLSTCOD

DNLSTCOD        ERASE                   # B(1)PRM DOWNLINK LIST CODE
DUMPCNT         ERASE                   # B(1)
LDATALST        ERASE   +25D            # (26D)
DNTMGOTO        EQUALS  LDATALST +1     # B(1)
TMINDEX         EQUALS  DNTMGOTO +1     # B(1)
DUMPLOC         EQUALS  TMINDEX         # CONTAINS ECADR OF AGC DP WORD BEING DUMPED
                                        # AND COUNT OF COMPLETE DUMPS ALREADY
                                        # SENT.
DNQ             EQUALS  TMINDEX +1      # B(1)
DNTMBUFF        EQUALS  DNQ +1          # B(22)PRM DOWNLINK SNAPSHOT BUFFER

# UNSWITCHED FOR DISPLAY INTERFACE ROUTINES.    (10D) FIVE MORE IN EBANK 2.
```

```
RESTREG         ERASE                   # B(1)PRM FOR DISPLAY RESTARTS
NVWORD          ERASE
MARKNV          ERASE
NVSAVE          ERASE
# (RETAIN THE ORDER OF CADRFLSH TO FAILREG +2 FOR DOWNLINK PURPOSES)
CADRFLSH        ERASE
CADRMARK        ERASE
TEMPFLSH        ERASE
FAILREG         ERASE   +2              # B(3)PRM 3 ALARM CODE REGISTERS

# VAC AREAS. -- BE CAREFUL OF PLACEMENT --       (220D)

VAC1USE         ERASE
VAC1            ERASE   +42D
VAC2USE         ERASE
VAC2            ERASE   +42D
VAC3USE         ERASE
VAC3            ERASE   +42D
VAC4USE         ERASE
VAC4            ERASE   +42D
VAC5USE         ERASE
VAC5            ERASE   +42D

# WAITLIST REPEAT FLAG.                 (1D)
RUPTAGN         ERASE
KEYTEMP2        =       RUPTAGN         # TEMP FOR KEYRUPT, UPRUPT

# STARALIGN ERASABLES.                  (13D)

STARCODE        ERASE                   # (1)
AOTCODE         =       STARCODE
STARALGN        ERASE   +11D
SINCDU          =       STARALGN
COSCDU          =       STARALGN +6

SINCDUX         =       SINCDU +4
SINCDUY         =       SINCDU
SINCDUZ         =       SINCDU +2
COSCDUX         =       COSCDU +4
COSCDUY         =       COSCDU
COSCDUZ         =       COSCDU +2

# PHASE TABLE AND RESTART COUNTERS              (12D)

-PHASE1         ERASE
```

```
PHASE1          ERASE
-PHASE2         ERASE
PHASE2          ERASE
-PHASE3         ERASE
PHASE3          ERASE
-PHASE4         ERASE
PHASE4          ERASE
-PHASE5         ERASE
PHASE5          ERASE
-PHASE6         ERASE
PHASE6          ERASE

# A**SR*T STORAGE.                      (6D)

CDUSPOT         ERASE   +5              # B(6)

CDUSPOTY        =       CDUSPOT
CDUSPOTZ        =       CDUSPOT +2
CDUSPOTX        =       CDUSPOT +4

# VERB 37 STORAGE                       (2D)

MINDEX          ERASE                   # B(1)TMP INDEX FOR MAJOR MODE
MMNUMBER        ERASE                   # B(1)TMP MAJOR MODE REQUESTED BY V37

# PINBALL INTERRUPT ACTION              (1D)

DSPCNT          ERASE                   # B(1)PRM COUNTER FOR DSPOUT

# PINBALL EXECUTIVE ACTION              (44D)

DSPCOUNT        ERASE                   # DISPLAY POSITION INDICATOR
DECBRNCH        ERASE                   # +DEC, -DEC, OCT INDICATOR
VERBREG         ERASE                   # VERB CODE
NOUNREG         ERASE                   # NOUN CODE
XREG            ERASE                   # R1 INPUT BUFFER
YREG            ERASE                   # R2 INPUT BUFFER
ZREG            ERASE                   # R3 INPUT BUFFER
XREGLP          ERASE                   # LO PART OF XREG (FOR DEC CONV ONLY)
YREGLP          ERASE                   # LO PART OF YREG (FOR DEC CONV ONLY)
HITEMOUT        =       YREGLP          # TEMP FOR DISPLAY OF HRS,MIN,SEC
                                        # MUST = LOTEMOUT-1.
ZREGLP          ERASE                   # LO PART OF ZREG (FOR DEC CONV ONLY)
LOTEMOUT        =       ZREGLP          # TEMP FOR DISPLAY OF HRS,MIN,SEC
                                        # MUST = HITEMOUT+1
MODREG          ERASE                   # MODE CODE
```

```
DSPLOCK         ERASE                   # KEYBOARD/SUBROUTINE CALL INTERLOCK
REQRET          ERASE                   # RETURN REGISTER FOR LOAD
LOADSTAT        ERASE                   # STATUS INDICATOR FOR LOADTST
CLPASS          ERASE                   # PASS INDICATOR CLEAR
NOUT            ERASE                   # ACTIVITY COUNTER FOR DSPTAB
NOUNCADR        ERASE                   # MACHINE CADR FOR NOUN
MONSAVE         ERASE                   # N/V CODE FOR MONITOR. (= MONSAVE1-1)
MONSAVE1        ERASE                   # NOUNCADR FOR MONITOR (MATBS1) = MONSAVE+1
MONSAVE2        ERASE                   # NVMONOPT OPTIONS
DSPTAB          ERASE   +11D            # 0-10D, DISPLAY PANEL BUFF.  11D, C/S LTS.
NVQTEM          ERASE                   # NVSUB STORAGE FOR CALLING ADDRESS
                                        # MUST = NVBNKTEM-1.
NVBNKTEM        ERASE                   # NVSUB STORAGE FOR CALLING BANK
                                        # MUST = NVQTEM+1
VERBSAVE        ERASE                   # NEEDED FOR RECYCLE
CADRSTOR        ERASE                   # ENDIDLE STORAGE
DSPLIST         ERASE                   # WAITING REG FOR DSP SYST INTERNAL USE
EXTVBACT        ERASE                   # EXTENDED VERB ACTIVITY INTERLOCK
DSPTEM1         ERASE   +2              # BUFFER STORAGE AREA 1 (MOSTLY FOR TIME)
DSPTEM2         ERASE   +2              # BUFFER STORAGE AREA 2 (MOSTLY FOR DEG)

DSPTEMX         EQUALS  DSPTEM2 +1      # B(2) S-S DISPLAY BUFFER FOR EXT. VERBS
NORMTEM1        EQUALS  DSPTEM1         # B(3)DSP NORMAL DISPLAY REGISTERS.

# DISPLAY FOR EXTENDED VERBS (V82, R04(V62), V41(N72) ) (2D)

OPTIONX         EQUALS  DSPTEMX         # (2) EXTENDED VERB OPTION CODE

# TBASES AND PHSPRDT S.                 (12D)

TBASE1          ERASE
PHSPRDT1        ERASE
TBASE2          ERASE
PHSPRDT2        ERASE
TBASE3          ERASE
PHSPRDT3        ERASE
TBASE4          ERASE
PHSPRDT4        ERASE
TBASE5          ERASE
PHSPRDT5        ERASE
TBASE6          ERASE
PHSPRDT6        ERASE

# UNSWITCHED FOR DISPLAY INTERFACE ROUTINES.    (6D)
```

```
NVWORD1         ERASE                   # B(1) PROBABLY FOR DISPLAY DURING SERVICER
EBANKSAV        ERASE
MARKEBAN        ERASE
EBANKTEM        ERASE
MARK2PAC        ERASE
R1SAVE          ERASE

# IMU COMPENSATION UNSWITCHED ERASABLE.         (1D)

1/PIPADT        ERASE

# SINGLE PRECISION SUBROUTINE TEMPORARIES       (2D)

TEMK            ERASE                   # (1)
SQ              ERASE                   # (1)

# UNSWITCHED RADAR ERASABLE

SAMPLIM         ERASE
SAMPLSUM        ERASE   +3
TIMEHOLD        ERASE   +1
RRTARGET        EQUALS  SAMPLSUM        # HALF U IT VECTOR IN SM OR NB AXES.
TANG            ERASE   +1              # DESIRE TRUNNION AND SHAFT ANGLES.
MODEA           EQUALS  TANG
MODEB           ERASE   +1              # DODES LOBBERS TANG +2.
NSAMP           EQUALS  MODEB
DESRET          ERASE
OLDATAGD        EQUALS  DESRET          # USED IN DATA READING ROUTINES.
DESCOUNT        ERASE

# ******   P22  ******                          (6D)

RSUBC           EQUALS  RRTARGET        # I(6) S-S CSM POSITION VECTOR.
```

```
# UNSWITCHED FOR ORBIT INTEGRATION                      (21D)

TDEC            ERASE   +20D            # I(2)
COLREG          EQUALS  TDEC +2         # I(1)
LAT             EQUALS  COLREG +1       # I(2)
LONG            EQUALS  LAT +2          # I(2)
ALT             EQUALS  LONG +2         # I(2)
YV              EQUALS  ALT +2          # I(6)
ZV              EQUALS  YV +6           # I(6)

# MISCELLANEOUS UNSWITCHED.                             (20D)

P40/RET         ERASE                   # (WILL BE PUT IN E6 WHEN THERE IS ROOM)
GENRET          ERASE                   # B(1) R61 RETURN CADR.
OPTION1         ERASE                   # B(1) NOUN 06 USES THIS
OPTION2         ERASE                   # B(1) NOUN 06 USES THIS
OPTION3         ERASE                   # B(1) NOUN 06 USES THIS
LONGCADR        ERASE +1                # B(2) LONGCALL REGISTER
LONGBASE        ERASE +1
LONGTIME        ERASE +1                # B(2) LONGCALL REGISTER
CDUTEMPX        ERASE                   # B(1)TMP
CDUTEMPY        ERASE                   # B(1)TMP
CDUTEMPZ        ERASE                   # B(1)TMP
PIPATMPX        ERASE                   # B(1)TMP
PIPATMPY        ERASE                   # B(1)TMP
PIPATMPZ        ERASE                   # B(1)TMP

DISPDEX         ERASE                   # B(1)
TEMPR60         ERASE                   # B(1)
PRIOTIME        ERASE                   # B(1)

# P27 (UPDATE PROGRAM) STORAGE                          (26D)

UPVERBSV        ERASE                   # B(1) UPDATE VERB ATTEMPTED.
UPTEMP          ERASE   +24D            # B(1)TMP SCRATCH
INTWAK1Q        EQUALS  UPTEMP          # (BORROWS UPTEMP REGISTERS)
# RETAIN THE ORDER OF COMPNUMB THRU UPBUFF +19D FOR DOWNLINK PURPOSES.
COMPNUMB        EQUALS  UPTEMP +1       # B(1)TMP NUMBER OF ITEMS TO BE UPLINKED
UPOLDMOD        EQUALS  COMPNUMB +1     # B(1)TMP INTERRUPTD PROGRAM MM
UPVERB          EQUALS  UPOLDMOD +1     # B(1)TMP VERB NUMBER
UPCOUNT         EQUALS  UPVERB +1       # B(1)TMP UPBUFF INDEX
UPBUFF          EQUALS  UPCOUNT +1      # B(20D)

# SPECIAL DEFINITION FOR SYSTEM TEST ERASABLE PGMS.     (2D)

EBUF2           EQUALS  UPTEMP          # B(2) FOR EXCLUSIVE USE OF SYSTEM TEST.
```

```
# PERM STATE VECTORS FOR BOOST AND DOWNLINK -- WHOLE MISSION -- (14D)

RN              ERASE   +5              # B(6)PRM
VN              ERASE   +5              # B(6)PRM
PIPTIME         ERASE   +1              # B(2)PRM (MUST BE FOLLOWED BY GDT/2)

# SERVICER -- MUST FOLLOW PIPTIME --            (19D)

GDT/2           ERASE   +19D            # B(6)TMP       ** MUST FOLLOW PIPTIME **
MASS            EQUALS  GDT/2 +6        # B(2)
WEIGHT/G        =       MASS
ABDELV          EQUALS  MASS +2         # (KALCMANU STORAGE)
PGUIDE          EQUALS  ABDELV +1       # (2)
DVTHRUSH        EQUALS  PGUIDE +2       # (1)
AVEGEXIT        EQUALS  DVTHRUSH +1     # (2)
AVGEXIT         =       AVEGEXIT
TEMX            EQUALS  AVEGEXIT +2     # (1)
TEMY            EQUALS  TEMX +1         # (1)
TEMZ            EQUALS  TEMY +1         # (1)
PIPAGE          EQUALS  TEMZ +1         # B(1)
OUTROUTE        EQUALS  PIPAGE +1       # B(1)

# PERMANENT LEM DAP STORAGE             (12D)

CH5MASK         ERASE                   # B(1)PRM
CH6MASK         ERASE                   # B(1)PRM JET FAILURE MASK.
DTHETASM        ERASE +5                #  (6)
SPNDX           ERASE                   # B(1)
RCSFLAGS        ERASE                   # AUTOPILOT FLAG WORD
                                        # BIT ASSIGNMENTS:
                                        #  1) ALTERYZ SWITCH (ZEROOR1)
                                        #  2) NEEDLER SWITCH
                                        #  3) NEEDLER SWITCH
                                        #  4) NEEDLER SWITCH
                                        #  5) NEEDLER SWITCH
                                        #  9) JUST-IN-DETENT SWITCH
                                        # 10) PBIT -- MANUAL CONTROL SWITCH
                                        # 11) QRBIT -- MANUAL CONTROL SWITCH
                                        # 12) PSKIP CONTROL (PJUMPADR)
                                        # 13) 1/ACCJOB CONTROL (ACCSET)
T5ADR           ERASE   +1              # GENADR OF NEXT LM DAP T5RUPT. * 2CADR *
                                        # BBCON  OF NEXT LM DAP T5RUPT.   2CADR

# ERASABLES FOR P64:  OVERLAY OF DTHETASM, WHICH IS UNUSED      (4D)
ZERLINA         EQUALS  DTHETASM        # B(1)  P64
```

```
ELVIRA          EQUALS  ZERLINA +1      # B(1)  P64
AZINCR1         EQUALS  ELVIRA  +1      # B(1)  P64
ELINCR1         EQUALS  AZINCR1 +1      # B(1)  P64

# RCS FAILURE MONITOR STORAGE           (1)

PVALVEST        ERASE                   # B(1)PRM

# KALCMANU/DAP INTERFACE                        (3D)

DELPEROR        ERASE                   # B(1)PRM COMMAND LAGS.
DELQEROR        ERASE                   # B(1)PRM
DELREROR        ERASE                   # B(1)PRM

# MODE SWITCHING ERASABLE.                      (9D)

# RETAIN THE ORDER OF IMODES30 AND IMODES33 FOR DOWNLINK PURPOSES
IMODES30        ERASE                   # B(1)
IMODES33        ERASE
MODECADR        ERASE   +2              # B(3)PRM
IMUCADR         EQUALS  MODECADR
OPTCADR         EQUALS  MODECADR +1
RADCADR         EQUALS  MODECADR +2
ATTCADR         ERASE   +2              # B(3)PRM
ATTPRIO         =       ATTCADR +2
MARKSTAT        ERASE

# T4RUPT ERASABLE                               (2D)

DSRUPTSW        ERASE
LGYRO           ERASE                   # (1)

# RENDEZVOUS RADAR TASK STORAGE                 (3D)

RRRET           ERASE   +2D             # B(1)TMP       P20'S, PERHAPS R29 & R12
RDES            EQUALS  RRRET +1        # B(1)TMP
RRINDEX         EQUALS  RDES +1         # B(1)TMP

# MEASINC                                       (4D)

WIXA            ERASE                   # B(1)
WIXB            ERASE                   # B(1)
ZIXA            ERASE                   # B(1)
ZIXB            ERASE                   # B(1)
```

```
# AGS DUMMY ID WORD.                                     (1D)

AGSWORD         ERASE

# SOME MISCELLANEOUS UNSWITCHED.                         (6D)

RATEINDX        ERASE                           # (1) USED BY KALCMANU
DELAYLOC        ERASE +2
LEMMASS         ERASE                           # KEEP CONTIGUOUS W. CSMMASS. (1) EACH
CSMMASS         ERASE

# LESS IS MORE.

# RENDEZVOUS AND LANDING RADAR DOWNLINK STORAGE.         (7D)
#
#       (NORMALLY USED DURING P20, BUT MAY ALSO)
#       (BE REQUIRED FOR THE V62 SPURIOUS TEST.)
#
#                   (PLEASE KEEP IN THIS ORDER)

DNRRANGE        ERASE   +6              # B(1)TMP
DNRRDOT         EQUALS  DNRRANGE +1     # B(1)TMP
DNINDEX         EQUALS  DNRRDOT +1      # B(1)TMP
DNLRVELX        EQUALS  DNINDEX +1      # B(1)TMP
DNLRVELY        EQUALS  DNLRVELX +1     # B(1)TMP
DNLRVELZ        EQUALS  DNLRVELY +1     # B(1)TMP
DNLRALT         EQUALS  DNLRVELZ +1     # B(1)TMP

# INCORPORATION UNSWITCHED                               (2D)

W.IND           EQUALS  PIPAGE          # B(1)
W.INDI          EQUALS  W.IND +1        # I(1)

# SUBROUTINE BALLANGS OF R60.

BALLEXIT        ERASE                   # B(1) SAVE LOCATION FOR BALLINGS SUBR EXIT

# SOME LEM DAP STORAGE.                                  (4D)

DAPDATR1        ERASE                   # B(1)DSP DAP CONFIG.
TEVENT          ERASE   +1              # B(2)DSP
DB              ERASE                   # B(1)TMP DEAD BAND.

# NOUN 87                                                (2D)
AZ              ERASE   +1D             # B(1) AZ AND EL MUST BE CONTIGUOUS
```

```
EL              EQUALS  AZ +1D          # B(1)

# P63, P64, P65, P66, AND P67.                  (1D)

WCHPHASE        ERASE                   # B(1)

# ERASABLES FOR THE R2 LUNAR POTENTIAL MODEL    (2D)

E3J22R2M        ERASE                   # I(1)
E32C31RM        ERASE                   # I(1)


RADSKAL         ERASE   +1              # LR ALT DOPPLER BIAS: 2T/LAMBDA SCALED
                                        # AT 1/(2(7) M/CS)
SKALSKAL        ERASE                   # LR ALT SCALE FACTOR RATIO: .2 NOM

END-UE          EQUALS                  # NEXT UNUSED UE ADDRESS

# SELF-CHECK ASSIGNMENTS                        (17D)

SELFERAS        ERASE   1357 - 1377     # *** MUST NOT BE MOVED ***
SFAIL           EQUALS  SELFERAS        # B(1)
ERESTORE        EQUALS  SFAIL +1        # B(1)
SELFRET         EQUALS  ERESTORE +1     # B(1) RETURN
SMODE           EQUALS  SELFRET +1      # B(1)
ALMCADR         EQUALS  SMODE +1        # B(2) ALARM-ABORT USER'S 2CADR
ERCOUNT         EQUALS  ALMCADR +2      # B(1)
SCOUNT          EQUALS  ERCOUNT +1      # B(3)
SKEEP1          EQUALS  SCOUNT +3       # B(1)
SKEEP2          EQUALS  SKEEP1 +1       # B(1)
SKEEP3          EQUALS  SKEEP2 +1       # B(1)
SKEEP4          EQUALS  SKEEP3 +1       # B(1)
SKEEP5          EQUALS  SKEEP4 +1       # B(1)
SKEEP6          EQUALS  SKEEP5 +1       # B(1)
SKEEP7          EQUALS  SKEEP6 +1       # B(1)
```

```
# EBANK-3 ASSIGNMENTS

                SETLOC  1400

# WAITLIST TASK LISTS.                              (26D)

LST1            ERASE   +7              # B(8D)PRM DELTA T'S.
LST2            ERASE   +17D            # B(18D)PRM TASK 2CADR ADDRESSES.

# RESTART STORAGE.                                  (2D)

RSBBQ           ERASE   +1              # B(2)PRM SAVE BB AND Q FOR RESTARTS

# MORE LONGCALL STORAGE.  (MUST BE IN LST1'S BANK.      (2D)

LONGEXIT        ERASE   +1              # B(2)TMP MAY BE SELDOM OVERLAYED.

# PHASE-CHANGE LISTS PART II.                       (12D)


PHSNAME1        ERASE                   # B(1)PRM
PHSBB1          ERASE                   # B(1)PRM
PHSNAME2        ERASE                   # B(1)PRM
PHSBB2          ERASE                   # B(1)PRM
PHSNAME3        ERASE                   # B(1)PRM
PHSBB3          ERASE                   # B(1)PRM
PHSNAME4        ERASE                   # B(1)PRM
PHSBB4          ERASE                   # B(1)PRM
PHSNAME5        ERASE                   # B(1)PRM
PHSBB5          ERASE                   # B(1)PRM
PHSNAME6        ERASE                   # B(1)PRM
PHSBB6          ERASE                   # B(1)PRM

# IMU COMPENSATION PARAMETERS                       (22D)

PBIASX          ERASE                   # B(1) PIPA BIAS, PIPA SCALE FACTOR TERMS
PIPABIAS        =       PBIASX          # INTERMIXED.
PIPASCFX        ERASE
PIPASCF         =       PIPASCFX
PBIASY          ERASE
PIPASCFY        ERASE
PBIASZ          ERASE
PIPASCFZ        ERASE

NBDX            ERASE                   # GYRO BIAS DRIFT
NBDY            ERASE
NBDZ            ERASE
```

```
ADIAX           ERASE                   # ACCELERATION SENSITIVE DRIFT ALONG THE
ADIAY           ERASE                   # INPUT AXIS
ADIAZ           ERASE

ADSRAX          ERASE                   # ACCELERATION SENSITIVE DRIFT ALONG THE
ADSRAY          ERASE                   # SPIN REFERENCE AXIS
ADSRAZ          ERASE

GCOMP           ERASE   +5              # CONTAINS COMPENSATING TORQUES

COMMAND         EQUALS  GCOMP
CDUIND          EQUALS  GCOMP   +3

GCOMPSW         ERASE

# STATE VECTORS FOR ORBIT INTEGRATION.          (44D)

#               (DIFEQCNT THUR XKEP MUST BE IN THE SAME
#               EBANK AS RRECTCSM, RRECTLEM ETC
#               BECAUSE THE COPY-CYCLES (ATOPCSM,
#               PTOACSM ETC) ARE EXECUTED IN BASIC.
#               ALL OTHER REFERENCES TO THIS GROUP
#               ARE BY INTERPRETIVE INSTRUCTIONS.)
#

DIFEQCNT        ERASE   +43D            # B(1)
# (UPSVFLAG...XKEP MUST BE KEPT IN ORDER)

UPSVFLAG        EQUALS  DIFEQCNT +1     # B(1)
RRECT           EQUALS  UPSVFLAG +1     # B(6)
VRECT           EQUALS  RRECT   +6      # B(6)
TET             EQUALS  VRECT   +6      # B(2)
TDELTAV         EQUALS  TET     +2      # B(6)
TNUV            EQUALS  TDELTAV +6      # B(6)
RCV             EQUALS  TNUV    +6      # B(6)
VCV             EQUALS  RCV     +6      # B(6)
TC              EQUALS  VCV     +6      # B(2)
XKEP            EQUALS  TC      +2      # B(2)

# PERMANENT STATE VECTORS AND TIMES.

# (DO NOT OVERLAY WITH ANYTHING AFTER BOOST)

# (RRECTCSM...XKEPCSM MUST BE KEPT IN THIS ORDER)

RRECTCSM        ERASE   +5              # B(6)PRM CSM VARIABLES.
RRECTOTH        =       RRECTCSM
VRECTCSM        ERASE   +5              # B(6)PRM
```

```
TETCSM          ERASE   +1              # B(2)PRM
TETOTHER        =       TETCSM
DELTACSM        ERASE   +5              # B(6)PRM
NUVCSM          ERASE   +5              # B(6)PRM
RCVCSM          ERASE   +5              # B(6)PRM
VCVCSM          ERASE   +5              # B(6)PRM
TCCSM           ERASE   +1              # B(2)PRM
XKEPCSM         ERASE   +1              # B(2)PRM

# (RRECTLEM...XKEPLEM MUST BE KEPT IN THIS ORDER)

RRECTLEM        ERASE   +5              # B(6)PRM LEM VARIABLES
RRECTHIS        =       RRECTLEM
VRECTLEM        ERASE   +5              # B(6)PRM
TETLEM          ERASE   +1              # B(2)PRM
TETTHIS         =       TETLEM
DELTALEM        ERASE   +5              # B(6)PRM
NUVLEM          ERASE   +5              # B(6)PRM
RCVLEM          ERASE   +5              # B(6)PRM
VCVLEM          ERASE   +5              # B(6)PRM
TCLEM           ERASE   +1              # B(2)PRM
XKEPLEM         ERASE   +1              # B(2)PRM

X789            ERASE   +5
TEPHEM          ERASE   +2
AZO             ERASE   +1
-AYO            ERASE   +1
AXO             ERASE   +1

# STATE VECTORS FOR DOWNLINK                    (12D)

R-OTHER         ERASE   +5              # B(6)PRM POS VECT (OTHER VECH) FOR DNLINK
V-OTHER         ERASE   +5              # B(6)PRM VEL VECT (OTHER VECH) FOR DNLINK

T-OTHER         =       TETCSM          #              TIME (OTHER VECH) FOR DNLINK

# REFSMMAT.                                     (18D)

REFSMMAT        ERASE   +17D            # I(18D)PRM

# ACTIVE VEHICLE CENTANG.  MUST BE DISPLAYED ANYTIME (ALMOST.)  (2D)

ACTCENT         ERASE   +1              # I(2) S-S CENTRAL ANGLE BETWEEN ACTIVE
                                        # VEHICLE AT TPI TIG AND TARGET VECTOR.

# **** USED IN CONICSEX (PLAN INERT ORIENT) ****
```

```
TIMSUBO          EQUALS  TEPHEM          # CSEC B-42 (TRIPLE PRECISION)

# LPS20.1 STORAGE        -- ALL ARE PRM --       (9D)

LS21X            ERASE                   # I(1)
LOSVEL           ERASE   +5              # I(6)
MLOSV            ERASE   +1              # I(2) MAGNITUDE OF LOS. METERS B-29

# ***** P22 ***** (OVERLAYS LPS 20.1 STORAGE)    (6D)
VSUBC            EQUALS  LOSVEL          # I(6) S-S CSM VELOCITY VECTOR

# PADLOADED ERASABLES FOR P20/P22                (6D)

RANGEVAR         ERASE   +1              # I(2) RR RANGE ERROR VARIANCE
RATEVAR          ERASE   +1              # I(2) RR RANGE RATE ERROR VARIANCE
RVARMIN          ERASE                   # I(1) MINIMUM RANGE ERROR VARIANCE
VVARMIN          ERASE                   # I(1) MINIMUM RANGE-RATE ERROR VARIANCE

# P32-P33 STORAGE                                (2D)

TCDH             ERASE   +1              # I(2) T2 CDH TIME IN CS. (ALSO DOWNLINKED)

END-E3           EQUALS  1777            # ** LAST LOCATION USED IN E3 **
```

```
# EBANK-4 ASSIGNMENTS

                SETLOC  2000

# E4 IS, FOR THE MOST PART RESERVED FOR PAD LOADED AND UNSHARABLE ERASE.

AMEMORY         EQUALS

# P20 STORAGE.  -- PAD LOADED --                (6D)

WRENDPOS        ERASE                   # B(1)PL        KM*2(-7)
WRENDVEL        ERASE                   # B(1)PL        KM(-1/2)*2(11)
WSHAFT          ERASE                   # B(1)PL        KM*2(-7)
WTRUN           ERASE                   # B(1)PL        KM*2(-7)
RMAX            ERASE                   # B(1)PL        METERS*2(-19)
VMAX            ERASE                   # B(1)PL        M/CSEC*2(-7)

# LUNAR SURFACE NAVIGATION                      (2D)

WSURFPOS        ERASE                   # B(1)PL
WSURFVEL        ERASE                   # B(1)PL

# P22 STORAGE.  -- PAD LOADED --                (2D)

SHAFTVAR        ERASE                   # B(1)PL        RAD SQ*2(12)
TRUNVAR         ERASE                   # B(1)PL        RAD SQ*2(10)

# CONISEX STORAGE.  -- PAD LOADED --

504LM           ERASE   +5              # I(6)  MOON LIBRATION VECTOR

# V47 (R47) AGS INITIALIZATION STORAGE.  -- PAD LOADED --       (2D)

AGSK            ERASE   +1

# LUNAR LANDING STORAGE.  -- PAD LOADED --      (6D)

RLS             ERASE   +5              # I(6)  LANDING SITE VECTOR -- MOON REF

# INTEGRATION STORAGE.                          (102D)

PBODY           ERASE   +101D           # I(1)
```

```
ALPHAV          EQUALS  PBODY +1        # I(6)
BETAV           EQUALS  ALPHAV +6       # I(6)
PHIV            EQUALS  BETAV +6        # I(6)
PSIV            EQUALS  PHIV +6         # I(6)
FV              EQUALS  PSIV +6         # I(6)  PERTURBING ACCELERATIONS
ALPHAM          EQUALS  FV +6           # I(2)
BETAM           EQUALS  ALPHAM +2       # I(2)
TAU.            EQUALS  BETAM +2        # I(2)
DT/2            EQUALS  TAU. +2         # I(2)
H               EQUALS  DT/2 +2         # I(2)
GMODE           EQUALS  H +2            # I(1)
IRETURN         EQUALS  GMODE +1        # I(1)
NORMGAM         EQUALS  IRETURN +1      # I(1)
RPQV            EQUALS  NORMGAM +1
ORIGEX          EQUALS  RPQV +6         # I(1)
KEPRTN          EQUALS  ORIGEX          # I(1)
RQVV            EQUALS  ORIGEX +1       # I(6)
RPSV            EQUALS  RQVV +6         # I(6)
XKEPNEW         EQUALS  RPSV +6         # I(2)
VECTAB          EQUALS  XKEPNEW +2      # I(36D)
VECTABND        EQUALS  VECTAB +35D     # END MARK

# THESE PROBABLY CAN SHARE MID-COURSE VARIABLES.        (6D)

VACX            EQUALS  VECTAB +6       # I(2)
VACY            EQUALS  VACX +2         # I(2)
VACZ            EQUALS  VACY +2         # I(2)

# SERVICER STORAGE (USED BY ALL POWERED FLIGHT PROGS.)  (18D)

XNBPIP          EQUALS  VECTAB +12D     # I(6)
YNBPIP          EQUALS  XNBPIP +6       # I(6)
ZNBPIP          EQUALS  YNBPIP +6       # I(6)

# SOME VERB 82 STORAGE                          (4D)

HAPOX           EQUALS  RQVV +4         # I(2)
HPERX           EQUALS  HAPOX +2        # I(2)

# V82 STORAGE                                   (6D)

VONE'           EQUALS  VECTAB +30D     # I(T)TMP NORMAL VELOCITY VONE / SQRT. MU

# R32(V83) STORAGE. -- SHARES WITH INTEGRATION STORAGE --       (28D)
```

```
BASETHV         EQUALS  RPQV            # I(6) BASE VEL VECTOR THIS VEH

BASETIME        EQUALS  RQVV            # I(2) TIME ASSOC WITH BASE VECS
ORIG            EQUALS  RQVV +2         # I(1) =0 FOR EARTH     =2 FOR MOON
STATEXIT        EQUALS  RQVV +3         # I(1) STQ ADDRESS FOR STATEXTP
BASEOTV         EQUALS  RQVV +4         # I(6) BASE VEL VECTOR OTHER VEH

BASEOTP         EQUALS  VECTAB +6       # I(6) BASE POS VECTOR OTHER VEH

BASETHP         EQUALS  VECTAB +30D     # I(6) BASE POS VECTOR THIS VEH

# KEPLER STORAGE. (KEPLER IS CALLED BY PRECISION INTEGRATION AND (2D)
# CONICS)

EPSILONT        ERASE   +1              # I(2)

# VERB 83 STORAGE                               (18D)

RANGE           ERASE   +17D            # I(2)DSP NOUN 54 DISTANCE TO OPTICAL SUBJ
RRATE           EQUALS  RANGE +2        # I(2)DSP NOUN 54 RATE OF APPROACH
RTHETA          EQUALS  RRATE +2        # I(2)DSP NOUN 54.
RONE            EQUALS  RTHETA +2       # I(6)TMP VECTOR STORAGE.  (SCRATCH)
VONE            EQUALS  RONE +6         # I(6)TMP VECTOR STORAGE.  (SCRATCH)

# VERB 67 STORAGE

WWPOS           =       RANGE           # NOUN 99 (V67)
WWVEL           =       RRATE           # NOUN 99 (V67)
WWBIAS          =       RTHETA          # NOUN 99 (V67)

# V82 STORAGE.  (CANNOT OVERLAY RONE OR VONE)   (11D) TWO SEPARAT LOCATIONS

V82FLAGS        EQUALS  VECTAB +6       # (1) FOR V82 BITS.
TFF             EQUALS  V82FLAGS +1     # I(2)
-TPER           EQUALS  TFF +2          # I(2)

HPERMIN         EQUALS  RANGE           # I(2) SET TO 300KFT FOR SR30.1
RPADTEM         EQUALS  HPERMIN +2      # I(2) PAD OR LANDING RADIUS FOR SR30.1
TSTART82        EQUALS  RPADTEM +2      # I(2) TEMP TIME STORAGE VOR V82.

# VARIOUS DISPLAY REGISTERS                     (6D) NOUN 84; P76
```

```
DELVOV          ERASE   +5D             # (6)

# ALIGNMENT PLANETARY -- INERTIAL TRANSFORMATION STORAGE.       (18D)

#               UNSHARED WHILE LM ON LUNAR SURFACE.

GSAV            ERASE   +17D            # I(6)
YNBSAV          EQUALS  GSAV +6         # I(6)
ZNBSAV          EQUALS  YNBSAV +6       # I(6)

# KALCMANU STORAGE, CAN OVERLAY GSAV.           (18D)

MFS             EQUALS  GSAV            # I(18)
MFI             EQUALS  MFS             # I
KEL             EQUALS  MFS             # I(18)
E01             EQUALS  MFS             # I(6)
E02             EQUALS  E01 +6          # I(6)

# LR VEL BEAM VECTORS.                          (26D)

# CAN OVERLAY GSAV WITH CARE, USED DURING POWERED DESCENT ONLY.

VZBEAMNB        EQUALS  GSAV            # I(6) LR VELOCITY BEAMS IN NB COORDS.
VYBEAMNB        EQUALS  VZBEAMNB +6     # I(6)
VXBEAMNB        EQUALS  VYBEAMNB +6     # I(6) PRESERVE Z,Y,X ORDER

LRVTIME         =       VXBEAMNB +6     # B(2) LR
LRXCDU          =       LRVTIME +2      # B(1) LR
LRYCDU          =       LRXCDU +1       # B(1) LR
LRZCDU          =       LRYCDU +1       # B(1) LR
PIPTEM          =       LRZCDU +1       # B(3) LR

# P32-P35, P72-P75 STORAGE.                     (40D)

T1TOT2          ERASE   +1              # (2)   TIME FROM CSI TO CDH
T2TOT3          ERASE   +1              # (2)
ELEV            ERASE   +1              # (2)
UP1             ERASE   +5              # (6)
DELVEET1        ERASE   +5              # I(6)  DV CSI IN REF
DELVEET2        ERASE   +5              # I(6)  DV CSH IN REF
RACT1           ERASE   +5              # (6)   POS VEC OF ACTIVE AT CSI TIME
RACT2           ERASE   +5              # (6)   POS VEC OF ACTIVE AT CDH TIME
```

```
RTSR1/MU         ERASE   +1              # (2)   SQ ROOT 1/MU STORAGE
RTMU             ERASE   +1              # (2)   MU STORAGE

# (THE FOLLOWING ERASABLES OVERLAY PORTIONS OF THE PREVIOUS SECTION)

+MGA             EQUALS  T1TOT2          # (2) S-S + MID GIM ANGL TO DELVEET3

UNRM             EQUALS  UP1             # I(6) S-S

DVLOS            EQUALS  RACT1           # I(6) S-S DELTA VELOCITY, LOS COORD-DISPLAY
ULOS             EQUALS  RACT2           # I(6) S-S UNIT LINE OF SIGHT VECTOR

NOMTPI           EQUALS  RTSR1/MU        # (2) S-S NOMINAL TPI-TIME FOR RECYCLE

# SOME P30 STORAGE.                              (4D)

HAPO             EQUALS  RTSR1/MU        # I(2)
HPER             EQUALS  HAPO    +2      # I(2)


# SOME P38-P39,P78-79 STORAGE            #       (6D)

DELTAR           EQUALS  DVLOS           # I(2)
DELTTIME         EQUALS  DELTAR +2       # I(2) TIME REPRESENTATION OF DELTAR
TARGTIME         EQUALS  DELTTIME +2     # I(2) TINT MINUS DELTTIME

TINTSOI          EQUALS  DELTAR          # I(2) TIME OF INTERCEPT FOR SOI PHASE

# THE FOLLOWING ARE ERASABLE LOADS DURING A PERFORMANCE TEST.

TRANSM1          =       WRENDPOS        # E4,1400
ALFDK            =       TRANSM1 +18D

# ****** THE FOLLOWING SECTIONS OVERLAY V83 AND DISPLAY STORAGE ******

# V47 (R47) AGS INITIALIZATION PROGRAM STORAGE.  (OVERLAYS V83) (14D)

AGSBUFF          EQUALS  RANGE           # B(14D)
AGSBUFFE         EQUALS  AGSBUFF +13D    # ENDMARK
```

```
# R36 OUT-OF-PLANE RENDEZVOUS DISPLAY STORAGE.  (OVERLAYS V83)  (12D)

RPASS36         EQUALS  RONE            # I(6) S-S
UNP36           EQUALS  RPASS36 +6      # I(6) S-S

# S-BAND ANTENNA GIMBAL ANGLES.  DISPLAYED BY R05 (V64).  (OVERLAYS V83)        (10D)
#                       (OPERATES DURING P00 ONLY)
ALPHASB         EQUALS  RANGE           # B(2)DSP NOUN 51.  PITCH ANGLE.
BETASB          EQUALS  ALPHASB +2      # B(2)DSP NOUN 51.  YAW ANGLE.
RLM             EQUALS  BETASB +2       # I(6)S S/C POSITION VECTOR.

# **** USED IN S-BAND ANTENNA FOR LM ****

YAWANG          EQUALS  BETASB
PITCHANG        EQUALS  ALPHASB

# NOUN 56 DATA -- COMPUTED AND DISPLAYED BY VERB 85.

RR-AZ           EQUALS  PITCHANG        # I(2) ANGLE BETWEEN LOS AND X-Z PLANE
RR-ELEV         EQUALS  RR-AZ +2        # I(2) ANGLE BETWEEN LOS AND Y-Z PLANE

# R04 (V62) RADAR TEST STORAGE.
# R04 IS RESTRICTED TO P00.

RSTACK          EQUALS  RANGE           # B(8) BUFFER FOR R04 NOUNS.

# INITVEL STORAGE.  ALSO USED BY P31, P34, P35, P74, P75, P10, P11, MIDGIM, S40.1 AND S40.9.    (18D)

#               (POSSIBLY RINIT & VINIT CAN OVERLAY DELVEET1 & 2 ABOVE)

RINIT           ERASE   +5              # I(6) ACTIVE VEHICLE POSITION
VINIT           ERASE   +5              # I(6) ACTIVE VEHICLE VELOCITY
VIPRIME         ERASE   +5              # I(6) NEW VEL REQUIRED AT INITIAL RADIUS.

# VARIOUS DISPLAY REGISTERS.  BALLANGS          (3D)

FDAIX           ERASE                   # I(1)
FDAIY           ERASE                   # I(1)
FDAIZ           ERASE                   # I(1)

# P34-P35 STORAGE.  DOWNLINKED.                 (2D)

DELVTPF         ERASE   +1              # I(2) DELTA V FOR TPF

# SOME R04(V63)-R77 RADAR TEST STORAGE.         (6D)
```

```
RTSTDEX         ERASE                   # (1)
RTSTMAX         ERASE                   # (1)
RTSTBASE        ERASE                   # (1)
RTSTLOC         ERASE                   # (1)
RSTKLOC         =       RTSTLOC
RSAMPDT         ERASE                   # (1)
RFAILCNT        ERASE                   # (1)

# LPS20.1 STORAGE.                                    (12D)

LMPOS           EQUALS RTSTDEX          # I(6)TMP  STORAGE FOR LM POS. VECTOR.
LMVEL           EQUALS LMPOS +6         # I(6)TMP  STORAGE FOR LM VEL. VECTOR.

# INITVEL STORAGE.  ALSU USED BY P31,34,35,74,75,S40.1 AND DOWNLINKED.  (6D)

DELVEET3        EQUALS  LMVEL +6        # I(6) DELTA V IN INERTIAL COORDINATES


END-E4          EQUALS                  # FIRST UNUSED LOCATION IN E4

# SECOND DPS GUIDANCE (LUNAR LANDING)  (OVERLAY P32-35, INITVEL)        (14D)

VHORIZ          EQUALS  PIPTEM +3       # I(2) DISPLAY
ANGTERM         EQUALS  VHORIZ +2       # I(6) GUIDANCE
HBEAMNB         EQUALS  ANGTERM +6      # I(6) LANDING RADAR

# R12 DOWNLINK QUANTITIES                             (5D)

LRXCDUDL        EQUALS  /LAND/ +2       # B(1) LANDING RADAR DOWNLINK
LRYCDUDL        EQUALS  LRXCDUDL +1     # B(1) LANDING RADAR DOWNLINK
LRZCDUDL        EQUALS  LRYCDUDL +1     # B(1) LANDING RADAR DOWNLINK
LRVTIMDL        EQUALS  LRZCDUDL +1     # B(2) LANDING RADAR DOWNLINK

# ASCENT GUIDANCE FOR LUNAR LANDING                   (54D)

AT              EQUALS  PIPTEM +3       # I(2)TMP ENGINE DATA -- THRUST ACC*2(9)
VE              EQUALS  AT +2           # I(2)TMP EXHAUST VELOCITY * 2(7)M/CS.
TTO             EQUALS  VE +2           # I(2)TMP TAILOFF TIME * 2(17)CS.
TBUP            EQUALS  TTO +2          # I(2)TMP (M/MDOT) * 2(17)CS.
RDOTD           EQUALS  TBUP +2         # I(2)TMP TARGET VELOCITY COMPONENTS
YDOTD           EQUALS  RDOTD +2        # I(2)TMP SCALING IS 2(7)M/CS.
ZDOTD           EQUALS  YDOTD +2        # I(2)TMP

/R/MAG          EQUALS  ZDOTD +2        # I(2)TMP
LAXIS           EQUALS  /R/MAG +2       # I(6)TMP
```

```
ZAXIS1          =       UHZP
RDOT            =       HDOTDISP
YDOT            =       LAXIS +6        # I(2)TMP VEL. NORMAL TO REF. PLANE*2(-7)
ZDOT            EQUALS  YDOT +2         # I(2)TMP DOWN RANGE VEL * 2(-7)
GEFF            EQUALS  ZDOT +2         # I(2)TMP EFFECTIVE GRAVITY

# THESE TWO GROUPS OF ASCENT GUIDANCE ARE SPLIT BY THE ASCENT-DESCENT SERVICER SECTION FOLLOWING THIS SECTION

Y               EQUALS  /LAND/ +2       # I(2)TMP OUT-OF-PLANE DIST *2(24)M
DRDOT           EQUALS  Y +2            # I(2)TMP RDOTD - RDOT
DYDOT           EQUALS  DRDOT +2        # I(2)TMP YDOTD - YDOT
DZDOT           EQUALS  DYDOT +2        # I(2)TMP ZDOTD - ZDOT
PCONS           EQUALS  DZDOT +2        # I(2)TMP CONSTANT IN ATR EQUATION
YCONS           EQUALS  PCONS +2        # I(2)TMP CONSTANT IN ATY EQUATION
PRATE           EQUALS  YCONS +2        # I(2)TMP RATE COEFF. IN ATR EQUATION
YRATE           EQUALS  PRATE +2        # I(2)TMP RATE COEFF. IN ATY EQUATION
ATY             EQUALS  YRATE +2        # I(2)TMP OUT-OF-PLANE THRUST COMP. *2(9)
ATR             EQUALS  ATY +2          # I(2)TMP RADIAL THRUST COMP. * 2(9)
ATP             EQUALS  ATR +2          # I(2)TMP DOWN-RANGE THRUST COMP
YAW             EQUALS  ATP +2          # I(2)TMP
PITCH           EQUALS  YAW +2          # I(2)TMP

# SERVICER FOR LUNAR ASCENT AND DESCENT         (14D)

G(CSM)          EQUALS  GEFF    +2      # I(6)  FOR UPDATE OF COMMAND MODULE STATE
R(CSM)          EQUALS  R-OTHER         #       VECTORS BY LEM: ANALOGS OF GDT/2,
V(CSM)          EQUALS  V-OTHER         #       R, AND V, RESPECTIVELY OF THE CSM
WM              EQUALS  G(CSM) +6       # I(6)TMP -- LUNAR ROTATION VECTOR (SM)
/LAND/          EQUALS  WM +6           # B(2)  LUNAR RADIUS AT LANDING SITE
```

```
# EBANK-5 ASSIGNMENTS

                SETLOC  2400

# W-MATRIX.  ESSENTIALLY UNSHARABLE.            (162D)

W               ERASE   +161D
ENDW            EQUALS  W +162D

# ******* OVERLAY NUMBER 1 IN EBANK 5 *******

#               W-MATRIX PADLOADS               (124D)

TLAND           EQUALS  W               # I(2)  NOMINAL TIME OF LANDING
RBRFG           EQUALS  TLAND +2        # I(6) BRAKING
VBRFG           EQUALS  RBRFG +6        # I(6)      PHASE
ABRFG           EQUALS  VBRFG +6        # I(6)           TARGET
VBRFG*          EQUALS  ABRFG +6        # I(2)                PARAMETERS:
ABRFG*          EQUALS  VBRFG* +2       # I(2)                     HIGH
JBRFG*          EQUALS  ABRFG* +2       # I(2)                          GATE
GAINBRAK        EQUALS  JBRFG* +2       # B(2)
TCGFBRAK        EQUALS  GAINBRAK +2     # B(1)
TCGIBRAK        EQUALS  TCGFBRAK +1     # B(1)
RAPFG           EQUALS  TCGIBRAK +1     # I(6) APPROACH
VAPFG           EQUALS  RAPFG +6        # I(6)      PHASE
AAPFG           EQUALS  VAPFG +6        # I(6)           TARGET
VAPFG*          EQUALS  AAPFG +6        # I(2)                PARAMETERS:
AAPFG*          EQUALS  VAPFG* +2       # I(2)                     LOW
JAPFG*          EQUALS  AAPFG* +2       # I(2)                          GATE
GAINAPPR        EQUALS  JAPFG* +2       # B(2)
TCGFAPPR        EQUALS  GAINAPPR +2     # B(1)
TCGIAPPR        EQUALS  TCGFAPPR +1     # B(1)
VIGN            EQUALS  TCGIAPPR +1     # I(2)  DESIRED SPEED FOR IGNITION
RIGNX           EQUALS  VIGN +2         # I(2)  DESIRED `ALTITUDE' FOR IGNITION
RIGNZ           EQUALS  RIGNX +2        # I(2)  DESIRED GROUND RANGE FOR IGNITION
KIGNX/B4        EQUALS  RIGNZ +2        # I(2)
KIGNY/B8        EQUALS  KIGNX/B4 +2     # I(2)
KIGNV/B4        EQUALS  KIGNY/B8 +2     # I(2)
LOWCRIT         EQUALS  KIGNV/B4 +2     # B(1) (HIGHCRIT MUST FOLLOW LOWCRIT)
HIGHCRIT        EQUALS  LOWCRIT +1      # B(1)
V2FG            EQUALS  HIGHCRIT +1     # I(6) DESIRED VELOCITY FOR P65.
TAUVERT         EQUALS  V2FG +6         # I(2) TIME CONSTANT FOR P65 VEL. NULLING.
DELQFIX         EQUALS  TAUVERT +2      # I(2) LR ALTITUDE DATA REASONABLE PARM.
LRALPHA         EQUALS  DELQFIX +2      # B(1) POS1 X ROTATION          * MUST *
```

```
LRBETA1         EQUALS  LRALPHA +1      # B(1) POS1 Y ROTATION          *  BE  *
LRALPHA2        EQUALS  LRBETA1 +1      # B(1) POS2 X ROTATION          *  IN  *
LRBETA2         EQUALS  LRALPHA2 +1     # B(1) POS2 Y ROTATION          * ORDER*
LRVMAX          EQUALS  LRBETA2 +1      # B(1) LR VEL WEIGHTING FUNCTIONS
LRVF            EQUALS  LRVMAX +1       # B(1) LR VEL WEIGHTING FUNCTIONS
LRWVZ           EQUALS  LRVF +1         # B(1) LR VEL WEIGHTING FUNCTIONS
LRWVY           EQUALS  LRWVZ +1        # B(1) LR VEL WEIGHTING FUNCTIONS
LRWVX           EQUALS  LRWVY +1        # B(1) LR VEL WEIGHTING FUNCTIONS
LRWVFZ          EQUALS  LRWVX +1        # B(1) LR VEL WEIGHTING FUNCTIONS
LRWVFY          EQUALS  LRWVFZ +1       # B(1) LR VEL WEIGHTING FUNCTIONS
LRWVFX          EQUALS  LRWVFY +1       # B(1) LR VEL WEIGHTING FUNCTIONS
LRWVFF          EQUALS  LRWVFX +1       # B(1) LR VEL WEIGHTING FUNCTIONS

ABVEL*          EQUALS  BUF             # B(1) LR TEMP
VSELECT*        EQUALS  BUF +1          # B(1) LR TEMP

RODSCALE        EQUALS  LRWVFF +1       # I(1) CLICK SCALE FACTOR FOR ROD
TAUROD          EQUALS  RODSCALE +1     # I(2) TIME CONSTANT FOR R.O.D.
LAG/TAU         EQUALS  TAUROD +2       # I(2) LAG TIME DIVIDED BY TAUROD (P66)
MINFORCE        EQUALS  LAG/TAU +2      # I(2) MINIMUM FORCE P66 WILL COMMAND
MAXFORCE        EQUALS  MINFORCE +2     # I(2) MAXIMUM FORCE P66 WILL COMMAND.
ABTCOF          EQUALS  MAXFORCE +2     # I(16) COEFFICIENTS FOR ABORT TFI POLYS.
VMIN            EQUALS  ABTCOF +16D     # I(2) MINIMUM VELOCITY FOR ABORT INJ.
YLIM            EQUALS  VMIN +2         # I(2) MAXIMUM CROSS-RANGE DIST. IN ABORTS
ABTRDOT         EQUALS  YLIM +2         # I(2) DESIRED RADIAL VEL. FOR ABORTS.
COSTHET1        EQUALS  ABTRDOT +2      # I(2) COS CONE 1 ANGLE FOR ABORTS.
COSTHET2        EQUALS  COSTHET1 +2     # I(2) COS OF CONE 2 ANGLE FOR ABORTS.

# SOME VARIABLES FOR SECOND DPS GUIDANCE.       (34D)

CG              EQUALS  COSTHET2 +2     # I(18D) GUIDANCE
RANGEDSP        EQUALS  CG +18D         # B(2) DISPLAY
OUTOFPLN        EQUALS  RANGEDSP +2     # B(2) DISPLAY
R60VSAVE        EQUALS  OUTOFPLN +2     # I(6)TMP SAVES VALUE OF POINTVSM THRU R51
RGU             EQUALS  R60VSAVE +6     # I(6) UNSHARED FOR DOWNLINK
VBIAS           EQUALS  R60VSAVE        # I(6) PIPA BIAS EQUIV. VELOCITY VECTOR.
L*WCR*T         =       BUF
H*GHCR*T        =       BUF     +1

# ALIGNMENT/SYSTEST/CALCSMSC COMMON STORAGE     (36D)

XSM             EQUALS  ENDW            # B(6)
YSM             EQUALS  XSM     +6      # B(6)
ZSM             EQUALS  YSM     +6      # B(6)

XDC             EQUALS  ZSM     +6      # B(6)
YDC             EQUALS  XDC     +6      # B(6)
ZDC             EQUALS  YDC     +6      # B(6)
```

```
XNB             =       XDC
YNB             =       YDC
ZNB             =       ZDC

# OVERLAYS WITHIN ALIGNMENT/SYSTEST/CALCSMSC COMMON STORAGE     (4D)

-COSB           EQUALS  XSM +2          # (2)TMP
SINB            EQUALS  -COSB +2        # (2)TMP

# MORE OVERLAYS TO ALIGNMENT/SYSTEST (THESE ARE P52)    (6D)

LANDLAT         EQUALS  STARAD          # (2) LATTITUDE, LONGITUDE
LANDLONG        EQUALS  LANDLAT +2      # (2)   AND ALTITUDE
LANDALT         EQUALS  LANDLONG +2     # (2)   OF LANDING SITE

# ALIGNMENT/SYSTEST COMMON STORAGE.             (31D)

STARAD          EQUALS  ZDC +6          # I(18D)TMP
STAR            EQUALS  STARAD +18D     # I(6)
GCTR            EQUALS  STAR +6         # B(1)
OGC             EQUALS  GCTR +1         # I(2)
IGC             EQUALS  OGC +2          # I(2)
MGC             EQUALS  IGC +2          # I(2)

# P57 ALIGNMENT (OVERLAY OF ALIGNMENT/SYSTEST COMMON STORAGE)   (12D)

GACC            =       STARAD          # (6) SS
GOUT            =       STARAD +6       # (6) SS

# OVERLAYS WITHIN ALIGNMENT/SYSTEST COMMON STORAGE      (24D)

VEARTH          EQUALS  STARAD          # (6)TMP
VSUN            EQUALS  VEARTH +6       # (6)TMP
VMOON           EQUALS  VSUN +6         # (6)TMP
SAX             EQUALS  VMOON +6        # (6)TMP

# P50'S, R50'S Q STORES                         (2D)

QMIN            EQUALS  MGC +2          # B(1)TMP
QMAJ            EQUALS  QMIN +1         # B(1)TMP

# **** USED IN P50S **** (SCATTERED OVERLAYS)

XSCI            EQUALS  STARAD
YSCI            EQUALS  XSCI +6
```

```
ZSCI            EQUALS  YSCI
CULTRIX         EQUALS  VEARTH          # VEARTH, VSUN, VMOON
VEC1            EQUALS  STARAD +12D
VEC2            EQUALS  STAR

# ALIGNMENT STORAGE.                                (23D)

OGCT            EQUALS  QMAJ +1         # I(6)
BESTI           EQUALS  OGCT +6         # I(1)
BESTJ           EQUALS  BESTI +1
STARIND         EQUALS  BESTJ +1
# RETAIN THE ORDER OF STARSAV1 TO STARSAV2 +5 FOR DOWNLINK PURPOSES
STARSAV1        EQUALS  STARIND +1      # I(6)
STARSAV2        EQUALS  STARSAV1 +6     # I(6)
TALIGN          EQUALS  STARSAV2 +6     # B(2) TIME OF IMU ALIGNMENT (DOWNLINKED)

# P32-35 + SERVICER

RTX1            EQUALS  TALIGN +2       # I(1) X1       -2 EARTH, -10 MOON
RTX2            EQUALS  RTX1 +1         # I(1) X2       0 EARTH, 2 MOON

ZPRIME          =       22D
PDA             =       22D
COSTH           =       16D
SINTH           =       18D
THETA           =       20D
STARM           =       32D
```

```
# ******* OVERLAY NUMBER 2 IN EBANK 5 *******

# CONICS ROUTINE STORAGE.                         (85D)

DELX            EQUALS   ENDW            # I(2)TMP
DELT            EQUALS   DELX +2         # I(2)TMP
URRECT          EQUALS   DELT +2         # I(6)TMP
RCNORM          EQUALS   34D             # I(2)TMP
XPREV           EQUALS   XKEP            # I(2)TMP
R1VEC           EQUALS   URRECT +6       # I(6)TMP
R2VEC           EQUALS   R1VEC +6        # I(6)TMP
TDESIRED        EQUALS   R2VEC +6        # I(2)TMP
GEOMSGN         EQUALS   TDESIRED +2     # I(1)TMP
UN              EQUALS   GEOMSGN +1      # I(6)TMP
VTARGTAG        EQUALS   UN +6           # I(1)TMP
VTARGET         EQUALS   VTARGTAG +1     # I(6)TMP
RTNLAMB         EQUALS   VTARGET +6      # I(1)TMP
U2              EQUALS   RTNLAMB +1      # I(6)TMP
MAGVEC2         EQUALS   U2 +6           # I(2)TMP
UR1             EQUALS   MAGVEC2 +2      # I(6)TMP
SNTH            EQUALS   UR1 +6          # I(2)TMP
CSTH            EQUALS   SNTH +2         # I(2)TMP
1-CSTH          EQUALS   CSTH +2         # I(2)TMP
CSTH-RHO        EQUALS   1-CSTH +2       # I(2)TMP
P               EQUALS   CSTH-RHO +2     # I(2)TMP
R1A             EQUALS   P +2            # I(2)TMP
RVEC            EQUALS   R1VEC           # I(6)TMP
VVEC            EQUALS   R1A +2          # I(6)TMP
RTNTT           EQUALS   RTNLAMB         # I(1)TMP
ECC             EQUALS   VVEC +6         # I(2)TMP
RTNTR           EQUALS   RTNLAMB         # I(1)TMP
RTNAPSE         EQUALS   RTNLAMB         # I(1)TMP
R2              EQUALS   MAGVEC2         # I(2)TMP
RTNPRM          EQUALS   ECC +2          # I(1)TMP
SGNRDOT         EQUALS   RTNPRM +1       # I(1)TMP
RDESIRED        EQUALS   SGNRDOT +1      # I(2)TMP
DELDEP          EQUALS   RDESIRED +2     # I(2)TMP
DEPREV          EQUALS   DELDEP +2       # I(2)TMP
TERRLAMB        EQUALS   DELDEP          # I(2)TMP
TPREV           EQUALS   DEPREV          # I(2)TMP
EPSILONL        EQUALS   DEPREV +2       # I(2)TMP
COGA            EQUALS   EPSILONL +2     # I(2)  COTAN OF INITIAL FLIGHT PATH ANGLE.
INDEP           EQUALS   COGA            #       USED BY SUBROUTINE `ITERATOR'.
```

```
# ******* OVERLAY NUMBER 3 IN EBANK 5 *******

# INCORP STORAGE.                               (18D)

ZI              EQUALS  ENDW            # I(18D)TMP

# INCORP/L SR22.3 STORAGE.                      (21D)

DELTAX          EQUALS  ZI +18D         # I(18)
VARIANCE        EQUALS  DELTAX +18D     # I(3)

# MEASUREMENT INCORPORATION -R22- STORAGE.      (49D)

GRP2SVQ         EQUALS  VARIANCE +3     # I(1)TMP QSAVE FOR RESTARTS
OMEGAM1         EQUALS  GRP2SVQ +1      # I(6)
OMEGAM2         EQUALS  OMEGAM1 +6      # I(6)
OMEGAM3         EQUALS  OMEGAM2 +6      # I(6)
HOLDW           EQUALS  OMEGAM3 +6      # I(18)
TDPOS           EQUALS  HOLDW +18D      # I(6)
TDVEL           EQUALS  TDPOS +6        # I(6)

TRIPA           EQUALS  DELTAX          # I(3)TMP
TEMPVAR         EQUALS  TRIPA +3        # I(3)TMP

# INCORPORATION/INTEGRATION Q STORAGE.          (1D)

EGRESS          EQUALS  TDVEL +6        # I(1)

# P30/P31 STORAGE.                              (1D) AND ONE OVERLAY

P30EXIT         EQUALS  EGRESS +1       # B(1)TMP

ORIGIN          EQUALS  P30EXIT         # I(1)TMP INTEX DURING INITVEL.
```

```
# SYSTEM TEST ERASABLES.  CAN OVERLAY W MATRIX. (127D)

# ******* OVERLAY NUMBER 0 IN EBANK 5 *******

AZIMUTH          EQUALS   W                # 2
LATITUDE         EQUALS   AZIMUTH +2       # 2
ERVECTOR         EQUALS   LATITUDE +2      # 6
LENGTHOT         EQUALS   ERVECTOR +6      # 1
LOSVEC           EQUALS   LENGTHOT +1      # 6
NDXCTR           EQUALS   LOSVEC +1        # 1
PIPINDEX         EQUALS   NDXCTR +1        # 1
POSITON          EQUALS   PIPINDEX +1      # 1
QPLACE           EQUALS   POSITON +1       # 1
QPLACES          EQUALS   QPLACE +1        # 1
SOUTHDR          EQUALS   QPLACES +1       # 7
TEMPTIME         EQUALS   SOUTHDR +7       # 2
TMARK            EQUALS   TEMPTIME +2      # 2
GENPL            EQUALS   TMARK +2
CDUTIMEI         =        GENPL
CDUTIMEF         =        GENPL +2
CDUDANG          =        GENPL +4
CDUREADF         =        GENPL +5
CDUREADI         =        GENPL +6
CDULIMIT         =        GENPL +7

TEMPADD          =        GENPL +4
TEMP             =        GENPL +5
NOBITS           =        GENPL +6
CHAN             =        GENPL +7

LOS1             =        GENPL +8D
LOS2             =        GENPL +14D

CALCDIR          EQUALS   GENPL +20D
CDUFLAG          EQUALS   GENPL +21D
GYTOBETQ         EQUALS   GENPL +22D
OPTNREG          EQUALS   GENPL +23D
SAVE             EQUALS   GENPL +24D       # THREE ONSEC LOC
SFCONST1         EQUALS   GENPL +27D
TIMER            EQUALS   GENPL +28D

DATAPL           EQUALS   GENPL +30D
RDSP             EQUALS   GENPL            # FIX LATER     POSSIBLY KEEP1
MASKREG          EQUALS   GENPL +64D
CDUNDX           EQUALS   GENPL +66D
RESULTCT         EQUALS   GENPL +67D
COUNTPL          EQUALS   GENPL +70D

CDUANG           EQUALS   GENPL +71D
```

```
AINLA           =       GENPL           # 110 DEC OR 156 OCT LOCATIONS

WANGO           EQUALS  AINLA           # VERT ERATE
WANGI           EQUALS  AINLA +2D       # HORIZONTAL ERATE
WANGT           EQUALS  AINLA +4D       # T
TORQNDX         =       WANGT
DRIFTT          EQUALS  AINLA +6D
ALX1S           EQUALS  AINLA +8D
CMPX1           EQUALS  AINLA +9D       # IND
ALK             EQUALS  AINLA +10D      # GAINS
VLAUNS          EQUALS  AINLA +22D
WPLATO          EQUALS  AINLA +24D
INTY            EQUALS  AINLA +28D      # SOUTH  IP INTE
ANGZ            EQUALS  AINLA +30D      # EAST A IS
INTZ            EQUALS  AINLA +32D      # EAST P P I
ANGY            EQUALS  AINLA +34D      # SOUTH
ANGX            EQUALS  AINLA +36D      # VE
DRIFTO          EQUALS  AINLA +38D      # VERT
DRIFTI          EQUALS  AINLA +40D      # SOU
VLAUN           EQUALS  AINLA +44D
ACCWD           EQUALS  AINLA +46D
POSNV           EQUALS  AINLA +52D
DPIPAY          EQUALS  AINLA +54D      # SOUTH
DPIPAZ          EQUALS  AINLA +58D      # NORTH  IP INCREMENT
ALTIM           EQUALS  AINLA +60D
ALTIMS          EQUALS  AINLA +61D      # INDEX
ALDK            EQUALS  AINLA +62D      # TIME  ONSTAN
DELM            EQUALS  AINLA +76D
WPLATI          EQUALS  AINLA +84D
GEOCOMPS        EQUALS  AINLA +86D
ERCOMP          EQUALS  AINLA +87D
ZERONDX         EQUALS  AINLA +93D

THETAN          =       ALK +4
FILDELV         EQUALS  THETAN +6       # AGS ALIGNMENT STORAGE
INTVEC          EQUALS  FILDELV +2
1SECXT          =       AINLA +94D
ASECXT          =       AINLA +95D
PERFDLAY        EQUALS  AINLA +96D      # B(2) DELAY TIME BEF. START DRIFT MEASURE
OVFLOWCK        EQUALS  AINLA +98D      # (1) SET MEANS OVERFLOW IN IMU PERF TEST

END-E5          EQUALS  STARSAV2 +6     # *** FIRST FREE LOCATION IN E5 ***
```

```
# EBANK-6 ASSIGNMENTS.

                SETLOC  3000

# DAP PAD-LOADED DATA.                                  (10D)

# ALL OF THE FOLLOWING EXCEPT PITTIME AND ROLLTIME ARE INITIALIZED IN FRESH START TO PERMIT IMMEDIATE USE OF DAP.

HIASCENT        ERASE                   # (1) MASS AFTER STAGING, SCALE AT B16 KG.
ROLLTIME        ERASE                   # (1) TIME TO TRIM Z GIMBAL IN R03, CSEC.
PITTIME         ERASE                   # (1) TIME TO TRIM Y GIMBAL IN R03, CSEC.
DKTRAP          ERASE                   # (1) DAP STATE            (POSSIBLE 77001
DKOMEGAN        ERASE                   # (1)   ESTIMATOR PARA-      (VALUES 00012
DKKAOSN         ERASE                   # (1)     METERS FOR THE             00074
LMTRAP          ERASE                   # (1)       DOCKED AND               77001
LMOMEGAN        ERASE                   # (1)         LEM-ALONE CASES        00000
LMKAOSN         ERASE                   # (1)           RESPECTIVELY         00074
DKDB            ERASE                   # (1) WIDTH OF DEADBAND FOR DOCKED RCS
                                        #     AUTOPILOT (DB=1.4DEG IN FRESH START
                                        #     DEADBAND = PI/DKDB RAD.

# PADLOADS FOR INTITIALIZATION OF DAP BIAS ACCELERATION (AT P12 IGNITION)       (2D)

IGNAOSQ         ERASE                   # B(1)PL
IGNAOSR         ERASE                   # B(1)PL

# AXIS TRANSFORMATION MATRIX -- GIMBAL TO PILOT AXES:   (5D)

M11             ERASE                   # SCALED AT 1
M21             ERASE                   # SCALED AT 1
M31             ERASE
M22             ERASE                   # SCALED AT 1.
M32             ERASE                   # SCALED AT 1.

# ANGLE MEASUREMENTS

OMEGAP          ERASE   +4              # BODY-AXIS ROT. RATES SCALED AT PI/4 AND
OMEGAQ          EQUALS  OMEGAP +1       # BODY-AXIS ACCELERATIONS SCALED AT PI/8.
OMEGAR          EQUALS  OMEGAP +2
# RETAIN THE ORDER OF ALPHAQ AND ALPHAR FOR DOWNLINK PURPOSES.
ALPHAQ          EQUALS  OMEGAP +3
ALPHAR          EQUALS  OMEGAP +4
OMEGAU          ERASE   +1
OMEGAV          =       OMEGAU +1

TRAPEDP         ERASE   +5
TRAPEDQ         =       TRAPEDP +1
TRAPEDR         =       TRAPEDP +2
NPTRAPS         =       TRAPEDP +3
```

```
NQTRAPS         =       TRAPEDP +4
NRTRAPS         =       TRAPEDP +5
EDOTP           =       EDOT
EDOTQ           ERASE   +1
EDOTR           =       EDOTQ +1        # MANY SHAREING NAMES
QRATEDIF        EQUALS  EDOTQ           # ALTERNATIVE NAMES:
RRATEDIF        EQUALS  EDOTR           # DELETE WHEN NO. OF REFERENCES = 0

URATEDIF        EQUALS  OMEGAU
VRATEDIF        EQUALS  OMEGAV
OLDXFORP        ERASE   +2              # STORED CDU READINGS FOR STATE
OLDYFORP        EQUALS  OLDXFORP +1     # DERIVATIONS: SCALED AT PI RADIANS (2'S)
OLDZFORQ        EQUALS  OLDXFORP +2

# RATE-COMMAND AND MINIMUM IMPULSE MODES

CH31TEMP        ERASE
STIKSENS        ERASE
TCP             ERASE
DXERROR         ERASE   +5
DYERROR         EQUALS  DXERROR +2
DZERROR         EQUALS  DXERROR +4
PLAST           ERASE
QLAST           ERASE
RLAST           ERASE
TCQR            ERASE

# OTHER VARIABLES                               (5D)

OLDPMIN         ERASE                   # THESE THREE USED IN MIN IMPUSE MODE
OLDQRMIN        ERASE
TEMP31          EQUALS  DAPTEMP1

SAVEHAND        ERASE   +1
PERROR          ERASE
QERROR          EQUALS  DYERROR
RERROR          EQUALS  DZERROR

# JET STATE CHANGE VARIABLES -- TIME (TOFJTCHG), JET BITS WRITTEN NOW   (10D)
# (JTSONNOW), AND JET BITS WRITTEN AT T6 RUPT (JTSATCHG).

NXT6ADR         ERASE
T6NEXT          ERASE   +1
T6FURTHA        ERASE   +1
NEXTP           ERASE   +2
NEXTU           =       NEXTP +1
NEXTV           =       NEXTP +2
-2JETLIM        ERASE   +1              # RATE COMMAND 4-JET RATE DIFFERENCE LIMIT
-RATEDB         EQUALS  -2JETLIM +1     # AND RATE DEADBAND FOR ASCENT OR DESCENT

TARGETDB        EQUALS  -RATEDB         # MAN. CONTROL TARGET DB COMPLEMENT.

# ***Q,R AXIS ERASABLES ***             (3)
```

```
PBIT            EQUALS  BIT10
QRBIT           EQUALS  BIT11
UERROR          EQUALS  DAPTREG5        # U,V-AXES ATT ERROR FOR RCS CONTROL LAWS.
VERROR          =       UERROR +1
RETJADR         ERASE
TEMPNUM         EQUALS  DAPTEMP4
NUMBERT         EQUALS  DAPTEMP5
ROTINDEX        EQUALS  DAPTEMP6
ROTEMP1         EQUALS  DAPTEMP1
ROTEMP2         EQUALS  DAPTEMP2
POLYTEMP        EQUALS  DAPTEMP3
SENSETYP        ERASE
ABSTJ           EQUALS  DAPTEMP1        # ABS VALUE OF JET-FIRING TIME
ABSEDOTP        EQUALS  DAPTEMP1

DPSBURN         EQUALS  DAPTREG4        # USED WITH SNUFFBIT.  VERY TEMPORARY.
```

```
# TRIM GIMBAL CONTROL LAW ERASABLES:            (11D)

GTSTEMPS        EQUALS  DAPTEMP1        # GTS IS PART OF THE JASK.
SHFTFLAG        EQUALS  GTSTEMPS +2     # COUNT BITS FOR GTSQRT SHIFTING.
ININDEX         EQUALS  GTSTEMPS +5     # INDEX FOR SHIFT LOOP IN GTSQRT.

SAVESR          EQUALS  AXISCTR         # CANNOT BE A DAPTEMP -- GTS USES THEM ALL.

SCRATCH         EQUALS  GTSTEMPS +7     # ROOTCYCL ERASABLE.
HALFARG         EQUALS  GTSTEMPS +8D    # ROOTCYCL ERASABLE.

K2THETA         EQUALS  GTSTEMPS        # D,P., K*ERROR, NEGUSUM
KCENTRAL        EQUALS  GTSTEMPS +2     # S.P., K FROM KQ OR KRDAP, AT PI/2(8)
K2CNTRAL        EQUALS  GTSTEMPS +3     # D.P., GTS SCRATCH CELLS.
WCENTRAL        EQUALS  GTSTEMPS +4     # S.P., OMEGA, AT PI/4 RAD/SEC
ACENTRAL        EQUALS  GTSTEMPS +5     # S.P., ALPHA, AT PI/4 RAD/SEC(2)
DEL             EQUALS  GTSTEMPS +6     # S.P., SGN FUNCTION VALUE.
A2CNTRAL        EQUALS  GTSTEMPS +7     # D.P., GTS SCRATCH CELLS.
QRCNTR          EQUALS  GTSTEMPS +9D    # S.P., INDEX FOR GTS LOOP THROUGH Q,R AXES
FUNCTION        EQUALS  GTSTEMPS +10D   # D.P., ARGUMENT FOR GRSQRT, SCRATCH FOR GTS.

NEGUQ           ERASE   +2              # NEGATIVE OF Q-AXIS GIMBAL DRIVE.
                EQUALS  NEGUQ +1        # DEFINED AND USED ELSEWHERE.
NEGUR           EQUALS  NEGUQ +2        # NEGATIVE OF R-AXIS GIMBAL DRIVE.

KQ              ERASE   +2              # S.P., JERK TERM FOR GTS, AT PI/2(8)
AXISCTR         EQUALS  KQ +1
KRDAP           EQUALS  KQ +2           # .3 ACCDOTR SCALED AT PI/2(8)

ACCDOTQ         ERASE   +3              # Q-JERK SCALED AT PI/2(7) UNSIGNED
QACCDOT         EQUALS  ACCDOTQ +1      # Q-JERK SCALED AT PI/2(7) SIGNED
ACCDOTR         EQUALS  ACCDOTQ +2      # R-JERK SCALED AT PI/2(7) UNSIGNED
RACCDOT         EQUALS  ACCDOTQ +3      # R-JERK SCALED AT PI/2(7) SIGNED

QDIFF           EQUALS  QERROR          # ATTITUDE ERRORS:
RDIFF           EQUALS  RERROR          # SCALED AT PI RADIANS.

# TORQUE VECTOR RECONSTRUCTION VARIABLES:       (18D)

JETRATE         EQUALS  DAPTREG1
JETRATEQ        EQUALS  JETRATE +1      # THE LAST CONTROL SAMPLE PERIOD OF 100 MS.
JETRATER        EQUALS  JETRATE +2      # SCALED AT PI/4 RADIANS/SECOND

DOWNTORK        ERASE   +5              # ACCUMULATED JET TORQUE COMMANDED ABOUT
POSTORKP        EQUALS  DOWNTORK        #       +,-P, +,-U, +,-V RESPECTIVELY.
NETTOTKP        EQUALS  DOWNTORK +1     #       EMPLOYED EXCLUSIVELY FOR DOWNLIST.
POSTORKU        EQUALS  DOWNTORK +2     #       NOT INITIALIZED:  PERMITTED TO OVERFLOW.
```

```
NEGTORKU        EQUALS  DOWNTORK +3     #       SCALED AT 32 JET-SEC, OR ABOUT 2.0 JET-
POSTORKV        EQUALS  DOWNTORK +4     #       MSEC. PER BIT.
NEGTORKV        EQUALS  DOWNTORK +5

NO.PJETS        ERASE   +2
NO.UJETS        =       NO.PJETS +1
NO.VJETS        =       NO.UJETS +1
TJP             ERASE   +2
TJU             =       TJP +1
TJV             =       TJP +2

L,PVT-CG        ERASE
1JACC           ERASE   +4              # ACCELERATIONS DUE TO 1 JET TORQUING
1JACCQ          EQUALS  1JACC +1        # SCALED AT PI/4 RADIANS/SECOND
1JACCR          EQUALS  1JACC +2
1JACCU          EQUALS  1JACC +3        # FOR U,V-AXES THE SCALE FACTOR IS DOFF:
1JACCV          EQUALS  1JACC +4        # SCALED AT PI/2 RADIANS/SECOND (FOR ASC)

# ASCENT VARIABLES                              (10D)

SKIPU           ERASE   +1
SKIPV           =       SKIPU +1

# THE FOLLOWING LM DAP ERASABLES ARE ZEROED IN THE STARTDAP SECTION OF THE DAPIDLER PROGRAM AND THE COASTASC
# SECTION OF THE AOSTASK.  THE ORDER MUST BE PRESERVED FOR THE INDEXING METHODS WHICH ARE EMPLOYED IN THOSE
# SECTIONS AND ELSEWHERE.

AOSQ            ERASE   +5              # OFFSET ACC. ESTIMATES, UPDATED IN D.P.,
AOSR            EQUALS  AOSQ +2         # AND SCALED AT PI/2.
AOSU            EQUALS  AOSQ +4         # UV-AXES OFFSET ACC. FROMED BY VECTOR
AOSV            EQUALS  AOSQ +5         # ADDITION OF Q,R. AT PI/2 RAD/SEC(2).

AOSQTERM        ERASE   +1              # (.1-.05K)AOS
AOSRTERM        EQUALS  AOSQTERM +1     # SCALED AT PI/4 RADIANS/SECOND.

# FOR TJET LAW SUBROUTINES:                     (TEMPS ONLY)

#NUMBERT        EQUALS  DAPTEMP5        # DEFINED IN QRAXIS.
EDOTSQ          EQUALS  DAPTEMP1
ROTSENSE        EQUALS  DAPTEMP2
FIREFCT         EQUALS  DAPTEMP3        # LOOKED AT BY PAXIS.
TTOAXIS         EQUALS  DAPTEMP4
ADRSDIF2        EQUALS  DAPTEMP6
HOLDQ           EQUALS  DAPTREG1
ADRSDIF1        EQUALS  DAPTREG2
HH              EQUALS  DAPTREG3        # DOUBLE PRECISION.
# HH +1         EQUALS  DAPTREG4
E               EQUALS  DAPTREG6        # TIME SHARE WITH VERROR
EDOT            EQUALS  OMEGAV
```

```
# INPUT TO TJET LAW (PERMANENT ERASABLES).      (48D)

TJETU           =       TJU             # EQUATE NAMES.  INDEXED BY -1, 0, +1.
BLOCKTOP        ERASE   +47D
1/ANET1         =       BLOCKTOP +16D   # THESE 8 PARAMETERS ARE SET UP BY 1/ACCS
1/ANET2         =       1/ANET1 +1      # FOR MINIMUM JETS ABOUT THE U-AXIS WHEN
1/ACOAST        =       1/ANET1 +4      # EDOT IS POSITIVE.  TJETLAW INDEXES BY
ACCFCTZ1        =       1/ANET1 +6      # ADRSDIFF FROM THESE REGISTERS TO PICK UP
ACCFCTZ5        =       1/ANET1 +7      # PARAMETERS FOR THE PROPER AXIS, NUMBER
FIREDB          =       1/ANET1 +10D    # OF JETS AND SIGN OF EDOT.  THERE ARE 48
COASTDB         =       1/ANET1 +12D    # REGISTERS IN ALL IN THIS BLOCK.
AXISDIST        =       1/ANET1 +14D    # FOUR NOT REFERENCED (P-AXIS) ARE FILLED
                                        # IN BY THE FOLLOWING:
ACCSWU          =       BLOCKTOP        # SET BY 1/ACCS TO SHOW WHETHER MAXIMUM
ACCSWV          =       ACCSWU +1       # JETS ARE REQUIRED BECAUSE OF AOS.
FLAT            =       BLOCKTOP +6     # WIDTH OF MINIMUM IMPULSE ZONE.
ZONE3LIM        =       BLOCKTOP +7     # HEIGHT OF MINIMUM IMPULSE ZONE (AT 4 SEC.)

COEFFQ          ERASE   +1              # COEFFQ AND COEFFR ARE USED IN ROT-TOUV
COEFFR          EQUALS  COEFFQ +1       # TO REXOLVE Q,R COMPONENTS INTO U,V COMP.

# VARIABLES FOR GTS-QRAXIS CONTROL EXCHANGE.    (4)

ALLOWGTS        EQUALS  NEGUQ +1        # INSERT INTO UNUSED LOCATION
COTROLER        ERASE                   # INDICATES WHICH CONTROL SYSTEM TO USE.
QGIMTIMR        ERASE   +2              # Q-GIMBAL DRIVE ITMER, DECISECONDS.
INGTS           EQUALS  QGIMTIMR +1     # INDICATOR OF CURRENT GTS CONTROL.
RGIMTIMR        EQUALS  QGIMTIMR +2     # R-GIMBAL DRIVE TIMER, DECISECONDS.

# PLEASE RETAIN THE ORDER OF CDUXD THRU CDUZD FOR DOWNLINK PURPOSES.

# KALCMANU:DAP INTERFACE                        (9D)

CDUXD           ERASE   +2              # CDU DESIRED REGISTERS:
CDUYD           EQUALS  CDUXD +1        # SCALED AT PI RADIANS (180 DEGREES)
CDUZD           EQUALS  CDUXD +2        # (STORE IN 2'S COMPLEMENT)

DELCDUX         ERASE   +2              # NEGATIVE OF DESIRED 100MS CDU INCREMENT:
DELCDUY         EQUALS  DELCDUX +1      # SCALED AT PI RADIANS (180 DEGREES)
DELCDUZ         EQUALS  DELCDUX +2      # (STORE IN 2'S COMPLEMENT)

# RETAIN THE ORDER OF OMEGAPD TO OMEGARD FOR DOWNLINK PURPOSES.

OMEGAPD         ERASE   +2              # ATTITUDE MANEUVER DESIRED RATES:
OMEGAQD         EQUALS  OMEGAPD +1      # (NOT EXPLICITLY REFERENCED IN GTS CNTRL)
OMEGARD         EQUALS  OMEGAPD +2      # SCALED AT PI/4 RADIANS/SECOND

# KALCMANU STORAGE.                             (24D)

MIS             ERASE   +23D            # I(18D)
```

```
COF               EQUALS  MIS +18D        # I(6)

# KALCMANU STORAGE.                               (33D)
BCDU              ERASE   +30D            # B(3)
KSPNDX            EQUALS  BCDU +3         # B(1)
KDPNDX            EQUALS  KSPNDX +1       # B(1)

TMIS              EQUALS  KDPNDX +1       # I(18) MUST BE IN SAME BANK AS RCS DAP
COFSKEW           EQUALS  TMIS +18D       # I(6)  MUST BE IN THE SAME BANK AS RCS DAP
CAM               EQUALS  COFSKEW +6      # I(2)  MUST BE IN THE SAME BANK AS RCS DAP

AM                ERASE   +1              # I(2) THIS WAS ONCE IN E5 OVERLAYING OGC

# FIRST-ODER OVERLAYS IN KALCMANU                 (25D)
KV1               EQUALS  TMIS            # I(6)
MFISYM            EQUALS  TMIS            # I
TMFI              EQUALS  TMIS            # I
NCDU              EQUALS  TMIS            # B
NEXTIME           EQUALS  TMIS +3         # B
TTEMP             EQUALS  TMIS +4         # B
KV2               EQUALS  TMIS +6         # I(6)
BIASTEMP          EQUALS  TMIS +6         # B
KV3               EQUALS  TMIS +12D       # I(6)
OGF               EQUALS  TMIS +12D       # I

BRATE             EQUALS  COFSKEW         # B
IG                EQUALS  COFSKEW         # I

TM                EQUALS  CAM             # B

# SECOND-ORDER OVERLAYS IN KALCMANU               (24D)

K1                =       KV1
K2                =       KV2
K3                =       KV3
P21               EQUALS  KV1             # I(2)
D21               EQUALS  KV1 +2          # I(2)
G21               EQUALS  KV1 +4          # I(2)
C2SQP             EQUALS  KV2             # I(2)
C2SQM             EQUALS  KV2 +2          # I(2)
C2PP              EQUALS  KV2 +4          # I(2)
C2MP              EQUALS  KV3             # I(2)
C1PP              EQUALS  KV3 +2          # I(2)
C1MP              EQUALS  KV3 +4          # I(2)
```

```
VECQTEMP        =       COFSKEW

DCDU            =       CDUXD
DELDCDU         =       DELCDUX
DELDCDU1        =       DELCDUY
DELDCDU2        =       DELCDUZ

# *  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *

# STORAGE FOR FINDCDUW

# OVERLAYING KALCMANU STORAGE:                    (26D)

ECDUW           EQUALS  MIS
ECDUWUSR        EQUALS  ECDUW           # B(1)TMP
QCDUWUSR        EQUALS  ECDUWUSR +1     # I(1)TMP
NDXCDUW         EQUALS  QCDUWUSR +1     # B(1)TMP
FLAGOODW        EQUALS  NDXCDUW +1      # B(1)TMP
FLPAUTNO        EQUALS  FLAGOODW +1     # B(1)TMP
UNFC/2          EQUALS  FLPAUTNO +1     # I(6)IN
UNWC/2          EQUALS  UNFC/2 +6       # I(6)IN
UNFV/2          EQUALS  UNWC/2 +6       # I(6) S-S
UNFVX/2         =       UNFV/2
UNFVY/2         =       UNFV/2 +2
UNFVZ/2         =       UNFV/2 +4
-DELGMB         EQUALS  UNFV/2 +6       # B(3)TMP

# DEFINED IN THE WORK AREA:                       (18D)

UNX/2           =       0
UNY/2           =       6
UNZ/2           =       14

# END OF FINDCDUW ERASABLES

# *  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *

# THE FOLLOWING ARE THE DAP REPLACEMENTS FOR THE ITEMPS AND RUPTREGS, NEEDED BECAUSE DAP IS NOW A TOB,JASK,JAB,TOSK
# ... ANYWAY, THE DAP CAN NOW BE INTERRUPTED.   (18D)

DAPTEMP1        ERASE   +17D
DAPTEMP2        EQUALS  DAPTEMP1 +1
DAPTEMP3        EQUALS  DAPTEMP1 +2
DAPTEMP4        EQUALS  DAPTEMP1 +3
DAPTEMP5        EQUALS  DAPTEMP1 +4
DAPTEMP6        EQUALS  DAPTEMP1 +5
DAPTREG1        EQUALS  DAPTEMP1 +6
DAPTREG2        EQUALS  DAPTEMP1 +7
DAPTREG3        EQUALS  DAPTEMP1 +8D
```

```
DAPTREG4        EQUALS  DAPTEMP1 +9D
DAPTREG5        EQUALS  DAPTEMP1 +10D
DAPTREG6        EQUALS  DAPTEMP1 +11D

DAPARUPT        EQUALS  DAPTEMP1 +12D
DAPLRUPT        EQUALS  DAPARUPT +1
DAPBQRPT        EQUALS  DAPARUPT +2
DAPZRUPT        EQUALS  DAPARUPT +4
                                        # (DAPZRUPT IS ALSO A JASK-IN-PROGRESS FLAG)

# NEEDLER (ATTITUDE ERROR EIGHT BALL DISPLAY) STORAGE.  (6D)

T5TEMP          EQUALS  ITEMP1
DINDX           EQUALS  ITEMP3
AK              ERASE   +2              # NEEDLER ATTITUDE INPUTS, SCALED AT 180
AK1             EQUALS  AK +1           # DEGREES.  P,Q,R AXES IN AK,AK1,AK2.
AK2             EQUALS  AK +2

EDRIVEX         ERASE   +2              # NEEDLER DISPLAY REGS AT 1800 DEGREES.
EDRIVEY         EQUALS  EDRIVEX +1      # SO THAT 384 BITS REPRESENT 42 3/16 DEGREES.
EDRIVEZ         EQUALS  EDRIVEX +2

# DOCKED JET INHIBITION COUNTERS                (3D)

PJETCTR         ERASE   +2
UJETCTR         EQUALS  PJETCTR +1
VJETCTR         EQUALS  PJETCTR +2

END-E6          EQUALS  VJETCTR
```

```
# EBANK-7 ASSIGNMENTS

                SETLOC  3400

# P35 CONSTANTS.        -- PAD LOADED --        (4D)

ATIGINC         ERASE   +1              # B(2)PL        *MUST BE AT 1400 FOR SYSTEMSTEST
PTIGINC         ERASE   +1              # B(2)PL

# AOTMARK STORAGE.      -- PAD LOADED --        (12D)

AOTAZ           ERASE   +5              # B(6)PL
AOTEL           ERASE   +5              # B(6)PL

# LANDING RADAR         -- PAD LOADED --        (2D)

LRHMAX          ERASE                   # B(1)
LRWH            ERASE                   # B(1)

# THROTTLE STORAGE.     -- PAD LOADED --        (1D)

ZOOMTIME        ERASE                   # B(1)PL TIME OF DPS THROTTLE-UP COMMAND

# P63 AND P64 CONSTANTS.        -- PAD LOADED --        (4D)

TENDBRAK        ERASE                   # B(1)  LANDING PHASE SWITCHING CRITERION.
TENDAPPR        ERASE                   # B(1)  LANDING PHASE SWITCHING CRITERION.
DELTTFAP        ERASE                   # B(1)  INCREMENT ADDED TO TTF/8 WHEN
                                        #        SWITCHING FROM P63 TO P64.
LEADTIME        ERASE                   # B(1)  TIME INCREMENT SPECIFYING HOW MUCH
                                        #        GUIDANCE IS PROJECTED FORWARD

# LANDING RADAR.        -- PAD LOADED --        (2D)

RPCRTIME        ERASE                   # B(1)  REPOSITIONING CRITERION (TIME)
RPCRTQSW        ERASE                   # B(1)  REPOSITIONING CRITERION (ANGLE)

# ASTEER.               -- PAD LOADED --        (2D)

TNEWA           ERASE   +1              # I(2)PL LAMBERT CYCLE PERIOD.

# P22 STORAGE -- OVERLAYS LANDING PADLOADS --   (5D)
```

```
REPOSCNT        EQUALS  TENDBRAK        # B(1)TMP COUNTS NUMBER OF PASSES THROUGH
                                        #         REPOSITION ROUTINE.
REPOSTM         EQUALS  REPOSCNT +1     # I(2)TMP PRESENT TIME PLUS INCREMENTS OF
                                        #         TEN SECONDS.
DELTATM         EQUALS  REPOSTM +2      # I(2)TMP TIME INTERVAL FOR RUNNING
                                        #         DESIGNATE TASK.

# *** RETAIN THE ORDER OF DELVSLV, TIG, RTARG, DELLT4 FOR UPDATE. ***

# P32-35 P72-75 STORAGE.                            (6D)

DELVLVC         ERASE   +5              # I(6) DELTA VELOCITY -- LOCAL VERTICAL COO
DELVSLV         =       DELVLVC         # (TEMP STORAGE OF SAME VECTOR)    -RDINATE

# P30-P40 INTERFACE UNSHARED.                       (2D)

TIG             ERASE   +1              # B(2)

# INITVEL STORAGE.  ALSO USED BY P34,35,74,75,10,11 OTHERS      (8D)

RTARG           ERASE   +5              # I(6) TARGET VECTOR
DELLT4          ERASE   +1              # I(2) TIME DIFFERENCE

# P30-P40 INTERFACE UNSHARED.                       (3D)

TTOGO           ERASE   +1              # B(2)
TFI             EQUALS  TTOGO
WHICH           ERASE                   # B(1)

# *** R21 ***                                       (1D)

LOSCOUNT        ERASE                   # B(1)

# L SR22.3 (RENDEZVOUS NAVIGATION) STORAGE.                 (4D)

# RETAIN THE ORDER OF AIG TO TRKMKCNT FOR DOWNLINK PURPOSES.

AIG             ERASE                   # B(1)OUT GIMBAL ANGLES
AMG             ERASE                   # B(1)OUT (MUST BE
AOG             ERASE                   # B(1)OUT  CONSECUTIVE)

TRKMKCNT        ERASE                   # B(1)TMP TEMPORARY MARK STORAGE.
MARKCTR         =       TRKMKCNT
```

```
# P32-P35, P72-P75 STORAGE.  -- PERMANENT --    (6)

NORMEX          ERASE                   # B(1)PRM SAVE FOR Q
QSAVED          ERASE                   # B(1)PRM SAVE FOR Q
RTRN            ERASE                   # B(1)PRM SAVE FOR Q
NN              ERASE   +1              # B(2)
SUBEXIT         ERASE                   # B(1)PRM SAVE Q

E7OVERLA        EQUALS                  # START OF E7 OVERLAYS
WHOCARES        EQUALS  E7OVERLA        # DUMMY FOR EBANK INSENSITIVE 2CADRS.

# LUNAR LANDING OVERLAYS                        (6D)

/AFC/           EQUALS  NORMEX          # B(2)TMP THROTTLE
FCODD           EQUALS  /AFC/ +2        # B(2)TMP THROTTLE
FP              EQUALS  FCODD +2        # B(2)TMP THROTTLE
```

```
# ******* OVERLAY NUMBER 0 IN EBANK 7 *******

# RENDEZVOUS GUIDANCE STORAGE --P32.....P35--   (89D)

TSTRT           EQUALS  DELDV           # MIDCOURSE START TIME
TDEC2           EQUALS  DELVCSI         # TEMP STORAGE FOR INTEGRATION TIME INPUT
KT              EQUALS  DELVTPI         # TEMP STORAGE FOR MIDCOURSE DELTA TIME
VACT1           ERASE   +5D             # VELOCITY VECTOR OF ACTIVE AT CSI TIME
RPASS1          ERASE   +5D             # POSITION VECTOR OF PASSIVE AT CSI TIME
VPASS1          ERASE   +5D             # VELOCITY VECTOR OF PASSIVE AT CSI TIME
VACT2           ERASE   +5D             # VELOCITY VECTOR OF ACTIVE AT CDH TIME
RPASS2          ERASE   +5D             # POSITION VECTOR OF PASSIVE AT CDH TIME
VPASS2          ERASE   +5D             # VELOCITY VECTOR OF PASSIVE AT CDH TIME
RACT3           ERASE   +5D             # POSITION VECTOR OF ACTIVE AT TPI TIME
VACT3           ERASE   +5D             # VELOCITY VECTOR OF ACTIVE AT TPI TIME
RPASS3          ERASE   +5D             # POSITION VECTOR OF PASSIVE AT TPI TIME
VPASS3          ERASE   +5D             # VELOCITY VECTOR OF PASSIVE AT TPI TIME
VACT4           ERASE   +5D             # VELOCITY VECTOR OF ACTIVE AT INTERCEPT
UNVEC           EQUALS  VACT3           # CDHMVR UNIT VECTOR TEMP STORAGE.
DELVCSI         ERASE   +1D             # THRUST VALUE AT CSI
DELVTPI         ERASE   +1D             # THRUST VALUE AT TPI OR MID
DELVMID         EQUALS  DELVTPI
DIFFALT         ERASE   +1D             # ALT DIFFERENT AT CDH
POSTCSI         ERASE   +1              # PERIGEE ALTITUDE AFTER CSI MANEUVER
POSTCDH         ERASE   +1              # PERIGEE ALTITUDE AFTER CDH MANEUVER
POSTTPI         ERASE   +1              # PERIGEE ALTITUDE AFTER TPI MANEUVER
LOOPCT          EQUALS  POSTTPI         # CSI NEWTON ITERATION COUNTER
HAFPA1          EQUALS  POSTCDH         # HALF PERIOD
GAMPREV         ERASE   +1              # PREVIOUS GAMMA
DVPREV          EQUALS  DELVTPI         # PREVIOUS DELVCSI
DELDV           ERASE   +1D
CSIALRM         ERASE   +1              # FIRST SOLUTION ALARM
VERBNOUN        ERASE
TITER           EQUALS  CSIALRM         # ITERATION COUNTER
RDOTV           ERASE   +1
VAPREC          EQUALS  VPASS1          # I(6) S-S PREC VEC FOR NOM TPI TIME (ACTIVE)
RAPREC          EQUALS  RPASS1          # I(6) S-S PREC VEC FOR NOM TPI TIME (ACTIVE)
VPPREC          EQUALS  VPASS2          # I(6) S-S PREC VEC FOR NOM TPI TIME (PASSIVE)
RPPREC          EQUALS  RPASS2          # I(6) S-S PREC VEC FOR NOM TPI TIME (PASSIVE)
DELEL           EQUALS  DELVTPI         # I(2) S-S
DELTEE          EQUALS  DELDV           # I(2) S-S
SECMAX          EQUALS  DELVCSI         # I(2) S-S MAX STOP SIZE FOR ROUTINE
DELTEEO         EQUALS  POSTTPI         # I(2) S-S BACK VALUES OF DELTA TIME
CENTANG         ERASE   +1              # I(2) CENTRAL ANGLE COVERED (TPI-TPF)
```

```
# SOME P47 STORAGE                                    (6D)

DELVIMU         ERASE   +5              # I(6)DSP NOUN 83 FOR P47 DELTA V (IMU)

# P30-P40 COMMON STORAGE.                             (3D)

TPASS4          ERASE   +1              # INTERCEPT TIME
QTEMP           ERASE                   # I(1)TMP COMMON RETURN SAVE REGISTER.

# P32,33,34 STORAGE.                                  (6D)

TCSI            ERASE   +1              # B(2)TMP CSI TIME IN CENTISECONDS
TTPI            ERASE   +1              # B(2)TMP TPI TIME IN CENTISECONDS
TTPIO           ERASE   +1              # B(2)TMP TTPI STORAGE FOR RECYCLE

# P30,P40 INTERFACE.                                  (21D)

RTIG            ERASE   +19D            # I(6)TMP
VTIG            EQUALS  RTIG +6         # I(6)TMP
DELVSIN         EQUALS  VTIG +6         # I(6)TMP
DELVSAB         EQUALS  DELVSIN +6      # I(2)TMP
VGDISP          =       DELVSAB

QTEMP1          ERASE                   # I(1)TMP HOLDS RETURN.
RGEXIT          EQUALS  QTEMP1          # SAVE Q
SAVQR52         EQUALS  QTEMP1

# INITVEL STORAGE.  (IN OVERLAY 0 AND OVERLAY 1.        (2D)
#       (CALLS LAMBERT, CONIC SUBROUTINES)

VTPRIME         EQUALS  VACT4           # TOTAL VELOCITY AT DESIRED RADIUS
ITCTR           EQUALS  RDOTV           # ITERATION COUNTER
COZY4           ERASE   +1              # COS OF ANGLE WHEN ROTATION STARTS
X1INPUT         EQUALS  DELDV           # X1 TEMP STORAGE
INTIME          EQUALS  GAMPREV         # TIME OF RINIT

# PERIAPO STORAGE.                                    (2D)

XXXALT          ERASE   +1              # RADIUS TO LAUNCH PAD OR LANDING SITE

END-IN/M        EQUALS  XXXALT +2       # NEXT AVAIL ERASABLE AFTER INITVEL/MIDGIM
```

```
# S40.1 STORAGE.                                    (12D)

UT              ERASE   +11D            # I(6) THRUST DIRECTION
VGTIG           EQUALS  UT +6           # I(6)OUT
VGPREV          =       VGTIG

# ASTEER STORAGE.                                   (22D)

VG              ERASE   +21D            # I(6)
RMAG            EQUALS  VG +6           # I(2)
MUASTEER        EQUALS  RMAG  +2        # I(2)
MU/A            EQUALS  MUASTEER +2     # I(2)
RTMAG           EQUALS  MU/A +2         # I(2)
R1C             EQUALS  RTMAG +2        # I(6)
SS              EQUALS  R1C +6          # I(2)

IC              =       DELVSIN
TIGSAVE         =       P21TIME
TIGSAVEP        =       SCAXIS
MUSCALE         =       SCAXIS +2

# P40 STORAGE.                                      (6D)

# F, MDOT, AND TDECAY MUST BE CONTIGUOUS FOR WLOAD

F               ERASE   +5              # I(2)TMP S40.1 GENERATES THIS FOR S40.3
MDOT            EQUALS  F +2            # I(2)TMP MASS CHNG RATE, KG/CS AT 2**3.
TDECAY          EQUALS  MDOT +2         # I(2)IN DELTA-T TAILOFF, (2**28)CS.
VEX             ERASE   +1              # I(2) EXHAUST VELOCITY FOR TGO COMPUTAT'N

# MIDTOAV1(2) STORAGE.  (CALLED BY P40,P41,P42)         (1D)

IRETURN1        ERASE                   # B(1) RETURN FROM MIDTOAV1 AND 2.
```

```
# ******* OVERLAY NUMBER 1 IN EBANK 7 *******

# INITVEL (CALLED BY P34,35,38,39,10,11,S40.9,S40.1)    (6D)

RTARG1          EQUALS  VACT1           # I(6)S TEMP STORAGE OF RTARG

# P35-P40 INTERFACE.                            (6D)

VPASS4          EQUALS  VPASS1          # I(6)TMP VELOCITY OF PASSIVE AT INTERCEPT

# INITVEL OVERLAYS RENDESVOUS GUIDANCE (LISTED IN OVERLAY 0)

# SOME P38-30,P78-79 STORAGE                            (2D)
TINT            EQUALS  TPASS4          # I(2) TIME OF INTERCEPT


# LAT-LONG TEMPORARIES.  CAN OVERLAY WITH S40.1         (3D)

ERADM           EQUALS  UT              # I(2)
INCORPEX        EQUALS  ERADM +2        # I(1)

# LRS24.1 STORAGE.  (CAN SHARE WITH P30'S)              (40D)

RLMSRCH         EQUALS  INCORPEX +1     # I(6)TMP LM POSITION VECTOR
VXRCM           EQUALS  RLMSRCH +6      # I(6)    CM V X R VECTOR
LOSDESRD        EQUALS  VXRCM +6        # I(6)    DESIRED LOS VECTOR
UXVECT          EQUALS  LOSDESRD +6     # I(6)    X-AXIS SRCH PATTERN COORDS
UYVECT          EQUALS  UXVECT +6       # I(6)    Y-AXIS SRCH PATTERN COORDS
DATAGOOD        EQUALS  UYVECT +6       # B(1)DSP FOR R1 -- ALL 1-S WHEN LOCKON
OMEGDISP        EQUALS  DATAGOOD +1     # B(2)    ANGLE OMEGA DISPLAYED IN R2
OMEGAD          =       OMEGDISP        #         PINBALL DEFINITION
NSRCHPNT        EQUALS  OMEGDISP +2     # B(1)TMP SEARCH PATTERN POINT COUNTER.
SAVLEMV         EQUALS  NSRCHPNT +1     # I(6)S-S SAVES LOSVEL
```

```
# ******* OVERLAY NUMBER 2 IN EBANK 7 *******

# INCORP STORAGE IN E7.                              (47D)

TX789           EQUALS  E7OVERLA        # I(6)
GAMMA           EQUALS  TX789 +6        # I(3)
OMEGA           EQUALS  GAMMA +3        # I(18)
BVECTOR         EQUALS  OMEGA +18D      # I(18)
DELTAQ          EQUALS  BVECTOR +18D    # I(2)

# AOTMARK STORAGE                                    (3D)

MARKCNTR        EQUALS  DELTAQ +2       # I(1)
XYMARK          EQUALS  MARKCNTR +1     # B(1)
MKDEX           EQUALS  XYMARK +1       # B(1)TMP INDEX FOR AOTMARK

# PLANET STORAGE                                     (8D)

PLANVEC         EQUALS  MKDEX +1        # (6) REFER VECTOR OF PLANET
TSIGHT          EQUALS  PLANVEC +6      # (2) TIME OF MARK OR EST TIME OF MARK

# LRS22.3 STORAGE.  (CAN SHARE WITH P30'S AND OVERLAY LRS24.1)  (30D)

LGRET           EQUALS  RLMSRCH         # I(1)TMP
RDRET           EQUALS  LGRET           # B(1) TEMP RETURN.
IGRET           EQUALS  LGRET           # B(1) TEMP RETURN.
MX              EQUALS  RDRET +1        # I(6)
MY              EQUALS  MX +6           # I(6)
MZ              EQUALS  MY +6           # I(6)
E0              EQUALS  MX              # I(2)
E1              EQUALS  MX +2           # I(2)
E2              EQUALS  MX +4           # I(2)
E3              EQUALS  E2 +2           # I(2)
SCALSHFT        EQUALS  MZ +6           # B(1) SCALE SHIFT FOR EARTH/MOON
RXZ             EQUALS  SCALSHFT +1     # I(2)
ULC             EQUALS  RXZ +2          # I(6)
SINTHETA        EQUALS  ULC +6          # I(2)

# ***** IN OVERLAY ONE *****

N49FLAG         EQUALS  RDOTMSAV        # B(1)S FLAG INDICATING V0649 RESPONSE

# LRS22.1 STORAGE.  (MUST NOT SHARE WITH P30'S)         (13D)

# (OUTPUTS ARE TO LRS22.3)
```

```
RRTRUN          EQUALS  SINTHETA +2     # B(2)OUT RR TRUNNION ANGLE
RRSHAFT         EQUALS  RRTRUN +2       # B(2)OUT RR SHAFT ANGLE
LRS22.1X        EQUALS  RRSHAFT +2      # B(1)TMP
RRBORSIT        EQUALS  LRS22.1X +1     # I(6)TMP RADAR BORESIGHT VECTOR.
RDOTMSAV        EQUALS  RRBORSIT +6     # B(2)S RR RANGE-RATE (FPS)

# LRS22.1 (SAME AS PREVIOUS SECTION) ALSO DOWNLINK FOR RR (R29)         (10D) CANNOT SHARE WITH L.A.D.

RDOTM           EQUALS  RDOTMSAV +2     # B(2)OUT RANGE-RATE READING
TANGNB          EQUALS  RDOTM +2        # B(2)TMP RR GIMBAL ANGLES
# RETAIN THE ORDER OF MKTIME TO RM FOR DOWNLINK PURPOSES
MKTIME          EQUALS  TANGNB +2       # B(2)OUT TIME OF RR READING
RM              EQUALS  MKTIME +2       # I(2)OUT RANGE READING
RANGRDOT        EQUALS  RM +2           # B(2) DOWNLINKED RAW RANGE AND RRATE

# R61LEM -- PREFERRED TRACKING ATTITUDE ROUTINE **IN OVERLAY ONE**
# (CALLED BY P20, R22LEM, LSR22.3)              (1D)

R65CNTR         EQUALS  RRBORSIT +5     # B(1)SS COUNT NUMBER OF TIMES PREFERRED
                                        #        TRACKING ROUTINE IS TO CYCLE
WHCHREAD        EQUALS  R65CNTR         # TELLS WHICH RR DATA TRIGGERED N49 DISPLAY

# P21 STORAGE                                   (2D)

P21TIME         EQUALS  RANGRDOT +2     # I(2)TMP

# KALCMANU, VECPOINT STORAGE.  CALLED BY R63, R61, R65.         (12D)

SCAXIS          EQUALS  P21TIME +2      # I(6)
POINTVSM        EQUALS  SCAXIS +6       # I(6)
```

```
# ******* OVERLAY NUMBER 3 IN EBANK 7 *******

# SERVICER STORAGE                                      (6D)

ABVEL           EQUALS  E7OVERLA        # B(2) DISPLAY
HDOTDISP        EQUALS  ABVEL +2        # B(2) DISPLAY
TTFDISP         EQUALS  HDOTDISP +2     # B(2) DISPLAY

# BURN PROG STORAGE.                                    (2D)

SAVET-30        EQUALS  TTFDISP +2      # B(2)TMP TIG-30 RESTART

# SERVICER STORAGE.                                     (69D)

VGBODY          EQUALS  SAVET-30 +2     # B(6)OUT SET BY S41.1 VG LEM, SC.COORDS
DELVCTL         =       VGBODY
DVTOTAL         EQUALS  VGBODY +6       # B(2) DISPLAY NOUN
GOBLTIME        EQUALS  DVTOTAL +2      # B(2) NOMINAL TIG FOR CALC. OF GOBLATE.
ABDVCONV        EQUALS  GOBLTIME +2     # I(2)
DVCNTR          EQUALS  ABDVCONV +2     # B(1)
TGO             EQUALS  DVCNTR +1       # B(2)
R               EQUALS  TGO +2          # I(6)
UNITGOBL        EQUALS  R               # I(6)
V               EQUALS  R +6
DELVREF         EQUALS  V               # I(6)
HCALC           EQUALS  DELVREF +6      # B(2) LR
UNIT/R/         EQUALS  HCALC +2        # I(6)

# (THE FOLLOWING SERVICER ERASABLES CAN BE SHARED WITH SECOND DPS GUIDANCE STORAGE)

RN1             EQUALS  UNIT/R/ +6      # B(6)
VN1             EQUALS  RN1 +6          # I(6)          ( IN ORDER )
PIPTIME1        EQUALS  VN1 +6          # B(2)          (    FOR   )
GDT1/2          EQUALS  PIPTIME1 +2     # I(6)          (   COPY   )
MASS1           EQUALS  GDT1/2 +6       # I(2)          (   CYCLE  )
R1S             EQUALS  MASS1 +2        # I(6)
V1S             EQUALS  R1S +6          # I(6)

# ALIGNMENT/S40.2,3 COMMON STORAGE.                     (18D)

XSMD            EQUALS  V1S +6          # I(6)
YSMD            EQUALS  XSMD +6         # I(6)
ZSMD            EQUALS  YSMD +6         # I(6)

XSCREF          =       XSMD
YSCREF          =       YSMD
```

```
ZSCREF          =       ZSMD

END-ALIG        EQUALS  ZSMD +6         # NEXT AVAIL ERASABLE AFTER ALIGN/S40.2,3

# ****** P22 ******                             (24D)

RSUBL           EQUALS  END-ALIG        # I(6)S-S       LM POSITION VECTOR
UCSM            EQUALS  RSUBL +6        # I(6)S-S       VECTOR U
NEWVEL          EQUALS  UCSM +6         # I(6)S-S       TERMINAL VELOCITY VECTOR
NEWPOS          EQUALS  NEWVEL +6       # I(6)S-S       TERMINAL POSITION VECTOR
LNCHTM          EQUALS  NEWPOS +6       # I(2)S-S       EST. LAUNCH TIME FOR LEM
TRANSTM         EQUALS  LNCHTM +2       # I(2)S-S       TRANSFER TIME
NCSMVEL         EQUALS  TRANSTM +2      # I(6)S-S       NEW CSM VELOCITY

# ****** P21 ******                             (18D)

P21ORIG         =       DISPDEX
P21BASER        EQUALS  RLMSRCH         # I(6)TMP
P21BASEV        EQUALS  P21BASER +6     # I(6)TMP
P21VEL          EQUALS  P21BASEV +6     # I(2)TMP       *** NOUN 91 ***
P21GAM          EQUALS  P21VEL +2       # I(2)TMP       *** NOUN 91 ***
P21ALT          EQUALS  P21GAM +2       # I(2)TMP       *** NOUN 91 ***
```

```
# ******* OVERLAY NUMBER 4 IN EBANK 7 *******

# VARIABLES FOR SECOND DPS GUIDANCE (THE LUNAR LANDING)          (18D)

# THESE ERASABLES MAY BE SHARED WITH CARE

OURTEMPS        =       RN1             # OVERLAY LAST PART OF SERVICER
LANDTEMP        =       OURTEMPS        # B(6)  GUIDANCE
TTF/8TMP        =       LANDTEMP +6     # B(2)  GUIDANCE
ELINCR          =       TTF/8TMP +2     # B(2)  GUIDANCE
AZINCR          =       ELINCR +2       # B(2)  GUIDANCE
KEEP-2          =       AZINCR +2       # B(2)  TP PREVENT PIPTIME1 OVERLAY
TABLTTF         =       KEEP-2 +2       # B(2)  GUIDANCE
TPIPOLD         =       TABLTTF +9D     # B(2)  GUIDANCE
E2DPS           EQUALS  OURPERMS

# THESE ERASABLES MUST NOT OVERLAY GOBLTIME OR SERVICER

PIFPSET         =       XSMD            # B(1)  THROTTLE
RTNHOLD         =       PIFPSET +1      # B(1)  THROTTLE
FWEIGHT         =       RTNHOLD +1      # B(2)  THROTTLE
PIF             =       FWEIGHT +2      # B(2)  THROTTLE
PSEUDO55        =       PIF +2          # B(1)  THROTTLE DOWNLINK
FC              =       PSEUDO55 +1     # B(2)  THROTTLE
TTHROT          =       FC +2           # B(1)  THROTTLE
FCOLD           =       TTHROT +1       # B(1)  THROTTLE

# THESE ERASABLES SHOULD NOT BE SHARED DURING P63, P64, P65, P66, P67

OURPERMS        =       FCOLD +1        # MUSTN'T OVERLAY OURTEMPS OR SERVICER
WCHPHOLD        =       OURPERMS        # B(1)  GUIDANCE
FILLER          =       WCHPHOLD +1
FLPASSO         =       FILLER +1       # B(1)  GUIDANCE
TPIP            =       FLPASSO +1      # B(2)
VGU             =       TPIP +2         # B(6)  GUIDANCE
LAND            =       VGU +6          # B(6)  GUIDANCE        CONTIGUOUS
TTF/8           =       LAND +6         # B(2)  GUIDANCE        CONTIGUOUS
ELIDUMMY        =       TTF/8 +2        # (1)   DUMMY FOR ELINCR1
AZIDUMMY        =       ELIDUMMY +1     # (1)   DUMMY FOR AZINCR1
ZERDUMMY        =       AZIDUMMY +1     # (1)   DUMMY FOR ZERLINA
ELVDUMMY        =       ZERDUMMY +1     # (1)   DUMMY FOR ELVIRA
LRADRET         =       ELVDUMMY +1     # B(1)  LR
VSELECT         =       LRADRET +1      # B(1)  LR
VMEAS           =       VSELECT +1      # B(2)  LR
HMEAS           =       VMEAS +2        # B(2)  LR
VN2             =       HMEAS +2        # B(6)  LR
```

```
GNUR            =       VN2             # B(6)  LR
GNUV            =       VN2             # B(6)  LR
LRADRET1        =       VN2             # B(1)  LR
DELTAH          =       VN2 +6          # B(2)  DISPLAY
FUNNYDSP        =       DELTAH +2       # B(2)  DISPLAY
EOURPERM        EQUALS  FUNNYDSP +2     # NEXT AVAILABLE ERASABLE AFTER OURPERMS

# (ERASABLES WHICH OVERLAY THE ABOVE BLOCK)

VDGVERT         =       ELIDUMMY        # B(2)  P65,P66
NIGNLOOP        =       ZERDUMMY        # B(1)  IGNALG
NGUIDSUB        =       ELVDUMMY        # B(1)  IGNALG
WCHVERT         =       ELVDUMMY        # B(1)  P65,P66,P67
FUELNEED        =       FUNNYDSP        # B(1)  DISPLAY
TREDES          =       FUNNYDSP        # B(1)  DISPLAY
LOOKANGL        =       FUNNYDSP +1     # B(1)  DISPLAY

# ERASABLES CONVENIENTLY DEFINABLE IN THE WORK AREA

PROJ            =       18D             # I(2)  GUIDANCE
UNLRB/2         =       20D             # I(6)  GUIDANCE (DURING P64 ONLY)
UNLR/2          =       20D             # I(6)  GUIDANCE

# THE END OF THE LUNAR LANDING ERASABLES

# R12 (FOR LUNAR LANDING)                               (6D)

LRLCTR          EQUALS  EOURPERM        # B(1) LR DATA TEST
LRRCTR          EQUALS  LRLCTR +1       # B(1)
LRMCTR          EQUALS  LRRCTR +1       # B(1)
LRSCTR          EQUALS  LRMCTR +1       # B(1)
STILBADH        EQUALS  LRSCTR +1       # B(1)
STILBADV        EQUALS  STILBADH +1     # B(1)

# LANDING ANALOGS DISPLAY STORAGE.                      (40D)

LATVMETR        EQUALS  STILBADV +1     # B(1)PRM LATVEL MONITOR METER (AN ORDER)
FORVMETR        EQUALS  LATVMETR +1     # B(1)PRM FORVEL MONITOR METER (-ED PAIR)
LATVEL          EQUALS  FORVMETR +1     # B(1)PRM LATERAL VELOCITY (AN ORDER)
FORVEL          EQUALS  LATVEL +1       # B(1)PRM FORWARD VELOCITY (-ED PAIR)
TRAKLATV        EQUALS  FORVEL +1       # B(1)PRM MONITOR FLG 4 LATVEL (AN ORDER)
TRAKFWDV        EQUALS  TRAKLATV +1     # B(1)PRM MONIT. FLAG FOR FORVEL (ED PAIR)
VHY             EQUALS  TRAKFWDV +1     # B(1)PRM VHY=VMP.UHYP (AN ORDER)
```

```
        VHZ             EQUALS  VHY +1          # B(1)PRM VHZ=VMP.UHZP (-ED PAIR)
        VVECT           EQUALS  VHZ +1          # B(3)PRM UPDATED S.P. VELOCITY VECTOR
        ALTRATE         EQUALS  VVECT +3        # B(1)PRM ALTITUDE RATE IN BIT UNITS
        ALTSAVE         EQUALS  ALTRATE +1      # B(2)PRM ALTITUDE IN BIT UNITS
        LADQSAVE        EQUALS  ALTSAVE +2      # B(1)PARM SAVE Q IN LAND1SP
        DT              EQUALS  LADQSAVE +1     # B(1)PRM TIME 1 MINUS (PIPTIME +1)
        DALTRATE        EQUALS  DT +1           # B(1)PRM ALTITUDE RATE ERROR CORRECTION
        UHYP            EQUALS  DALTRATE +1     # B(6)PRM SM UNIT VECTOR
        QAXIS           =       UHYP
        UHZP            EQUALS  UHYP +6         # B(6)PRM SM UNIT VECTOR
        DELVS           EQUALS  UHZP +6         # B(6)PRM DELVS = WMXR
        ALTBITS         EQUALS  DELVS +6        # B(2)PRM ALTITUDE IN BIT UNITS. 2.34 FT/BIT
        RUNIT           EQUALS  ALTBITS +2      # B(3)PRM SM HALF-UNIT R VECTOR
        LASTLADW        EQUALS  RUNIT +2        # ONLY A TAG TO SIGNIFY LAST L.A.D. WORD

# P66 ERASABLES (R.O.D.)                        (1D)

        RODCOUNT        EQUALS  RUNIT +3

# P66 ERASABLES (R.O.D.)                        (14D)

        RODSCAL1        EQUALS  RM              # B(1)
        LASTTPIP        EQUALS  RODSCAL1 +1     # I(2)
        THISTPIP        EQUALS  LASTTPIP +2     # B(2)
        OLDPIPAX        EQUALS  THISTPIP +2     # B(1)
        OLDPIPAY        EQUALS  OLDPIPAX +1     # B(1)
        OLDPIPAZ        EQUALS  OLDPIPAY +1     # B(1)
        DELVROD         EQUALS  OLDPIPAZ +1     # B(6)

# NOUN 63 COMPONENT                             (2D)

        HCALC1          EQUALS  DELVROD +6      # I(2)
```

```
# ******* OVERLAY NUMBER 5 IN EBANK 7 *******

# ASCENT GUIDANCE ERASABLES.                    (21D)

RCO             EQUALS  END-ALIG        # I(2)TMP TARGET RADIUS AND OUT-OF-PLANE
YCO             EQUALS  RCO +2          # I(2)TMP DISTANCE, SCALED AT 2(24).
1/DV1           EQUALS  YCO +2          # B(2)TMP ATMAG
1/DV2           EQUALS  1/DV1 +2        # B(2)TMP ATMAG
1/DV3           EQUALS  1/DV2 +2        # B(2)TMP ATMAG
XRANGE          EQUALS  1/DV3 +2        # B(2)TMP
ENGOFFDT        EQUALS  XRANGE +2       # B(1)TMP
VGVECT          EQUALS  ENGOFFDT +1     # I(6)OUT VELOCITY-TO-BE-GAINED.
TXO             EQUALS  VGVECT +6       # I(2)TMP TIME AT WHICH X-AXIS OVERRIDE
                                        # IS ALLOWED.

# END OF THE ASCENT GUIDANCE ERASABLES

# THE FOLLOWING CARDS KEEP THE ASSEMBLER HAPPY UNTIL THE SYMBOLS ARE DELETED FROM THE PINBALL NOUN TABLES.

END-E7.0        EQUALS  IRETURN1 +1     # FIRST UNUSED LOCATION IN E7 OVERLAY 0
END-E7.1        EQUALS  N49FLAG +1      # FIRST UNUSED LOCATION IN E7 OVERLAY 1
END-E7.2        EQUALS  POINTVSM +6     # FIRST UNUSED LOCATION IN E7 OVERLAY 2
END-E7.3        EQUALS  END-ALIG        # FIRST UNUSED LOCATION IN E7 OVERLAY 3
END-E7.4        EQUALS  3777            # ** LAST LOCATION USED IN E7 OVERLAY 4 **
END-E7.5        EQUALS  TXO +2          # FIRST UNUSED LOCATION IN E7 OVERLAY 5
END-E7          EQUALS  3777            # ** LAST LOCATION USED IN E7 **
```

```
                SETLOC  4000

                COUNT*  $$/RUPTS                # FIX-FIX LEAD INS
                INHINT                          # GO
                CAF     GOBB
                XCH     BBANK
                TCF     GOPROG

                DXCH    ARUPT                   # T6RUPT
                EXTEND
                DCA     T6ADR
                DTCB

                DXCH    ARUPT                   # T5RUPT - AUTOPILOT
                EXTEND
                DCA     T5ADR
                DTCB

                DXCH    ARUPT                   # T3RUPT
                CAF     T3RPTBB
                XCH     BBANK
                TCF     T3RUPT

                DXCH    ARUPT                   # T4RUPT
                CAF     T4RPTBB
                XCH     BBANK
                TCF     T4RUPT

                DXCH    ARUPT                   # KEYRUPT1
                CAF     KEYRPTBB
                XCH     BBANK
                TCF     KEYRUPT1

                DXCH    ARUPT                   # KEYRUPT2
                CAF     MKRUPTBB
                XCH     BBANK
                TCF     MARKRUPT

                DXCH    ARUPT                   # UPRUPT
                CAF     UPRPTBB
                XCH     BBANK
                TCF     UPRUPT

                DXCH    ARUPT                   # DOWNRUPT
                CAF     DWNRPTBB
                XCH     BBANK
                TCF     DODOWNTM

                DXCH    ARUPT                   # RADAR RUPT
                CAF     RDRPTBB
```

```
                XCH     BBANK
                TCF     RADAREAD

                DXCH    ARUPT           # RUPT10 IS USED ONLY BY LANDING GUIDANCE
                CA      RUPT10BB
                XCH     BBANK
                TCF     PITFALL


                EBANK=  LST1            # RESTART USES E0, E3
GOBB            BBCON   GOPROG

                EBANK=  PERROR
T6ADR           2CADR   DOT6RUPT

                EBANK=  LST1
T3RPTBB         BBCON   T3RUPT

                EBANK=  KEYTEMP1
KEYRPTBB        BBCON   KEYRUPT1

                EBANK=  AOTAZ
MKRUPTBB        BBCON   MARKRUPT

UPRPTBB         =       KEYRPTBB

                EBANK=  DNTMBUFF
DWNRPTBB        BBCON   DODOWNTM

                EBANK=  RADMODES
RDRPTBB         BBCON   RADAREAD

                EBANK=  M11
T4RPTBB         BBCON   T4RUPT

                EBANK=  ELVIRA
RUPT10BB        BBCON   PITFALL
```

```
                BANK    12
                SETLOC  T4RUP
                BANK

                EBANK=  M11
                COUNT*  $$/T4RPT
T4RUPT          TS      BANKRUPT

                EXTEND
                QXCH    QRUPT

                CCS     DSRUPTSW        # GOES 7(-1)0 AROUND AND AROUND
                TCF     NORMT4 +1
                TCF     NORMT4

                TCF     QUIKDSP

NORMT4          CAF     SEVEN
                TS      RUPTREG1
                TS      DSRUPTSW

                BLOCK   02
                SETLOC  FFTAG10
                BANK

                COUNT*  $$/T4RPT
100MRUPT        =       OCT37766        # (DEC 16374)
                # RELTAB IS A PACKED TABLE. RELAYWORD CODE IN UPPER 4 BITS, RELAY CODE
                # IN LOWER 5 BITS.

RELTAB          OCT     04025
                OCT     10003
                OCT     14031
                OCT     20033
                OCT     24017
                OCT     30036
                OCT     34034
                OCT     40023
                OCT     44035
                OCT     50037
                OCT     54000
RELTAB11        OCT     60000
```

```
                # SWITCHED-BANK PORTION

                BANK     12
                SETLOC   T4RUP
                BANK

                COUNT*   $$/T4RPT
CDRVE           CCS      DSPTAB +11D
                TC       DSPOUT
                TC       DSPOUT

                XCH      DSPTAB +11D
                MASK     LOW11
                TS       DSPTAB +11D
                AD       RELTAB11
                EXTEND
                WRITE    OUT0
                TC       HANG20
```

```
                # DSPOUT PROGRAM, PUTS OUT DISPLAYS

DSPOUTSB        TS      NOUT
                CS      ZERO
                TS      DSRUPTEM        # SET TO -0 FOR 1ST PASS THRU DSPTAB
                XCH     DSPCNT
                AD      NEGO            # TO PREVENT +0
                TS      DSPCNT
DSPSCAN         INDEX   DSPCNT
                CCS     DSPTAB
                CCS     DSPCNT          # IF DSPTAB ENTRY +, SKIP
                TCF     DSPSCAN -2      # IF DSPCNT +, TRY AGAIN
                TCF     DSPLAY          # IF DSPTAB ENTRY -, DISPLAY
TABLNTH         OCT     12              # DEC 10, LENGTH OF DSPTAB
                CCS     DSRUPTEM        # IF DSRUPTEM=+0, 2ND PASS THRU DSPTAB
120MRUPT        DEC     16372           # (DSPCNT = 0).  +0 INTO NOUT.
                TS      NOUT
                TC      Q
                TS      DSRUPTEM        # IF DSRUPTEM=-0, 1ST PASS THRU DSPTAB
                CAF     TABLNTH         # (DSPCNT=0).+0 INTO DSRUPTEM. PASS AGAIN
                TCF     DSPSCAN -1

DSPLAY          AD      ONE
                INDEX   DSPCNT
                TS      DSPTAB          # REPLACE POSITIVELY
                MASK    LOW11           # REMOVE BITS 12 TO 15
                TS      DSRUPTEM
                CAF     HI5
                INDEX   DSPCNT
                MASK    RELTAB          # PICK UP BITS 12 TO 15 OF RELTAB ENTRY
                AD      DSRUPTEM
                EXTEND
                WRITE   OUTO

                TCF     Q+1

DSPOUT          CCS     FLAGWRD5        # IS DSKY FLAG ON
                CAF     ZERO            # NO
                TCF     NODSPOUT        # NO
                CCS     NOUT            # YES
                TC      DSPOUTSB
                TCF     NODSPOUT        # NO DISPLAY REQUESTS

HANG20          CS      14,11,9
                ADS     DSRUPTSW

                CAF     20MRUPT

SETTIME4        TS      TIME4
```

```
                # THE STATUS OF THE PROCEED PUSHBUTTON IS MONITORED EVERY 120 MILLISECONDS VIA THE CHANNEL 32 BIT 14 INBIT.
                #  THE STATE OF THIS INBIT IS COMPARED WITH ITS STATE DURING THE PREVIOUS T4RUPT AND IS PROCESSED AS FOLLOWS.
                #       IF PREV ON AND NOW ON   -- BYPASS.
                #       IF PREV ON AND NOW OFF  -- UPDATE IMODES33.
                #       IF PREV OFF AND NOW ON  -- UPDATE IMODES33 AND PROCESS VIA PINBALL.
                #       IF PREV OFF AND NOW OFF -- BYPASS.
                # THE LOGIC EMPLOYED REQUIRES ONLY 9 MCT (APPROX. 108 MICROSECONDS) OF COMPUTER TIME WHEN NO CHANGES OCCUR.

PROCEEDE        CA      IMODES33        # MONITOR FOR PROCEED BUTTON
                EXTEND
                RXOR    CHAN32
                MASK    BIT14
                EXTEND
                BZF     T4JUMP          # NO CHANGE

                LXCH    IMODES33
                EXTEND
                RXOR    LCHAN
                TS      IMODES33        # UPDATE IMODES33
                MASK    BIT14
                CCS     A
                TCF     T4JUMP          # WAS ON -- NOW OFF

                CAF     CHRPRIO         # WAS OFF -- NOW ON
                TC      NOVAC
                EBANK=  DSPCOUNT
                2CADR   PROCKEY
```

```
                # JUMP TO APPROPRIATE ONCE-PER SECOND (0.96 SEC ACTUALLY) ACTIVITY

T4JUMP          INDEX   RUPTREG1
                TCF     +1

                TC      RCSMONIT
                TCF     RRAUTCHK
                TCF     IMUMON
                TCF     DAPT4S
                TC      RCSMONIT
                TCF     RRAUTCHK
                TCF     IMUMON
                TCF     DAPT4S

20MRUPT         =       OCT37776        # (DEC 16382)
```

```
                # ADDITIONAL ROUTINES FOR 20MS. KEYBOARD ACTIVITY

NODSPOUT        EXTEND
                WRITE   OUT0

                CAF     120MRUPT        #SET FOR NEXT CCRIVE
                TCF     SETTIME4

QUIKDSP         CAF     BIT14
                MASK    DSRUPTSW
                EXTEND
                BZF     QUIKOFF         # WROTE LAST TIME, NOW TURN OFF RELAYS.

                CCS     NOUT
                TC      DSPOUTSB
                TCF     NODSPY          # NOUT=0 OR BAD RETURN FROM DSPOUTSB
                CS      BIT14           # GOOD RETURN (WE DISPLAYED SOMETHING)
QUIKRUPT        ADS     DSRUPTSW

                CAF     20MRUPT
                TS      TIME4

                CAF     BIT9
                ADS     DSRUPTSW

                TC      RESUME

NODSPY          EXTEND
                WRITE   OUT0

SYNCT4          CAF     20MRUPT
                ADS     TIME4

                CAF     BIT9
                ADS     DSRUPTSW

                CCS     DSRUPTSW
                TC      RESUME
OCT37737        OCT     37737
                TC      SYNCT4
                TC      RESUME

QUIKOFF         EXTEND
                WRITE   OUT0
                CAF     BIT14           # RESET DSRUPTSW TO SEND DISPLAY NEXT PASS
                TCF     QUIKRUPT

14,11,9         OCT     22400
```

```
# PROGRAM NAME:  IMUMON

# FUNCTIONAL DESCRIPTION:  THIS PROGRAM IS ENTERED EVERY 480 MS.  IT DETECTS CHANGES OF THE IMU STATUS BITS IN
# CHANNEL 30 AND CALLS THE APPROPRIATE SUBROUTINES.  THE BITS PROCESSED AND THEIR RELEVANT SUROUTINES ARE:

#       FUNCTION                BIT     SUBROUTINE CALLED
#       --------                ---     -----------------
#       TEMP IN LIMITS           15     TLIM
#       ISS TURN-ON REQUEST      14     ITURNON
#       IMU FAIL                 13     IMUFAIL (SETISSW)
#       IMU CDU FAIL             12     ICDUFAIL (SETISSW)
#       IMU CAGE                 11     IMUCAGE
#       IMU OPERATE               9     IMUOP

# THE LAST SAMPLED STATE OF THESE BITS IS LEFT IN IMODES30.  ALSO, EACH SUBROUTINE CALLED FINDS THE NEW
# VALUE OF THE BIT IN A, WITH Q SET TO THE PROPER RETURN LOCATION NXTIFAIL.

# CALLING SEQUENCE:  T4RUPT EVERY 480 MILLISECONDS.

# JOBS OR TASKS INITIATED:  NONE.

# SUBROUTINES CALLED:  TLIM, TURNON, SETISSW, IMUCAGE, IMUOP.

# ERASABELE INITIALIZATION:
#       FRESH START OR RESTART WITH NO GROUPS ACTIVE:  C((MODES30) = OCT 37411).
#       RESTART WITH ACTIVE GROUPS:     C(IMODES30) = (B(IMODES30)AND(OCT 00035)) PLUS OCT 37400.
#                                       THIS LEAVES IMU FAIL BITS INTACT.

# ALARMS:  NONE.

# EXIT:  TNONTEST.

# OUTPUT:  UPDATED IMODES30 WITH CHANGES PROCESSED BY APPROPRIATE SUBROUTINE.

IMUMON          CA      IMODES30        # SEE IF THERE HAS BEEN A CHANGE IN THE
                EXTEND                  # RELEVANT BITS OF CHAN 30.
                RXOR    CHAN30
                MASK    30RDMSK
                EXTEND
                BZF     TNONTEST        # NO CHANGE IN STATUS

                TS      RUPTREG1        # SAVE BITS WHICH HAVE CHANGED.
                LXCH    IMODES30        # UPDATE IMODES30.
                EXTEND
                RXOR    LCHAN
                TS      IMODES30

                CS      ONE
                XCH     RUPTREG1
                EXTEND
```

```
                BZMF    TLIM            # CHANGE IN IMU TEMP.
                TCF     NXTIFBIT        # BEGIN BIT SCAN.

 -1             AD      ONE             # (RE-ENTERS HERE FROM NXTIFAIL.)
NXTIFBIT        INCR    RUPTREG1        # ADVANCE BIT POSITION NUMBER.
 +1             DOUBLE
                TS      A               # SKIP IF OVERFLOW.
                TCF     NXTIFBIT        # LOOK FOR BIT.

                XCH     RUPTREG2        # SAVE OVERFLOW-CORRECTED DATA.
                INDEX   RUPTREG1        # SELECT NEW VALUE OF THIS BIT.
                CAF     BIT14
                MASK    IMODES30
                INDEX   RUPTREG1
                TC      IFAILJMP

NXTIFAIL        CCS     RUPTREG2        # PROCESS ANY ADDITIONAL CHANGES.
                TCF     NXTIFBIT -1
```

```
# PROGRAM NAME:  TNONTEST.

# FUNCTIONAL DESCRIPTION:  THIS PROGRAM HONORS REQUESTS FOR ISS INITIALIZATION.  ISS TURN-ON (CHANNEL 30 BIT 14)
# AND ISS OPERATE (CHANNEL 30 BIT 9) REQUESTS ARE TREATED AS A PAIR AND PROCESSING TAKES PLACE .480 SECONDS
# AFTER EITHER ONE APPEARS.  THIS INITIALIZATION TAKES ON ONE OF THE FOLLOWING THREE FORMS:

#       1) ISS TURN-ON:  IN THIS SITUATION THE COMPUTER IS OPERATING WHEN THE ISS IS TURNED ON.  NOMINALLY,
#       BOTH ISS TURN-ON AND ISS OPERATE APPEAR.  THE PLATFORM IS CAGED FOR 90 SECONDS AND THE ICDU'S ZEROED
#       SO THAT AT THE END OF THE PROCESS THE GIMBAL LOCK MONITOR WILL FUNCTION PROPERLY.

#       2) ICDU INITIALIZATION:  IN THIS CASE THE COMPUTER WAS PROBABLY TURNED ON WITH THE ISS IN OPERATE OR
#       A FRESH START WAS DONE WIT THE ISS IN OPERATE.  IN THIS CASE ONLY ISS OPERATE IS ON.  THE ICDU'S ARE
#       ZEROED SO THE GIMBAL LOCK MONITOR WILL FUNCTION.  AN EXCEPTION IS IF THE ISS IS IN GIMBAL LOCK AFTER
#       A RESTART, THE ICDU'S WILL NOT BE ZEROED.

#       3) RESTART WITH RESTARTABLE PROGRAM USING THE IMU:  IN THIS CASE, NO INITIALIZATION TAKES PLACE SINCE
#       IT IS ASSUMED THT THE USING PROGRAM DID THE INITIALIZATION AND THEREFORE T4RUPT SHOULD NOT INTERFERE.

# IMODES30 BIT 7 IS SET = 1 BY THE FIRST BIT (CHANNEL 30 BIT 14 OR 9) WHICH ARRIVES.  FOLLOWING THIS, TNONTEST IS
# ENTERED, FINDS BIT 7 = 1 BUT BIT 8 = 0, SO IT SETS BIT 8 = 1 AND EXITS.  THE NEXT TIME IT FINDS BIT 8 = 1 AND
# PROCEEDS, SETTING BITS 8 AND 7 = 0.  AT PROCTNON, IF ISS TURN-ON REQUEST IS PRESENT, THE ISS IS CAGED (ZERO +
# COARSE).  IF ISS OPERATE IS NOT PRESENT PROGRAM ALARM 00213 IS ISSUED.  AT THE END OF A 90 SECOND CAGE, BIT 2
# OF IMODES30 IS TESTED.  IF IT IS = 1, ISS TURN-ON WAS NOT PRESENT FOR THE ENTIRE 90 SECONDS.  IN THAT CASE, IF
# THE ISS TURN-ON REQUEST IS PRESENT TEH 90 SECOD WAIT IS REPEATED.  OTHERWISE NO ACTION OCURS UNLESS A PROGRAM
# WAS WAITING FOR THE INITIALIZATION IN WHIC CASE TH PROGRAM IS GIVEN AN IMUSTALL ERROR RETURN.  IF THE DELAY
# WENT PROPERLY, THE ISS DELAY OUTBIT IS SENT AND THE ICDU'S ZEROED.  A TASK IS INITIATED TO REMOVE THE PIPA FAIL
# INHIBIT BIT IN 10.24 SECONDS.  IF A MISSION PROGRAM WAS WAITING IT IS INFORMED VIA ENDIMU.

# AT PROCTNON, IF ONLY ISS OPERATE IS PRESENT (OPONLY), THE CDU'S ARE ZEROED UNLESS THE PLATFORM IS IN COARSE
# ALIGN (= GIMBAL LOCK HERE) OR A MISSIN PROGRAM IS USING THE IMU (INUSEFLG = 1).

# CALLING SEQUENCE:  T4RUPT EVERY 480 MILLISECONDS AFTER IMUMON.

# JOBS OR TASKS INITIATED:  1) ENDTNON, 90 SECONDS AFER CAGING STARTED.  2) ISSUP, 4 SECONDS AFTER CAGING DONE.
#       3) PFAILOK, 10.24 SECONDS AFTER INITIALIZATION COMPLETED.  4) UNZ2, 320 MILLISECONDS AFTER ZEROING
#       STARTED.

# SUBROUTINES CALLED: CAGESUB, CAGESUB2, ZEROICDU, ENDIMU, IMUBAD, NOATTOFF, SETISSW, VARDELAY.

# ERASABLE INITIALIZATION:  SEE IMUMON.

# ALARMS:  PROGRAM ALARM 00213 IF ISS TURN-ON REQUESTED WITHOUT ISS OPERATE.

# EXIT:  ENDTNON EXITS TO C33TEST.  TASKS HAVING TO DO WITH INITIALIZATION EXIT AS FOLLOWS:  MISSION PROGRAM
# WAITING AND INITIALIZATION COMPLETE, EXIT TO ENDIMU, MISSION PROGRAM WAITING AND INITIALIZATION FAILED, EXIT TO
# IMUBAD, IMU NOT IN USE, EXIT TO TASKOVER.

# OUTPUT:  ISS INITIALIZED.

TNONTEST        CS      IMODES30        # AFTER PROCESSING ALL CHANGES, SEE IF IT
```

```
                MASK    BIT7            # IS TIME TO ACT ON A TURN-ON SEQUENCE.
                CCS     A
                TCF     C33TEST         # NO -- EXAMINE CHANNEL 33.

                CAF     BIT8            # SEE IF FIRST SAMPLE OR SECOND.
                MASK    IMODES30
                CCS     A
                TCF     PROCTNON        # REACT AFTER A SECOND SAMPLE.

                CAF     BIT8            # IF FIRST SAMPLE, SET BIT TO REACT NEXT
                ADS     IMODES30        # TIME.
                TCF     C33TEST

                # PROCESS IMU TURN-ON REQUESTS AFTER WAITING 1 SAMPLE FOR ALL SIGNALS TO ARRIVE.

PROCTNON        CS      BITS7&8
                MASK    IMODES30
                TS      IMODES30
                MASK    BIT14           # SEE IF TURN-ON REQUEST.
                CCS     A
                TCF     OPONLY          # OPERATE ON ONLY.

                CS      IMODES30        # IF TURN-ON REQUEST, WE SHOUD HAVE IMU
                MASK    BIT9            # OPERATE.
                CCS     A
                TCF     +3

                TC      ALARM           # ALARM IF NOT
                OCT     213

 +3             TC      CAGESUB
                CAF     90SECS
                TC      WAITLIST
                EBANK=  M11
                2CADR   ENDTNON

                TCF     C33TEST

RETNON          CAF     90SECS
                TC      VARDELAY

ENDTNON         CS      BIT2            # RESET TURN-ON REQUEST FAIL BIT.
                MASK    IMODES30
                XCH     IMODES30
                MASK    BIT2            # IF IT WAS OFF, SEND ISS DELAY COMPLETE.
                EXTEND
                BZF     ENDTNON2

                CAF     BIT14           # IF IT WAS ON AND TURN-ON REQUEST NOW.
```

```
                MASK    IMODES30        # PRESENT, RE-ENTER 90 SEC DELAY IN WL.
                EXTEND
                BZF     RETNON

                CS      FLAGWRD0        # IF IT IS NOT ON NOW, SEE IF A PROG WAS
                MASK    IMUSEBIT        # WAITING.
                CCS     A
                TCF     TASKOVER
                TC      POSTJUMP
                CADR    IMUBAD          # UNSUCCESSFUL TURN-ON.

ENDTNON2        CAF     BIT15           # SEND ISS DELAY COMPLETE.
                EXTEND
                WOR     CHAN12

                TC      IBNKCALL        # TURN OFF NO ATT LAMP.
                CADR    NOATTOFF

UNZ2            TC      ZEROICDU

                CS      BITS4&5         # REMOVE ZERO AND COARSE.
                EXTEND
                WAND    CHAN12

                CAF     BIT11           # WAIT 10 SECS FOR CTRS TO FIND GIMBALS
                TC      VARDELAY

ISSUP           CS      OCT54           # REMOVE CAGING, IMU FAIL INHIBIT BIT, AND
                MASK    IMODES30        # ICDUFAIL INHIBIT FLAGS.
                TS      IMODES30

                CS      BIT6            # ENABLE DAP
                MASK    IMODES33
                TS      IMODES33

                CS      FLAGWRD2        # TEST DRIFTFLG: IF ON DO NOTHING BECAUSE
                MASK    DRFTBIT         # IMUCOMP SHOUD BE ALL SET UP (RESTART
                EXTEND                  # WITH IMUSE DOWN).  IF OFF, SET DRIFTFLG
                BZF     +4              # AND 1/PIPADT TO GET FREEFALL IMUCOMP
                ADS     FLAGWRD2        # GOING (FRESH START OR ISS TURN-ON).
                CA      TIME1
                XCH     1/PIPADT        # CANNOT GET HERE IF RESTART WITH IMUSE UP

                TC      SETISSW         # ISS WARNING MIGHT HAVE BEEN INHIBITED.

                CS      BIT15           # REMOVE IMU DELAY COMPLETE DISCRETE.
                EXTEND
                WAND    CHAN12

                CAF     4SECS           # DON'T ENABLE PROG ALARM ON PIP FAIL FOR
```

```
                TC      WAITLIST        # ANOTHER 4 SECS.
                EBANK=  CDUIND
                2CADR   PFAILOK

                TCF     TASKOVER

OPONLY          CAF     BIT4            # IF OPERATE ON ONLY, AND WE ARE IN COARSE
                EXTEND                  # ALIGN, DON'T ZERO THE CDUS BECAUSE WE
                RAND    CHAN12          # MIGHT BE IN GIMBAL LOCK.
                CCS     A
                TCF     C33TEST

                CAF     IMUSEBIT        # OTHERWISE, ZERO THE COUNTERS.
                MASK    FLAGWRD0        # UNLESS SOMEONE IS USING TH IMU.
                CCS     A
                TCF     C33TEST

                TC      CAGESUB2        # SET TURNON FLAGS.

ISSZERO         TC      IBNKCALL        # TURN OFF NO ATT LAMP.
                CADR    NOATTOFF        # IMU CAGE OFF ENTRY.

                CAF     BIT5            # ISS CDU ZERO
                EXTEND
                WOR     CHAN12

                TC      ZEROICDU
                CAF     BIT6            # WAIT 300 MS. FOR AGS TO RECEIVE SIGNAL.
                TC      WAITLIST
                EBANK=  M11
                2CADR   UNZ2

                TCF     C33TEST
```

```
# PROGRAM NAME:  C33TEST
#
# FUNCTIONAL DESCRIPTION:  THIS PROGRAM MONITORS THREE FLIP-FLOP INBITS OF CHANNEL 33 AND CALLS THE APPROPRIATE
# SUBROUTINE TO PROCESS A CHANGE.  IT IS ANALOGOUS TO IMUMON, WHICH MONITORS CHANNEL 30, EXCEPT THAT IT READS
# CHANNEL 33 WITH A WAND INSTRUCTION BECAUSE A `WRITE' PULSE IS REQUIRED TO RESET THE FLIP-FLOPS.  THE BITS
# PROCESSED AND THE SUBROUTINES CALLED ARE:
#       BIT     FUNCTION                SUBROUTINE
#       ---     --------                ----------
#        13     PIPA FAIL               PIPFAIL
#        12     DOWNLINK TOO FAST       DNTMFAST
#        11     UPLINK TOO FAST         UPTMFAST
#
# UPON ENTRY TO THE SUBROUTINE, THE NEW BIT STATE IS IN A.
#
# CALLING SEQUENCE:  EVERY 480 MILLISECONDS AFTER TNONTEST.
#
# JOBS OR TASKS INITIATED:  NONE.
#
# SUBROUTINES CALLED:  PIPFAIL, DNTMFAST AND UPTMFAST ON BIT CHANGES.
#
# ERASABLE INITIALIZATION:  C(IMODES33) = OCT 16000 ON A FRESH START OR RESTART, THEREFORE, THESE ALARMS WILL
# REAPPEAR IF THE CONDITIONS PERSIST.
#
# ALARMS:  NONE.
#
# EXIT:  GLOCKMON.
#
# OUTPUT:  UPDATED BITS 13, 12, AND 11 OF IMODES33 WITH CHANGES PROCESSED.

C33TEST         CA      IMODES33                # SEE IF RELEVANT CHAN33 BITS HAVE
                MASK    33RDMSK
                TS      L                       # CHANGED.
                CAF     33RDMSK
                EXTEND
                WAND    CHAN33                  # RESETS FLIP-FLOP INPUTS
                EXTEND
                RXOR    LCHAN
                EXTEND
                BZF     GLOCKMON                # ON NO CHANGE.

                TS      RUPTREG1                # SAVE BITS WHICH HAVE CHANGED.
                LXCH    IMODES33
                EXTEND
                RXOR    LCHAN
                TS      IMODES33                # UPDATED IMODES33.

                CAF     ZERO
                XCH     RUPTREG1
                DOUBLE
```

```
                TCF     NXTIBT +1               # SCAN FOR BIT CHANGES.

 -1             AD      ONE
NXTIBT          INCR    RUPTREG1
 +1             DOUBLE
                TS      A                       # (CODING IDENTICAL TO CHAN 30).
                TCF     NXTIBT

                XCH     RUPTREG2
                INDEX   RUPTREG1                # GET NEW VALUE OF BIT WHICH CHANGED.
                CAF     BIT13
                MASK    IMODES33
                INDEX   RUPTREG1
                TC      C33JMP
NXTFL33         CCS     RUPTREG2                # PROCESS POSSIBLE ADDITIONAL CHANGES.
                TCF     NXTIBT -1
```

```
# PROGRAM NAME:  GLOCKMON
#
# FUNCTIONAL DESCRIPTION:  THIS PROGRAM MONITORS THE CDUZ COUNTER TO DETERMINE WHETHER THE ISS IS IN GIMBAL LOCK
# AND TAKES ACTION IF IT IS.  THREE REGIONS OF MIDDLE GIMBAL ANGLE (MGA) ARE USED:
#
#       1) ABS(MGA) LESS THAN OR EQUAL TO 70 DEGREES -- NORMAL MODE.
#       2) ABS(MGA) GREATER THAN 70 DEGREES AND LESS THAN OR EQUAL TO 85 DEGREES -- GIMBAL LOCK LAMP TURNED ON.
#       3) ABS(MGA) GREATER THAN 85 DEGREES -- ISS PUT IN COARSE ALIGN AND NO ATT LAMP TURNED ON.
#
# CALLING SEQUENCE:  EVERY 480 MILLISECONDS AFTER C33TEST.
#
# JOBS OR TASKS INITIATED:  NONE.
#
# SUBROUTINES CALLED:   1) SETCOARS WHEN ABS(MGA) GREATER THEN 85 DEGREES AND ISS NOT IN COARSE ALIGN.
#                       2) LAMPTEST BEFORE TURNING OFF GIMBAL LOCK LAMP.
#
# ERASABLE INITIALIZATION:
#               1) FRESH START OR RESTART WITH NO GROUPS ACTIVE:  C(CDUZ) = 0, IMODES30 BIT 6 = 0, IMODES33 BIT 1 =  0.
#               2) RESTART WTIH GROUPS ACTIVE:  SAME AS FRESH START EXCEPT C(CDUZ) NOT CHANGED SO GIMBAL MONITOR
#                                               PROCEEDS AS BEFORE.
#
# ALARMS:       1) MGA REGION (2) CAUSES GIMBAL LOCK LAMP TO BE LIT.
#               2) MGA REGION (3) CAUSES THE ISS TO BE PUT IN COARSE ALIGN AND THE NO ATT LAMP TO BE LIT IF EITHER NOT
#                  SO ALREADY.

GLOCKMON        CCS     CDUZ
                TCF     GLOCKCHK                # SEE IF MAGNITUDE OF MGA IS GREATER THAN
                TCF     SETGLOCK                # 70 DEGREES.
                TCF     GLOCKCHK
                TCF     SETGLOCK

GLOCKCHK        AD      -70DEGS
                EXTEND
                BZMF    SETGLOCK -1             # NO LOCK.

                AD      -15DEGS                 # SEE IF ABS(MGA) GREATER THAN 85 DEGREES
                EXTEND
                BZMF    NOGIMRUN

                CAF     BIT4                    # IF SO, SYSTEM SHOULD BE IN COARSE ALIGN
                EXTEND                          # TO PREVENT GIMBAL RUNAWAY.
                RAND    CHAN12
                CCS     A
                TCF     NOGIMRUN

                TC      IBNKCALL
                CADR    SETCOARS

                CAF     SIX                     # ENABLE ISS ERROR COUNTERS IN 60 MS.
                TC      WAITLIST
```

```
                EBANK=  CDUIND
                2CADR   CA+ECE

NOGIMRUN        CAF     BIT6                    # TURN ON GIMBAL LOCK LAMP.
                TCF     SETGLOCK

 -1             CAF     ZERO
SETGLOCK        AD      DSPTAB +11D             # SEE IF PRESENT STATE OF GIMBAL LOCK LAMP
                MASK    BIT6                    # AGREES WITH DESIRED STATE BY HALF ADDING
                EXTEND                          # THE TWO.
                BZF     GLOCKOK                 # OK AS IS.

                MASK    DSPTAB +11D             # IF OFF, DON'T TURN ON IF IMU BEING CAGED.
                CCS     A
                TCF     GLAMPTST                # TURN OFF UNLESS LAMP TEST IN PROGRESS.

                CAF     BIT6
                MASK    IMODES30
                CCS     A
                TCF     GLOCKOK

GLINVERT        CS      DSPTAB +11D             # INVERT GIMBAL LOCK LAMP.
                MASK    BIT6
                AD      BIT15                   # TO INDICATE CHANGE IN DSPTAB +11D.
                XCH     DSPTAB +11D
                MASK    OCT37737
                ADS     DSPTAB +11D
                TCF     GLOCKOK

GLAMPTST        TC      LAMPTEST                # TURN OFF UNLESS LAMP TEST IN PROGRESS.
                TCF     GLOCKOK
                TCF     GLINVERT

-70DEGS         DEC     -.38888                 # -70 DEGREES SCALED IN HALF-REVOLUTIONS.
-15DEGS         DEC     -.08333
```

```
# PROGRAM NAME:  TLIM.
#
# FUNCTIONAL DESCRIPTION:  THIS PROGRAM MAINTAINS THE TEMP LAMP (BIT 4 OF CHANNEL 11) ON THE DSKY TO AGREE WITH
# THE TEMP SIGNAL FROM THE ISS (BIT 15 OF CHANNEL 30).  HOWEVER, THE LIGHT WILL NOT BE TURNED OFF IF A LAMP TEST
# IS IN PROGRESS.
#
# CALLING SEQUENCE:  CALLED BY IMUMON ON A CHANGE OF BIT 15 OF CHANNEL 30.
#
# JOBS OR TASKS INITIATED:  NON.
#
# SUBROUTINES CALLED:  LAMPTEST.
#
# ERASABLE INITIALIZATION:  FRESH START AND RESTART TURN THE TEMP LAMP OFF.
#
# ALARMS:  TEMP LAMP TURNED ON WHEN THE IMU TEMP GOES OUT OF LIMITS.
#
# EXIT:  NXTIFAIL.
#
# OUTPUT:  SERVICE OF TEMP LAMP.                      IN A, EXCEPT FOR TLIM.

TLIM            MASK    POSMAX                  # REMOVE BIT FROM WORD OF CHANGES AND SET
                TS      RUPTREG2                # DSKY TEMP LAMP ACCORDINGLY.

                CCS     IMODES30
                TCF     TEMPOK
                TCF     TEMPOK

                CAF     BIT4                    # TURN ON LAMP.
                EXTEND
                WOR     DSALMOUT
                TCF     NXTIFAIL

TEMPOK          TC      LAMPTEST                # IF TEMP NOW OK, DON'T TURN OFF LAMP IF
                TCF     NXTIFAIL                # LAMP TEST IN PROGRESS.

                CS      BIT4
                EXTEND
                WAND    DSALMOUT                # TURN OFF LAMP
                TCF     NXTIFAIL
```

```
# PROGRAM NAME:  ITURNON.
#
# FUNCTIONAL DESCRIPTION:  THIS PROGRAM IS CALLED BY IMUMON WHEN A CHANGE OF BIT 14 OF CHANNEL 30 (ISS TURN-ON
# REQUEST) IS DETECTED.  UPON ENTRY, ITURNON CHECKS IF A TURN-ON DELAY SEQUENCE HAS FAILED, AND IF SO, IT EXITS.
# IF NOT, IT CHECKS WHETHER THE TURN-ON REQUEST CHANGE IS TO ON OR OFF.  IF ON, IT SETS BIT7 OF IMODES30 TO 1 SO
# THAT TNONTEST WILL INITIATE THE ISS INITIALIZATION SEQUENCE.  IF OFF, THE TURN-ON DELAY SIGNAL, CHANNEL 12 BIT
# 15, IS CHECKED AND IF IT IS ON, ITURNON EXITS.  IF THE DEALY SIGNAL IS OFF, PROGRAM ALARM 00207 IS ISSUED, BIT 2
# OF IMODES30 IS SET TO 1 AND THE PROGRAM EXITS.
#
# THE SETTING OF BIT 2 OF IMODES30 (ISS DELAY SEQUENCE FAIL) INHIBITS THIS ROUTINE AND IMUOP FROM
# PROCESSING ANY CHANGES.  THIS BIT WILL BE RESET BY THE ENDTNON ROUTINE WHEN THE CURRENT 90 SECOND DELAY PERIOD
# ENDS.
#
# CALLING SEQUENCE:  FROM IMUMON WHEN ISS TURN-ON REQUEST CHANGES STATE.
#
# JOBS OR TASKS INITITIATED:  NONE.
#
# SUBROUTINES CALLED:  ALARM, IF THE ISS TURN-ON REQUEST IS NOT PRESENT FOR 90 SECONDS.
#
# ERASABLE INITIALIZATION:  FRESH START AND RESTART SET BIT 15 OF CHANNEL 12 AND BITS 2 AND 7 OF IMODES30 TO 0,
# AND BIT 14 OF IMODES30 TO 1.
#
# ALARMS: PROGRAM ALARM 00207 IS ISSUED IF THE ISS TURN-ON REQUEST SIGNAL IS NOT PRESENT FOR 90 SECONDS.
#
# EXIT:  NXTIFAIL.
#
# OUTPUT:  BIT 7 OF IMODES30 TO START ISS INITIALIZATION, OR BIT 2 OF IMODES30 AND PROGRAM ALARM 00207 TO INDICATE
# A FAILED TURN-ON SEQUENCE.

ITURNON         CAF     BIT2            # IF DELAY REQUEST HAS GONE OFF
                MASK    IMODES30        # PREMATURELY, DO NOT PROCESS ANY CHANGES
                CCS     A               # UNTIL THE CURRENT 90 SEC WAIT EXPIRES.
                TCF     NXTIFAIL

                CAF     BIT14           # SEE IF JUST ON OR OFF.
                MASK    IMODES30
                EXTEND
                BZF     ITURNON2        # IF JUST ON.

                CAF     BIT15
                EXTEND                  # SEE IF DELAY PRESENT DISCRETE HAS BEEN
                RAND    CHAN12          # SENT.  IF SO, ACTION COMPLETE
                EXTEND
                BZF     +2
                TCF     NXTIFAIL

                CAF     BIT2            # IF NOT, SET BIT TO INDICATE REQUEST NOT
                ADS     IMODES30        # PRESENT FOR FULL DURATION.
                TC      ALARM
                OCT     207
                TCF     NXTIFAIL
```

```
ITURNON2        CS      IMODES30        # SET BIT7 TO INDICATE WAIT OF 1 SAMPLE
                MASK    BIT7
                ADS     IMODES30
                CAF     RRINIT
                TS      RADMODES
                TCF     NXTIFAIL

RRINIT          OCT     00102
```

```
# PROGRAM NAME:  IMUCAGE.
#
# FUNCTIONAL DESCRIPTION:  THIS PROGRAM PROCESSES CHANGES OF THE IMUCAGE INBIT, CHANNEL 30 BITS 11.  IF THE BIT
# CHANGES TO 0 (CAGE BUTTON PRESSED), THE ISS IS CAGED (ICDU ZERO + COARSE ALIGN + NO ATT LAMP) UNTIL THE
# ASTRONAUT SELECTS ANOTHER PROGRAM TO ALIGN THE ISS.  ANY PULSE TRAINS TO THE ICDU'S AND GYRO'S ARE TERMINATED,
# THE ASSOCIATE OUTCOUNTERS ARE ZEROED AND THE GYRO'S ARE DE-SELECTED.  NO ACTION OCCURS WHEN THE BUTTON IS
# RELEASED (INBIT CHANGES TO 1).
#
# CALLING SEQUENCE:  BY IMUMON WHEN IMU CAGE BIT CHANGES.
#
# JOBS OR TASKS INITIATED:  NONE.
#
# SUBROUTINES CALLED:  CAGESUB.
#
# ERASABLE INITIALZATION:  FRESH START AND RESTART SET BIT 11 OF IMODES30 TO 1.
#
# ALARMS: NONE.
#
# EXIT:  NXTIFAIL.
#
# OUTPUT:  ISS CAGED, COUNTERS ZEROED, PULSE TRAINS TERMINATED AND NO ATT LAMP LIT.

IMUCAGE         CCS     A               # NO ACTION OF GOING OFF.
                TCF     ISSZERO
                CS      OCT77000        # TERMINATE ICDU, RCDU, GYRO PULSE TRAINS
                EXTEND
                WAND    CHAN14

                CS      OCT272          # KNOCK DOWN DISPLAY INERTIAL DATA, IMU
                EXTEND                  # ERROR COUNTER ENABLE, ZERO ICDU, COARSE
                WAND    CHAN12          # ALIGN ENABLE, RR ERROR COUNTER ENABLE.

                CS      ENGONBIT        # INSURE ENGONFLG IS CLEAR.
                MASK    FLAGWRD5
                TS      FLAGWRD5
                CS      PRIO30          # TURN ENGINE OFF.
                EXTEND
                RAND    DSALMOUT
                AD      BIT14
                EXTEND
                WRITE   DSALMOUT        # FORCE BIT14=1, BIT13=0.

                TC      CAGESUB1

                TC      IBNKCALL        # KNOCK DOWN TRACK, REFSMMAT, DRIFT FLAGS
                CADR    RNDREFDR

                CS      ZERO
                TS      CDUXCMD
                TS      CDUYCMD
```

```
                TS      CDUZCMD
                TS      GYROCMD

                CS      OCT740          # HAVING WAITED AT LEAST 27 MCT FROM
                EXTEND                  # GYRO PULSE TRAIN TERMINATION, WE CAN
                WAND    CHAN14          # DE-SELECT THE GYROS.
                TCF     NXTIFAIL
```

```
# PROGRAM NAME:  IMUOP.
#
# FUNCTIONAL DESCRIPTION:  THIS PROGRAM PROCESSES CHANGES IN THE ISS OPERATE DISCRETE, BIT 9 OF CHANNEL 30.
# IF THE INBIT CHANGES TO 0, INDICATING ISS ON, IMUOP GENERALLY SETS BIT 7 OF IMODES30 TO 1 TO REQUEST ISS
# INITIALIZATION VIA TNONTEST.  AN EXCEPTION IS DURING A FAILED ISS DELAY DURING WHICH BIT 2 OF IMODES30 IS SET
# TO 1 AND NO FURTHER INITIALIZATION IS REQUIRED.  WHEN THE INBIT CHANGES TO 1, INDICATING ISS OFF, IMUSEFLG IS
# TESTED TO SEE IF ANY PROGRAM WAS USING THE ISS.  IF SO, PROGRAM ALARM 00214 IS ISSUED.
#
# CALLING SEQUENCE:  BY IMUMON WHEN BIT 9 OF CHANNEL 30 CHANGES.
#
# JOBS OR TAKS INITIATED:  NONE.
#
# SUBROUTINES CALLED:  ALARM, IF ISS IS TURNED OFF WHILE IN USE.
#
# ERASABLE INITIALIZATION:  ON FRESH START AND RESTART, BIT 9 OF IMODES30 IS SET TO 1 EXCEPT WHEN THE GIMBAL LOCK
# LAMP IS ON, IN WHICH CASE IT IS SET TO 0.  THIS PREVENTS ICDU ZERO BY TNONTEST WITH THE ISS IN GIMBAL LOCK.
#
# ALARMS:  PROGRAM ALARM 00214 IF THE ISS IS TURNED OFF WHILE IN USE.
#
# EXIT:  NXTIFAIL.
#
# OUTPUT:  ISS INITIALIZATION REQUEST (IMODES30 BIT 7) OR PROGRAM ALARM 00214.

IMUOP           EXTEND
                BZF     IMUOP2

                CS      IMODES33                # DISABLE DAP
                MASK    BIT6
                ADS     IMODES33

                TC      IBNKCALL                # KNOCK DOWN TRACK, REFSMMAT, DRIFT FLAGS
                CADR    RNDREFDR

                CS      BITS7&8                 # KNOCK DOWN RENDEZVOUS, IMUUSE FLAGS
                MASK    FLAGWRD0
                XCH     FLAGWRD0                # IF GOING OFF, ALARM IF PROG USING IMU.
                COM
                MASK    IMUSEFLG
                CCS     A
                TCF     NXTIFAIL

                TC      ALARM
                OCT     214
                TCF     NXTIFAIL

IMUOP2          CAF     BIT2                    # SEE IF FAILED ISS TURN-ON SEQ IN PROG.
                MASK    IMODES30
                CCS     A
                TCF     NXTIFAIL                # IF SO, DON'T PROCESS UNTIL PRESENT 90
                TCF     ITURNON2                # SECONDS EXPIRES.
```

```
# PROGRAM NAME:  PIPFAIL
#
# FUNCITONAL DESCRIPTION:  THIS PROGRAM PROCESSES CHANGES OF BIT 13 OF CHANNEL 33, PIPA FAIL.  IT SETS BIT 10 OF
# IMODES30 TO AGREE.  IT CALLS SETISSW IN CASE A PIPA FAIL NECESSITATES AN ISS WARNING.  IF NOT, I.E., IMODES30
# BIT 1 = 1, AND A PIPA FAIL IS PRESENT AND THE ISS NOT BEING INITIALIZED, PROGRAM ALARM 0212 IS ISSUED.
#
# CALLING SEQUENCE:  BY C33TEST ON CHANGES OF CHANNEL 33 BIT 13.
#
# JOBS OR TASKS INITIATED:  NONE.
#
# SUBROUTINES CALLED:  1) SETISSW, AND 2) ALARM (SEE FUNCITONAL DESCRIPTION).
#
# ERASABLE INITIALZIZATION:  SEE IMUMON FOR INITIALIZATION OF IMODES30.  THE RELEVANT BITS ARE 5, 7, 8, 9, AND 10.
#
# ALARMS:  PROGRAM ALARM 00212 IF PIPA FAIL IS PRESENT BUT NEITHER ISS WARNING IS TO BE ISSUED NOR THE ISS IS
# BEING INITIALIZED.
#
# EXIT:  NXTFL33.
#
# OUTPUT:  PROGRAM ALARM 00212 AND ISS WARNING MAINTENANCE.

PIPFAIL         CCS     A                       # SET BIT10 IN IMODES30 SO ALL ISS WARNING
                CAF     BIT10                   # INFO IS IN ONE REGISTER.
                XCH     IMODES30
                MASK    -BIT10
                ADS     IMODES30

                TC      SETISSW

                CS      IMODES30                # IF PIP FAIL DOESN'T LIGHT ISS WARNING, DO
                MASK    BIT1                    # A PROGRAM ALARM IF IMU OPERATING BUT NOT
                CCS     A                       # CAGED OR BEING TURNED ON.
                TCF     NXTFL33

                CA      IMODES30
                MASK    OCT1720
                CCS     A
                TCF     NXTFL33                 # ABOVE CONDITION NOT MET.

                TC      ALARM
                OCT     212
                TCF     NXTFL33
```

```
# PROGRAM NAMES:  DNTMFAST, UPTMFAST
#
# FUNCTIONAL DESCRIPTION:  THESE PROGRAMS PROCESS CHANGES OF BITS 12 AND 11 OF CHANNEL 33.  IF A BIT CHANGES TO A
# 0, A PROGRAM ALARM IS ISSUED.  THE LAARMS ARE:
#
#       BIT     ALARM   CAUSE
#       ---     -----   -----
#        12     01105   DOWNLINK TOO FAST
#        11     01106   UPLINK TOO FAST
#
# CALLING SEQUENCE:  BY C33TEST ON A BIT CHANGE.
#
# SUBROUTINES CALLED:  ALARM, IF A BIT CHANGES TO A 0.
#
# ERASABLE INITIALIZATION:  FRESH START OR RESTART, BITS 12 AND 11 OF IMODES33 ARE SET TO 1.
#
# ALARMS:  SET FUNCTGIONAL DESCRIPTION.
#
# EXIT:  NXTFL33.
#
# OUTPUT:  PROGRAM ALARM ON A BIT CHANGE TO 0.

DNTMFAST        CCS     A                       # DO PROG ALARM IF TM TOO FAST.
                TCF     NXTFL33

                TC      ALARM
                OCT     1105
                TCF     NXTFL33

UPTMFAST        CCS     A                       # SAME AS DNLINK TOO FAST WITH DIFFERENT
                TCF     NXTFL33                 # ALARM CODE.

                TC      ALARM
                OCT     1106
                TCF     NXTFL33
```

```
# PROGRAM NAME:  SETISSW
#
# FUNCTIONAL DESCRIPTION:  THIS PROGRAM TURNS THE ISS WARNING LAMP ON AND OFF (CHANNEL 11 BIT 1 = 1 FOR ON,
# 0 FOR OFF) DEPENDING ON THE STATUS OF IMODES30 BITS 13 (IMU FAIL) AND 4 (INHIBIT IMU FAIL), 12 (ICDU FAIL) AND
# 3 (INHIBIT ICDU FAIL), AND 10 (PIPA FAIL) AND 1 (INHIBIT PIPA FAIL).  THE LAMP IS LEFT ON IF A LAMP TEST IS IN
# PROGRESS.
#
# CALLING SEQUENCE:  CALLED BY IMUMON ON CHANGES TO IMU FAIL AND ICDU FAIL.  CALLED BY IFAILOK AND PFAILOK UPON
# REMOVAL OF THE FAIL INHIBITS.  CALLED BY PIPFAIL WHEN THE PIPA FAIL DISCRETE CHANGES.  IT IS CALLED BY PIPUSE
# SINCE THE PIPA FAIL PROGRAM ALARM MAY NECESSITATE AN ISS WARNING, AND LIKEWISE BY PIPFREE WHEN THE ALARM DEPARTS
# AND IT IS CALLED BY IMUZERO3 AND ISSUP AFTER THE FAIL INHIBITS HAVE BEEN REMOVED.
#
# JOBS OR TASKS INITIAZTED:  NONE.
#
# SUBROUTINES CALLED:  NONE.
#
# ERASABLE INITIALIZATION:
#
#       1) IMODES30 -- SEE IMUMON.
#       2) IMODES33 BIT 1 = 0 (LAMP TEST NOT IN PROGRESS).
#
# ALARMS:  ISS WARNING.
#
# THE FOLLOWING PROGRAM ALARMS WILL SHOW WHICH FAILURE CAUSED THE ISS WARN
#       PROGRAM ALARM 00777     PIPA FAIL
#       PROGRAM ALARM 03777     ICDU FAIL
#       PROGRAM ALARM 04777     ICDU, PIPA FAILS
#       PROGRAM ALARM 07777     IMU FAIL
#       PROGRAM ALARM 10777     IMU, PIPA FAILS
#       PROGRAM ALARM 13777     IMU, ICDU FAILS
#       PROGRAM ALARM 14777     IMU, ICDU, PIPA FAILS
#
# EXIT: VIA Q.
#
# OUTPUT: ISS WARNING LAMP SET PROPERLY.

SETISSW         CAF     OCT15                   # SET ISS WARNING USING THE FAIL BITS IN
                MASK    IMODES30                # BITS 13, 12, AND 10 OF IMODES30 AND THE
                EXTEND                          # FAILURE INHIBIT BITS IN POSITIONS
                MP      BIT10                   # 4, 3, AND 1.
                CA      IMODES30
                EXTEND
                ROR     LCHAN                   # 0 INDICATES FAILURE
                COM
                MASK    OCT15000
                CCS     A
                TCF     ISSWON                  # FAILURE.

ISSWOFF         CAF     BIT1                    # DON'T TURN OFF ISS WARNING IF LAMP TEST
                MASK    IMODES33                # IN PROGRESS.
```

```
                CCS     A
                TC      Q

                CS      BIT1
                EXTEND
                WAND    DSALMOUT
                TC      Q

ISSWON          EXTEND
                QXCH    ITEMP6
                TC      VARALARM                # TELL EVERYONE WHAT CAUSED THE ISS WARNING
                CAF     BIT1
                EXTEND
                WOR     DSALMOUT
                TC      ITEMP6

CAGESUB         CS      BITS6&15                # SET OUTBITS AND INTERNAL FLAGS FOR
                EXTEND                          # SYSTEM TURN-ON OR CAGE.  DISABLE THE
                WAND    CHAN12                  # ERROR COUNTER AND REMOVE THE IMU DELAY COMP.
                CAF     BITS4&5                 # SEND ZERO AND COARSE.
                EXTEND
                WOR     CHAN12

CAGESUB1        CS      DSPTAB +11D             # TURN ON NO ATT LAMP
                MASK    OC40010
                ADS     DSPTAB +11D

CAGESUB2        CS      IMODES30                # SET FLAGS TO INDICATE CAGING OR TURN-ON
                MASK    OCT75                   # AND INHIBIT ALL ISS WARNING INFO
                ADS     IMODES30

                CS      IMODES33                # DISABLE DAP AUTO AND HOLD MODES
                MASK    BIT6
                ADS     IMODES33

                TC      Q

IMUFAIL         EQUALS  SETISSW
ICDUFAIL        EQUALS  SETISSW
```

```
# JUMP TABLES AND CONSTANTS.

IFAILJMP        TCF     ITURNON                 # CHANNEL 30 DISPATCH.
                TCF     IMUFAIL
                TCF     ICDUFAIL
                TCF     IMUCAGE
30RDMSK         OCT     76400                   # (BIT 10 NOT SAMPLED HERE).
                TCF     IMUOP

C33JMP          TCF     PIPFAIL                 # CHANNEL 33 DISPATCH.
                TCF     DNTMFAST
                TCF     UPTMFAST

# SUBROUTINE TO SKIP IF LAMP TEST NOT IN PROGRESS.
LAMPTEST        CS      IMODES33                # BIT 1 OF IMODES33 = 1 IF LAMP TEST IN
                MASK    BIT1                    # PROGRESS.
                CCS     A
                INCR    Q
                TC      Q

33RDMSK         EQUALS  PRIO16
OC40010         OCT     40010
OCT54           OCT     54
OCT75           OCT     75
OCT272          OCT     00272
BITS7&8         OCT     300
OCT1720         OCT     1720
OCT740          OCT     00740
OCT15000        EQUALS  PRIO15
OCT77000        OCT     77000
BITS6&15        OCT     40040
-BIT10          OCT     -1000

90SECS          DEC     9000
120MS           =       OCT14                   # (DEC12)
GLOCKOK         EQUALS  RESUME
```

```
# PROGRAM NAME:  RRAUTCHK
#
# FUNCITONAL DESCRIPTION:
# RRAUTCHK IS THE RENDEZFOUS RADAR INBIT MONITOR.  INITIALLY THE RR
# POWER ON AUTO (CHAN 33 BIT 2) INBIT IS CHECKED.  IF NO CHANGE, THE
# PROGRAM EXITS TO RRCDUCHK.  IF A CHANGE, RADMOES IS UPDATED
# AND A CHECK MADE IF RR POWER HAS JUST COME ON.  IF JUST OFF, A CHECK
# IS MADE TO SEE IF A PROGRAM WAS USING THE RR (STATE BIT 7).  IF NO,
# THE PROGRAM EXITS TO RRCDUCHK.  IF YES, PROGRAM ALARM 00514
# IS REQUESTED BEFORE EXITING TO RRCDUCHK.  IF RR POWER HAS JUST COME
# ON, A CHECK IS MADE TO SEE IF A PROGRAM WAS USING THE RR (STATE BIT 7)
# SEQUENCE.  IF NO, RADMODES IS UPDATED TO INDICATE RR CDU ZERO AND
# RR TURN-ON SEQUENCE (BITS 13, 1).  A 10 MILLISECOND WAITLIST CALL
# IS THEN SET FOR RRTURNON BEFORE THE PROGRAM EXITS TO NORRGMON.
#
# CALLING SEQUENCE:
# T4RUPT EVERY 480 MILLISECONDS
#
# ERASABLE INITIALIZATION REQUIRED:
# RADMODES, STATE.
#
# SUBROUTINES CALLED:
# WAITLIST.
#
# JOBS OR TASKS INITIATED:
# RRTURNON
#
# ALARMS:  PROGRAM ALARM 00514 -- RADAR GOES OUT OF AUTO MODE WHILE BEING
# USED
#
# EXIT:  RRCDUCHK, NORRGMON

RRAUTCHK        CA      RADMODES                        # SEE IF CHANGE IN RR AUTO MODE BIT.
                EXTEND
                RXOR    CHAN33
                MASK    AUTOMBIT
                EXTEND
                BZF     RRCDUCHK

                LXCH    RADMODES                        # UPDATE RADMODES.
                EXTEND
                RXOR    LCHAN
                MASK    OCT05776                        # CLR CONT. DES., REMODE, REPOS, CDUZERO,
                TS      RADMODES                        # AND TURNON BITS.
                MASK    BIT2                            # SEE IF JUST ON.
                CCS     A
                TCF     RRCDUCHK -3                     # OFF.  GO DISABLE RR CDU ERROR COUNTERS.
                CA      OCT10001                        # SET RRCDUZRO AND TURNON BITS.
                ADS     RADMODES
```

```
                CAF     ONE
                TC      WAITLIST
                EBANK=  LOSCOUNT
                2CADR   RRTURNON

                TCF     NORRGMON

OCT05776        OCT     5776
```

```
# PROGRAM NAME: RRCDUCHK
#
# FUNCTIONAL DESCRIPTION:
# RRCDUCHK CHECKS FOR RR CDU FAIL (CHAN 30 BIT 7).  INITIALLY THE
# RR CDU FAIL BIT IS SAMPLED (CHAN 30 BIT 7).  IF NO CHANGE, THE
# PROGRAM EXITS TO RRGIMON.  IF A CHANGE, THE RR AUTO MODE
# (RADMODES BIT 2) BIT IS CHECKED.  IF NOT IN RR AUTO MODE, THE
# PROGRAM EXITS TO NORRGMOON.  IF IN AUTO MODE, RADMODES BIT 7
# (RR CDU OK) IS UPDATED AND IF P-20 IS OPERATING PROGRAM ALARM 00515 IS
# REQUESTED.  CONTROL IS TRANSFERRED TO SETTRKF TO UPDATE
# THE TRACKER FAIL LAMP (DSPTAB+11D BIT 8).  CONTROL RETURNS TO
# RRGIMON.
#
# CALLING SEQUENCE:
# EVERY 480 MILLISECONDS FROM RRAUTCHK (VIA T4RUPT) UNLESS A
# TURN-ON SEQUENCE HAS JUST BE INITIATED.
#
# ERASABLE INITIALIZATION REQUIRED:
# RADMODES
#
# SUBROUTINES CALLED:
# SETTRKF
#
# JOBS OR TASKS INITIATED:
# NONE
#
# ALARMS:
# TRACKER FAIL
# PROGRAM ALARM 00515 -- RRCDU FAIL DURING P-20
#
# EXIT:
# RRGIMON, NORRGMON

 -3             CS      BIT2
                EXTEND
                WAND    CHAN12                  # AT TURNON, DISABLE CDU ERROR COUNTERS.

RRCDUCHK        CA      RADMODES                # LAST SAMPLED BIT IN RADMODES.
                EXTEND
                RXOR    CHAN30
                MASK    RCDUFBIT
                EXTEND
                BZF     RRGIMON

                CAF     AUTOMBIT                # IF RR NOT IN AUTO MODE, DON'T CHANGE BIT
                MASK    RADMODES                # 7 OF RADMODES.  IF THIS WERE NOT DONE,
                CCS     A                       # THE TRACKER FAIL MIGHT COME ON WHEN
                TCF     NORRGMON                # JUST READING LR DATA.

                CAF     RCDUFBIT                # SET BIT 7 OF RADMODES FOR SETTRKF.
```

```
                LXCH    RADMODES                # UPDATE RADMODES.
                EXTEND
                RXOR    L
                TS      RADMODES

                CA      RADMODES                # DID RR CDU FAIL
                MASK    RCDUFBIT
                CCS     A
                TCF     TRKFLCDU                # NO
                CS      FLAGWRD0                # RNDVFLG P20 OR P22 OPERATING
                MASK    RNDVZBIT
                CCS     A
                TCF     TRKFLCDU                # NO
                TC      ALARM                   # YES
                OCT     00515
TRKFLCDU        TC      SETTRKF                 # UPDATE TRAKER FAIL LAMP ON DSKY.
```

```
# PROGRAM NAME:  RRGIMON
#
# FUNCTIONAL DESCRIPTION:
# RRGIMON IS THE RR GIMBAL LIMIT MONITOR.  INITIALLY THE FOLLOWING IS
# CHECKED:  REMOD, RR CDU'S BEING ZEROED, REPOSITION, AND RR
# NOT IN AUTO MODE (RADMODES BITS 14, 13, 11, 2).  IF ANY OF THESE
# EXIST THE PROGRAM EXITS TO GPMATRIX.  IF NONE ARE PRESENT RRLIMCHK
# IS CALLED TO SEE IF THE PRESENT RR CDU ANGLES (OPTY, OPTX) ARE WITHIN
# THE LIMITS OF THE CURRENT MODE.  IF WITHIN LIMITS, THE PROGRAM EXITS
# TO NORRGMON.  IF NOT WITHIN LIMITS, THE REPOSITION FLAG (RADMODES
# BIT 11) IS SET, THE RR AUTO TRACKER AND RR ERROR COUNTER
# (CHAN 12 BITS 14, 2) ARE DISABLED, AND A 20 MILLISECOND WAITLIST
# CALL IS SET FOR DORREPOS AFTER WHICH THE PROGRAM EXITS TO NORRGMON.
#
# CALLING SEQUENCE:
# EVERY 480 MILLISECONDS FROM RRCDUCHK (VIA T4RUPT) UNLESS TURN-ON
# HAS JUST BEEN INITIATED VIA RRAUTCHK OR IF THERE HAS BEEN A CHANGE IN
# THE RR CDU FAIL BIT (CHAN 30 BIT 7) AND THE RR IS NOT IN THE AUTO MODE
# (RADMODES BIT 2).
#
# ERASABLE INITIALZATION:  RADMODES
#
# SUBROUTINES CALLED:
# RRLIMCHK, WAITLIST
#
# JOBS OR TASKS INITIATED:
# DORREPOS
#
# ALARMS:
# NONE
#
# EXIT:
# NORRGMON

RRGIMON         CAE     FLAGWRD5                # IS NO ANGLE MONITOR FLAG SET
                MASK    NORRMBIT
                CCS     A
                TCF     NORRGMON                # YES -- SKIP LIMIT CHECK
                CS      FLAGWRD7                # IS SERVICER RUNNING?
                MASK    AVEGFBIT
                CCS     A
                TCF     +5                      # NO. DO R25
                CA      FLAGWRD6                # YES. IS MUNFLAG SET?
                MASK    MUNFLBIT
                CCS     A
                TCF     NORRGMON                # YES. DON'T DO R25
 +5             CAF     OCT32002                # INHIBIT BY REMODE, ZEROING, MONITOR.
                MASK    RADMODES                # OR RR NOT IN AUTO.
                CCS     A
                TCF     NORRGMON
```

```
                TC      RRLIMCHK                        # SET IF ANGLES IN LIMITS.
                ADRES   CDUT

                TCF     MONREPOS

                TCF     NORRGMON                        # (ADDITIONAL CODING MAY GO HERE).

MONREPOS        CAF     REPOSBIT                        # SET FLAG TO SHOW REPOSITION IN PROGRESS.
                ADS     RADMODES

                CS      OCT20002                        # DISABLE TRACKER AND ERROR COUNTER.
                EXTEND
                WAND    CHAN12

                CAF     TWO
                TC      WAITLIST
                EBANK=  LOSCOUNT
                2CADR   DORREPOS

                TCF     NORRGMON

OCT32002        OCT     32002
OCT20002        OCT     20002
OCT02100        OCT     02100                           # P20, P22 MASK BITS.
```

```
# PROGRAM NAME:  GPMATRIX (DAPT4S) MCD. NO. 2 DATE: OCTOBER 27, 1966
#
# AUTHOR:  JOHNATHAN D. ADDLELSTON (ADAMS ASSOCIATES)
#
# MODIFIED:  7FEB. 1968 BY P. S. WEISSMAN TO DELETE COMPUTATION OF MR12 AND MR13, WHICH ARE NO LONGER REQUIRED.
#
# THIS PROGRAM CALCULATES ALL THE SINGLE-PRECISION MATRIX ELEMENTS WHICH ARE USED BY LEM DAP TO TRANSFORM VECTORS
# FROM GIMBAL TO PILOT (BODY) AXES AND BACK AGAIN.  THESE ELEMENTS ARE USED EXCLUSIVELY BY BASIC LANGUAGE ROUTINES
# AND THEREFORE ARE NOT ARRAYED FOR USE BY INTERPRETIVE PROGRAMS.
#
# CALLING SEQUENCE:  GPMATRIX IS TRANSFERRED TO FROM DAPT4S AND IS THUS EXECUTED 4 TIMES A SECOND BY T4RUPT.
# DAPT4S IS LISTED IN T4JUMP TABLE TWICE EXPLICITLY AND ALSO OCCURS AFTER RRAUTCHK (WHICH IS ALSO LISTED TWICE).
#
# SUBROUTINES CALLED: SPSIN, SPCOS.
#
# NORMAL EXIT MODE:  TCF RESUME
#
# ALARM AND ABORT MODES:  NONE.
#
# INPUT: CDUX, CDUY, CDUZ.
#
# OUTPUT:  M11, M21, M32, M22, M32.
#
# AOG = CDUX, AIG = CDUY, AMG = CDUZ: MNEMONIC IS : OIM = XYZ
#
#               *       *      SING(MG)                0                       1       *
#               M   =   *      COS(MG)COS(OG)          SIN(OG)                 0       *
#                GP     *      -COS(MG)SIN(OG)         COS(OG)                 0       *
#
#               *       *      0                       COS(OG)/COS(MG)         -SIN(OG)/COS(MG)        *
#               M   =   *      0                       SIN(OG)                 COS(OG)                 *
#                PG     *      1                       -SIN(MG)COS(OG)/COS(MG) SIN(MG)SIN(OG)/COS(MG)  *

                EBANK=  M11
DAPT4S          EQUALS  GPMATRIX

# T4RUPT DAP LOGIC:

GPMATRIX        CAE     CDUZ                    # SINGLE ENTRY POINT
                TC      SPSIN                   # SIN(CDUZ) = SIN(MG)
                TS      M11                     # SCALED AT 1

                CAE     CDUZ
                TC      SPCOS                   # COS(CDUZ) = COS(MG)
                TS      COSMG                   # SCALED AT 1 (ONLY A FACTOR)

                CAE     CDUX
                TC      SPSIN                   # SIN(CDUX) = SIN(OG)
                TS      M22                     # SCALED AT 1 (ALSO IS MR22)

                CS      M22
```

```
                EXTEND
                MP      COSMG                   # -SIN(OG)COS(MG)
                TS      M31                     # SCALED AT 1

                CAE     CDUX
                TC      SPCOS                   # COS(CDUX) = COS(OG)
                TS      M32                     # SCALED AT 1 (ALSO IS MR23)

                EXTEND
                MP      COSMG                   # COS(OG)COS(MG)
                TS      M21                     # SCALED AT 1

                TC      RESUME

NORRGMON        EQUALS  DAPT4S
ENDDAPT4        EQUALS  RESUME
```

```
# PROGRAM DESCRIPTION
#
# AUTHOR:  J S MILLER
#
# MODIFIED 6 MARCH 1968 BY P S WEISSMAN TO SET UP JOB FOR 1/ACCS WHEN THE MASKS ARE CHANGED.
#
#      THIS ROUTINE IS ATTACHED TO T4RUPT, AND IS ENTERED EVERY 480 MS.  ITS FUNCTION IS TO EXAMINE THE LOW 8 BITS
# OF CHANNEL 32 TO SEE IF ANY ISOLATION-VALVE CLOSURE BITS HAVE APPEARED OR DISAPPEARED (THE CREW IS WARNED OF JET
# FAILURES BY LAMPS LIT BY THE GRUMMAN FAILURE-DETECTION CIRCUITRY; THEY MAY RESPOND BY OPERATING SWITCHES WHICH
# ISOLATE PAIRS OF JETS FROM THE PROPELLANT TANKS AND SET BITS IN CHANNEL 32).  IN THE EVENT THAT CHANNEL 32 BITS
# DIFFER FROM 'PVALVEST', THE RECORD OF ACTIONS TAKEN BY THIS ROUTINE, THE APPROPRIATE BITS IN 'CH5MASK' &
# 'CH6MASK', USED BY THE DAP JET-SELECTION LOGIC, ARE UPDATED, AS IS 'PVALVEST'.  TO SPEED UP & SHORTEN THE
# ROUTINE, NO MORE THAN ONE CHANGE IS ACCEPTED PER ENTRY.  THE HIGHEST-NUMBERED BIT IN CHANNEL 32 WHICH REQUIRES
# ACTION IS THE ONE PROCESSED.
#
#      THE CODING IN THE FAILURE MONITOR HAS BEEN WRITTEN SO AS TO HAVE ALMOST COMPLETE RESTART PROTECTION.  FOR
# EXAMPLE, NO ASSUMPTION IS MADE WHEN SETTING A 'CH5MASK' BIT TO 1 THAT THE PREVIOUS STATE IS 0, ALTHOUGH IT OF
# COURSE SHOULD BE.  ONE CASE WHICH MAY BE SEEN TO EVADE PROTECTION IS THE OCCURRENCE OF A RESTART AFTER UPDATING
# ONE OR BOTH DAP MASK-WORDS BUT BEFORE UPDATING 'PVALVEST', COUPLED WITH A CHANGE IN THE VALVE-BIT BACK TO ITS
# FORMER STATE.  THE CONSEQUENCE OF THIS IS THAT THE NEXT ENTRY WOULD NOT SEE THE CHANGE INCOMPLETELY INCORP-
# ORATED BY THE LAST PASS (BECAUSE IT WENT AWAY AT JUST THE RIGHT TIME), BUT THE DAP MASK-WORDS WILL BE INCORRECT.
# THIS COMBINATION OF EVENTS SEEMS QUITE REMOTE, BUT NOT IMPOSSIBLE UNLESS THE CREW OPERATES THE SWITCHES AT HALF-
# SECOND INTERVALS OR LONGER.  IN ANY EVENT, A DISAGREEMENT BETWEEN REALITY AND THE DAP MASKS WILL BE CURED IF
# THE MISINTERPRETED SWITCH IS REVERSED AND THEN RESTORED TO ITS CORRECT POSITION (SLOWLY).
#
# CALLING SEQUENCE:
#
#               TCF     RCSMONIT        (IN INTERRUPT MODE, EVERY 480 MS.)
#
# EXIT:  TCF  RCSMONEX  (ALL PATHS EXIT VIA SUCH AN INSTRUCTION)
RCSMONEX        EQUALS  RESUME

# ERASABLE INITIALIZATION REQUIRED:
#
#          VIA FRESH START:  PVALVEST          = +0  (ALL JETS ENABLED)
#                            CH5MASK, CH6MASK  = +0  (ALL JETS OK)
#
# OUTPUT:  CH5MASK & CH6MASK UPDATED  (1'S WHERE JETS NOT TO BE USED, IN CHANNEL 5 & 6 FORMAT)
#          PVALTEST UPDATED  (1.5 WHEN VALVE CLOSURES HAVE BEEN TRANSLATED INTO CH5MASK & CH6MASK; CHAN 32 FORMAT)
#          JOB TO DO 1/ACCS.
#
# DEBRIS:  A, L, Q AND DEBRIS OF NOVAC.
#
# SUBROUTINE CALLED:  NOVAC.

                EBANK=  CH5MASK

                BANK    23
                SETLOC  RCSMONT
                BANK
```

```
                COUNT*  $$/T4RCS

RCSMONIT        EQUALS  RCSMON

RCSMON          CS      ZERO
                EXTEND
                RXOR    CHAN32          # PICK UP + INVERT INVERTED CHANNEL 32.
                MASK    LOW8            # KEEP JET-FAIL BITS ONLY.
                TS      Q

                CS      PVALVEST        #       -   -
                MASK    Q               # FORM PC + PC.
                TS      L               #   (P = PREVIOUS ISOLATION VALVE STATE,
                CS      Q               #    C = CURRENT VALVE STATE (CH 32)).
                MASK    PVALVEST
                ADS     L               # RESULT NZ INDICATES ACTION REQUIRED.

                EXTEND
                BZF     RCSMONEX        # QUIT IF NO ACTION REQUIRED.

                EXTEND
                MP      BIT7            # MOVE BITS 8 - 1 OF A TO 14 - 7 OF L.
                XCH     L               # ZERO TO L IN THE PROCESS.

 -3             INCR    L
                DOUBLE                  # BOUND TO GET OVERFLOW IN THIS LOOP.
                OVSK                    # SINCE WE ASSURED INITIAL NZ IN A.
                TCF     -3

                INDEX   L
                CA      BIT8 -1         # SAVE THE RELEVANT BIT (8 - 1).
                TS      Q
                MASK    PVALVEST        # LOOK AT PREVIOUS VALVE STATE BIT.
                CCS     A
                TCF     VOPENED         # THE VALVE HAS JUST BEEN OPENED.

                CS      CH5MASK         # THE VALVE HAS JUST BEEN CLOSED.
                INDEX   L
                MASK    5FAILTAB
                ADS     CH5MASK         # SET INHIBIT BIT FOR CHANNEL 5 JET.

                CS      CH6MASK
                INDEX   L
                MASK    6FAILTAB
                ADS     CH6MASK         # SET INGIBIT BIT FOR CHANNEL 6 JET.

                CA      Q
                ADS     PVALVEST        # RECORD ACTION TAKEN.

                TCF     1/ACCFIX        # SET UP 1/ACCJOB AND EXIT.
```

```
VOPENED         INDEX   L                 # A VALVE HAS JUST BEEN OPENED.
                CS      5FAILTAB
                MASK    CH5MASK
                TS      CH5MASK           # REMOVE INHIBIT BIT FOR CHANNEL 5 JET.

                INDEX   L
                CS      6FAILTAB
                MASK    CH6MASK
                TS      CH6MASK           # REMOVE INHIBIT BIT FOR CHANNEL 6 JET.

                CS      Q
                MASK    PVALVEST
                TS      PVALVEST          # RECORD ACTION TAKEN.

1/ACCFIX        CAF     PRIO27            # SET UP 1/ACCS SO THAT THE SWITCH CURVES
                TC      NOVAC             #   FOR TJETLAW CAN BE MODIFIED IF CH5MASK
                EBANK=  AOSQ              #   HAS BEEN ALTERED.
                2CADR   1/ACCJOB

                TCF     RCSMONEX          # EXIT.


5FAILTAB        EQUALS  -1                # CH 5 JET BIT CORRESPONDING TO CH 32 BIT:
                OCT     00040             # 8
                OCT     00020             # 7
                OCT     00100             # 6
                OCT     00200             # 5
                OCT     00010             # 4
                OCT     00001             # 3
                OCT     00004             # 2
                OCT     00002             # 1

6FAILTAB        EQUALS  -1                # CH 6 JET BIT CORRESPONDING TO CH 32 BIT:
                OCT     00010             # 8
                OCT     00020             # 7
                OCT     00004             # 6
                OCT     00200             # 5
                OCT     00001             # 4
                OCT     00002             # 3
                OCT     00040             # 2
                OCT     00100             # 1
```

```
                BANK    22
                SETLOC  DOWNTELM
                BANK

                EBANK=  DNTMBUFF

# SPECIAL DOWNLINK OP CODES
#       OP CODE         ADDRESS (EXAMPLE)       SENDS...                BIT 15          BITS 14-12      BITS 11-0
#       -------         -----------------       --------                ------          ----------      ---------
#       1DNADR          TIME2                   (2 AGC WDS)             0               0               ECADR
#       2DNADR          TEPHEM                  (4 AGC WDS)             0               1               ECADR
#       3DNADR          VGBODY                  (6 AGC WDS)             0               2               ECADR
#       4DNADR          STATE                   (8 AGC WDS)             0               3               ECADR
#       5DNADR          UPBUFF                  (10 AGC WDS)            0               4               ECADR
#       6DNADR          DSPTAB                  (12 AGC WDS)            0               5               ECADR
#       DNCHAN          30                      CHANNELS                0               7               CHANNEL
#                                                                                                       ADDRESS
#       DNPTR           NEXTLIST                POINTS TO NEXT          0               6               ADRES
#                                               LIST
#
# DOWNLIST FORMAT DEFINITIONS AND RULES --
# 1. END OF A LIST = -XDNADR (X = 1 TO 6), -DNPTR, OR -DNCHAN.
# 2. SNAPSHOT SUBLIST = LIST WHICH STARTS WITH A -1DNADR.
# 3. SNAPSHOT SUBLIST CAN ONLY CONTAIN 1DNADRS.
# 4. TIME2 1DNADR MUST BE LOCATED IN THE CONTROL LIST OF A DOWNLIST.
# 5. ERASABLE DOWN TELEMETRY WORDS SHOULD BE GROUPED IN SEQUENTIAL
#    LOCATIONS AS MUCH AS POSSIBLE TO SAVE STORAGE USED BY DOWNLINK LISTS.

                COUNT*  $$/DLIST
ERASZERO        EQUALS  7
UNKNOWN         EQUALS  ERASZERO
SPARE           EQUALS  ERASZERO                        # USE SPARE TO INDICATE AVAILABLE SPACE
LOWIDCOD        OCT     77340                           # LOW ID CODE

NOMDNLST        EQUALS  LMCSTADL                        # FRESH START AND POST P27 DOWNLIST

AGSLIST         EQUALS  LMAGSIDL

UPDNLIST        EQUALS  LMAGSIDL                        # UPDATE PROGRAM (P27) DOWNLIST
```

```
# LM ORBITAL MANEUVERS LIST
#
# -------------------- CONTROL LIST -------------------------

LMORBMDL        EQUALS                          # SEND ID BY SPECIAL CODING
                DNPTR   LMORBM01                # COLLECT SNAPSHOT
                6DNADR  DNTMBUFF                # SEND SNAPSHOT
                1DNADR  DELLT4                  # DELLT4,+1
                3DNADR  RTARG                   # RTARG,+1...+5
                1DNADR  ELEV                    # ELEV,+1
                1DNADR  TEVENT                  # TEVENT,+1
                6DNADR  REFSMMAT                # REFSMMAT +0...+11D
                1DNADR  TCSI                    # TCSI,+1
                3DNADR  DELVEET1                # DELVEET1 +0...+5
                3DNADR  VGTIG                   # VGTIG +0...+5
                1DNADR  DNLRVELZ                # DNLRVELZ,DNLRALT
                1DNADR  TPASS4                  # TPASS4,+1
                DNPTR   LMORBM02                # COMMON DATA
                1DNADR  TIME2                   # TIME2/1
                DNPTR   LMORBM03                # COLLECT SNAPSHOT
                6DNADR  DNTMBUFF                # SEND SNAPSHOT
                DNPTR   LMORBM04                # COMMON DATA
                2DNADR  POSTORKU                # POSTORKU,NEGTORKU,POSTORKV,NEGTORKV
                1DNADR  SPARE
                1DNADR  TCDH                    # TCDH,+1
                3DNADR  DELVEET2                # DELVEET2 +0...+5
                1DNADR  TTPI                    # TTPI,+1
                3DNADR  DELVEET3                # DELVEET3 +0...+5
                1DNADR  DNRRANGE                # DNRRANGE,DNRRDOT
                2DNADR  DNLRVELX                # DNLRVELX,DNLRVELY,DNLRVELZ,DNLRALT
                1DNADR  DIFFALT                 # DIFFALT,+1
                1DNADR  LEMMASS                 # LEMMASS,CSMMASS
                1DNADR  IMODES30                # IMODES30,IMODES33
                1DNADR  TIG                     # TIG,+1
                DNPTR   LMORBM05                # COMMON DATA
                DNPTR   LMORBM06                # COMMON DATA
                1DNADR  SPARE                   # FORMERLY PIF
                -1DNADR TGO                     # TGO,+1

# -------------------- SUB-LISTS --------------------------

LMORBM01        -1DNADR R-OTHER +2              # R-OTHER +2,+3         SNAPSHOT
                1DNADR  R-OTHER +4              # R-OTHER +4,+5
                1DNADR  V-OTHER                 # V-OTHER,+1
                1DNADR  V-OTHER +2              # V-OTHER +2,+3
                1DNADR  V-OTHER +4              # V-OTHER +4,+5
                1DNADR  T-OTHER                 # T-OTHER,+1
                -1DNADR R-OTHER                 # R-OTHER +0,+1

LMORBM02        2DNADR  REDOCTR                 # REDOCTR,THETAD,+1,+2  COMMON DATA
```

```
                                1DNADR  RSBBQ                           # RSBBQ,+1
                                2DNADR  OMEGAP                          # OMEGAP,OMEGAQ,OMEGAR,GARBAGE
                                2DNADR  CDUXD                           # CDUXD,CDUYD,CDUZD,GARBAGE
                                2DNADR  CDUX                            # CDUX,CDUY,CDUZ,CDUT
                                6DNADR  STATE                           # STATE +0...+11D (FLAGWORDS)
                                -6DNADR DSPTAB                          # DSPTAB TABLES

LMORBM03                        -1DNADR RN +2                           # RN +2,+3              SNAPSHOT
                                1DNADR  RN +4                           # RN +4,+5
                                1DNADR  VN                              # VN,+1
                                1DNADR  VN +2                           # VN +2,+3
                                1DNADR  VN +4                           # VN +4,+5
                                1DNADR  PIPTIME                         # PIPTIME,+1
                                -1DNADR RN                              # RN,+1

LMORBM04                        2DNADR  OMEGAPD                         # OMEGAPD,OMEGAQD,OMEGARD,GARBAGE
                                3DNADR  CADRFLSH                        # CADRFLSH,+1,+2,FAILREG,+1,+2
                                -1DNADR RADMODES                        # RADMODES,DAPBOOLS     COMMON DATA

LMORBM05                        2DNADR  OMEGAP                          # OMEGAP,OMEGAQ,OMEGAR,GARBAGE
                                2DNADR  CDUXD                           # CDUXD,CDUYD,CDUZD,GARBAGE
                                2DNADR  CDUX                            # CDUX,CDUY,CDUZ,CDUT
                                1DNADR  ALPHAQ                          # ALPHAQ,ALPHAR         COMMON DATA
                                1DNADR  POSTORKP                        # POSTORKP,NEGTORKP
                                DNCHAN  11                              # CHANNELS 11,12
                                DNCHAN  13                              # CHANNELS 13,14
                                DNCHAN  30                              # CHANNELS 30,31
                                -DNCHAN 32                              # CHANNELS 32,33

LMORBM06                        1DNADR  PIPTIME1                        # PIPTIME,+1            COMMON DATA
                                -3DNADR DELV                            # DELV +0...+5

# --------------------------------------------------------------------
```

```
# LM COAST AND ALIGNMENT DOWNLIST
#
# ---------------------- CONTROL LIST --------------------------------

LMCSTADL        EQUALS                                  # SEND ID BY SPECIAL CODING
                DNPTR    LMCSTA01                       # COLLECT SNAPSHOT
                6DNADR   DNTMBUFF                       # SEND SNAPSHOT
                1DNADR   AGSK                           # AGSK,+1
                1DNADR   TALIGN                         # TALIGN,+1
                2DNADR   POSTORKU                       # POSTORKU,NEGTORKU,POSTORKV,NEGTORKV
                1DNADR   DNRRANGE                       # DNRRANGE,DNRRDOT
                1DNADR   TEVENT                         # TEVENT,+1
                6DNADR   REFSMMAT                       # REFSMMAT +0...+11D
                1DNADR   AOTCODE                        # AOTCODE,GARBAGE
                3DNADR   RLS                            # RLS +0...+5
                2DNADR   DNLRVELX                       # DNLRVELX,DNLRVELY,DNLRVELZ,DNLRALT
                DNPTR    LMCSTA06                       # COMMON DATA
                DNPTR    LMCSTA02                       # COMMON DATA
                1DNADR   TIME2                          # TIME2/1
                DNPTR    LMCSTA03                       # COLLECT SNAPSHOT
                6DNADR   DNTMBUFF                       # SEND SNAPSHOT
                DNPTR    LMCSTA04                       # COMMON DATA
                DNPTR    LMCSTA07                       # COMMON DATA
                2DNADR   DNLRVELX                       # DNLRVELX,DNLRVELY,DNLRVELZ,DNLRALT
                2DNADR   CDUS                           # CDUS,PIPAX,PIPAY,PIPAZ
                1DNADR   LASTYCMD                       # LASTYCMD,LASTXCMD
                1DNADR   LEMMASS                        # LEMMASS,CSMMASS
                1DNADR   IMODES30                       # IMODES30,IMODES33
                1DNADR   TIG                            # TIG,+1
                DNPTR    LMCSTA05                       # COMMON DATA
                -6DNADR  DSPTAB                         # DSPTAB +0...+11D TABLE

# ---------------------- SUB-LISTS -------------------------

LMCSTA01        EQUALS   LMORBM01                       # COMMON DOWNLIST DATA
LMCSTA02        EQUALS   LMORBM02                       # COMMON DOWNLIST DATA
LMCSTA03        EQUALS   LMORBM03                       # COMMON DOWNLIST DATA
LMCSTA04        EQUALS   LMORBM04                       # COMMON DOWNLIST DATA
LMCSTA05        EQUALS   LMORBM05                       # COMMON DOWNLIST DATA

LMCSTA06        2DNADR   X789                           # X789 +0...+3         COMMON DATA
                -1DNADR  LASTYCMD                       # LASTYCMD,LASTXCMD
LMCSTA07        3DNADR   OGC                            # OGC,+1,IGC,+1,MGC,+1  COMMON DATA
                1DNADR   BESTI                          # BESTI,BESTJ
                3DNADR   STARSAV1                       # STARSAV1 +0...+5
                -3DNADR  STARSAV2                       # STARSAV2 +0...+5
```

```
# ----------------------------------------------------------
```

```
# LM RENDEZVOUS AND PRE-THRUST DOWNLIST
#
# -------------------- CONTROL LIST -----------------------

LMRENDDL        EQUALS                              # SEND ID BY SPECIAL CODING
                DNPTR    LMREND01                   # COLLECT SNAPSHOT
                6DNADR   DNTMBUFF                   # SEND SNAPSHOT
                DNPTR    LMREND07                   # COLLECT SNAPSHOT
                4DNADR   DNTMBUFF                   # SEND SNAPSHOT
                1DNADR   DELLT4                     # DELLT4,+1
                3DNADR   RTARG                      # RTARG +0...+5
                3DNADR   DELVSLV                    # DELVSLV +0...+5
                1DNADR   TCSI                       # TCSI,+1
                3DNADR   DELVEET1                   # DELVEET +0...+5
                1DNADR   SPARE
                1DNADR   TPASS4                     # TPASS4,+1
                DNPTR    LMREND06                   # COMMON DATA
                DNPTR    LMREND02                   # COMMON DATA
                1DNADR   TIME2                      # TIME2/1
                DNPTR    LMREND03                   # COLLECT SNAPSHOT
                6DNADR   DNTMBUFF                   # SEND SNAPSHOT
                DNPTR    LMREND04                   # COMMON DATA
                2DNADR   POSTORKU                   # POSTORKU,NEGTORKU,POSTORKV,NEGTORKV
                1DNADR   SPARE
                1DNADR   TCDH                       # TCDH,+1
                3DNADR   DELVEET2                   # DELVEET2 +0...+5
                1DNADR   TTPI                       # TTPI,+1
                3DNADR   DELVEET3                   # DELVEET3 +0...+5
                1DNADR   ELEV                       # ELEV,+1
                2DNADR   CDUS                       # CDUS,PIPAX,PIPAY,PIPAZ
                1DNADR   LASTYCMD                   # LASTYCMD,LASTXCMD
                1DNADR   LEMMASS                    # LEMMASS,CSMMASS
                1DNADR   IMODES30                   # IMODES30,IMODES33
                1DNADR   TIG                        # TIG,+1
                DNPTR    LMREND05                   # COMMON DATA
                1DNADR   DELTAR                     # DELTAR,+1
                1DNADR   CENTANG                    # CENTANG,+1
                1DNADR   NN                         # NN,+1
                1DNADR   DIFFALT                    # DIFFALT,+1
                1DNADR   DELVTPF                    # DELVTPF,+1
                -1DNADR  SPARE

# -------------------- SUB-LISTS -------------------------

LMREND01        EQUALS   LMORBM01                   # COMMON DOWNLIST DATA
LMREND02        EQUALS   LMORBM02                   # COMMON DOWNLIST DATA
LMREND03        EQUALS   LMORBM03                   # COMMON DOWNLIST DATA
```

```
LMREND04        EQUALS  LMORBM04                        # COMMON DOWNLIST DATA
LMREND05        EQUALS  LMORBM05                        # COMMON DOWNLIST DATA
LMREND06        EQUALS  LMCSTA06                        # COMMON DOWNLIST DATA

LMREND07        -1DNADR AIG                             # AIG,AMG                SNAPSHOT
                1DNADR  AOG                             # AOG,TRKMKCNT
                1DNADR  TANGNB                          # TANGNB,+1
                1DNADR  MKTIME                          # MKTIME,+1
                -1DNADR RANGRDOT                        # DNRRANGE,DNRRDOT

# -----------------------------------------------------------
```

```
# LM DESCENT AND ASCENT DOWNLIST

# ---------------------- CONTROL LIST ------------------------

LMDSASDL        EQUALS                          # SEND ID BY SPECIAL CODING
                DNPTR    LMDSAS07               # COLLECT SNAPSHOT
                DNPTR    LMDSAS08               # SEND SNAPSHOT
                1DNADR   TEVENT                 # TEVENT,+1
                3DNADR   UNFC/2                 # UNFC/2 +0...+5
                3DNADR   VGVECT                 # VGVECT +0...+5
                1DNADR   TTF/8                  # TTF/8,+1
                1DNADR   DELTAH                 # DELTAH,+1
                3DNADR   RLS                    # RLS +0...+5
                1DNADR   SPARE
                DNPTR    LMDSAS09               # COMMON DATA
                DNPTR    LMDSAS02               # COMMON DATA
                1DNADR   TIME2                  # TIME2/1
                DNPTR    LMDSAS03               # COLLECT SNAPSHOT
                6DNADR   DNTMBUFF               # SEND SNAPSHOT
                DNPTR    LMDSAS04               # COMMON DATA
                2DNADR   POSTORKU               # POSTORKU,NEGTORKU,POSTORKV,NEGTORKV
                3DNADR   RGU                    # RGU +0...+5
                3DNADR   VGU                    # VGU +0...+5
                3DNADR   LAND                   # LAND +0...+5
                1DNADR   AT                     # AT,+1
                1DNADR   TLAND                  # TLAND,+1
                1DNADR   FC                     # FC,GARBAGE
                1DNADR   LASTYCMD               # LASTYCMD,LASTXCMD
                1DNADR   LEMMASS                # LEMMASS,CSMMASS
                1DNADR   IMODES30               # IMODES30,IMODES33
                1DNADR   TIG                    # TIG,+1
                DNPTR    LMDSAS05               # COMMON DATA
                DNPTR    LMDSAS06               # COMMON DATA
                1DNADR   PSEUDO55               # PSEUDO55,GARBAGE
                -1DNADR  TTOGO                  # TTOGO,+1

# ---------------------- SUB-LISTS ------------------------

LMDSAS02        EQUALS   LMORBM02               # COMMON DOWNLIST DATA
LMDSAS03        EQUALS   LMORBM03               # COMMON DOWNLIST DATA
LMDSAS04        EQUALS   LMORBM04               # COMMON DOWNLIST DATA
LMDSAS05        EQUALS   LMORBM05               # COMMON DOWNLIST DATA
LMDSAS06        EQUALS   LMORBM06               # COMMON DOWNLIST DATA

LMDSAS07        -1DNADR  LRZCDUDL               # LRZCDUDL,GARBAGE          SNAPSHOT
                1DNADR   VSELECT                # VSELECT,GARBAGE
                1DNADR   LRVTIMDL               # LRVTIMDL,+1
```

```
                1DNADR  VMEAS                        # VMEAS,+1
                1DNADR  MKTIME                       # MKTIME,+1
                1DNADR  HMEAS                        # HMEAS,+1
                1DNADR  RM                           # RM,+1
                1DNADR  AIG                          # AIG,AMG
                1DNADR  AOG                          # AOG,TRKMKCNT
                1DNADR  TANGNB                       # TANGNB,+1
                1DNADR  MKTIME                       # MKTIME,+1
                -1DNADR LRXCDUDL                     # LRXCDUDL,LRYCDUDL

LMDSAS08        6DNADR  DNTMBUFF                     # SEND SNAPSHOT
                -5DNADR DNTMBUFF +12D

LMDSAS09        EQUALS  LMCSTA06                     # COMMON DOWNLIST DATA

# ---------------------------------------------------------
```

```
# LM LUNAR SURFACE ALIGN DOWNLIST

# ---------------------- CONTROL LIST ---------------------

LMLSALDL        EQUALS                          # SEND ID BY SPECIAL CODING
                DNPTR    LMLSAL01               # COLLECT SNAPSHOT
                6DNADR   DNTMBUFF               # SEND SNAPSHOT
                DNPTR    LMLSAL07               # COLLECT SNAPSHOT
                4DNADR   DNTMBUFF               # SEND SHAPSHOT
                1DNADR   TALIGN                 # TALIGN,+1
                6DNADR   REFSMMAT               # REFSMMAT +0...+11D
                6DNADR   YNBSAV                 # YNBSAV +0...+5,SNBSAV +0...+5
                DNPTR    LMLSAL08               # COMMON DATA
                DNPTR    LMLSAL02               # COMMON DATA
                1DNADR   TIME2                  # TIME2/1
                DNPTR    LMLSAL03               # COLLECT SNAPSHOT
                6DNADR   DNTMBUFF               # SEND SHAPSHOT
                DNPTR    LMLSAL04               # COMMON DATA
                DNPTR    LMLSAL09               # COMMON DATA
                3DNADR   GSAV                   # GSAV +0...+5
                1DNADR   AGSK                   # AGSK,+1
                1DNADR   LASTYCMD               # LASTYCMD,LASTXCMD
                1DNADR   LEMMASS                # LEMMASS,CSMMASS
                1DNADR   IMODES30               # IMODES30,IMODES33
                1DNADR   TIG                    # TIG,+1
                DNPTR    LMLSAL05               # COMMON DATA
                DNPTR    LMLSAL06               # COMMON DATA
                1DNADR   SPARE
                -1DNADR  SPARE

# ---------------------- SUB-LISTS ----------------------

LMLSAL01        EQUALS   LMORBM01               # COMMON DOWNLIST DATA
LMLSAL02        EQUALS   LMORBM02               # COMMON DOWNLIST DATA
LMLSAL03        EQUALS   LMORBM03               # COMMON DOWNLIST DATA
LMLSAL04        EQUALS   LMORBM04               # COMMON DOWNLIST DATA
LMLSAL05        EQUALS   LMORBM05               # COMMON DOWNLIST DATA
LMLSAL06        EQUALS   LMORBM06               # COMMON DOWNLIST DATA
LMLSAL07        EQUALS   LMREND07               # COMMON DOWNLIST DATA
LMLSAL08        EQUALS   LMCSTA06               # COMMON DOWNLIST DATA
LMLSAL09        EQUALS   LMCSTA07               # COMMON DOWNLIST DATA
```

```
# -----------------------------------------------------
```

```
# LM AGS INITIALIZATION AND UPDATE DOWNLIST

# ---------------------- CONTROL LIST --------------------

LMAGSIDL        EQUALS                                  # SEND IO BY SPECIAL CODING
                3DNADR  AGSBUFF +0                      # AGSBUFF +0...+5
                1DNADR  AGSBUFF +12D                    # AGSBUFF +12D,GARBAGE
                3DNADR  AGSBUFF +1                      # AGSBUFF +1...+6
                1DNADR  AGSBUFF +13D                    # AGSBUFF +13D, GARBAGE
                3DNADR  AGSBUFF +6                      # AGSBUFF +6...+11
                1DNADR  AGSBUFF +12D                    # AGSBUFF +12D,GARBAGE
                3DNADR  AGSBUFF +7                      # AGSBUFF +7...+12D
                1DNADR  AGSBUFF +13D                    # AGSBUFF +13D,GARBAGE
                6DNADR  COMPNUMB                        # COMPNUMB,UPOLDMOD,UPVERB,UPCOUNT,
                                                        # UPBUFF +0...+7
                6DNADR  UPBUFF +8D                      # UPBUFF +8D...+19D
                DNPTR   LMAGSI02                        # COMMON DATA
                1DNADR  TIME2                           # TIME2/1
                DNPTR   LMAGSI03                        # COLLECT SNAPSHOT
                6DNADR  DNTMBUFF                        # SEND SNAPSHOT
                DNPTR   LMAGSI04                        # COMMON DATA
                2DNADR  POSTORKU                        # POSTORKU,NEGTORKU,POSTORKV,NEGTORKV
                1DNADR  SPARE
                1DNADR  SPARE
                1DNADR  AGSK                            # AGSK,+1
                6DNADR  UPBUFF                          # UPBUFF +0...+11D
                4DNADR  UPBUFF +12D                     # UPBUFF +12D...+19D
                1DNADR  LEMMASS                         # LEMMASS,CSMMASS
                1DNADR  IMODES30                        # IMODES30,IMODES33
                1DNADR  SPARE
                DNPTR   LMAGSI05                        # COMMON DATA
                -6DNADR DSPTAB                          # DSPTAB +0...+11D

# ---------------------- SUB-LISTS ---------------------

LMAGSI02        EQUALS  LMORBM02                        # COMMON DOWNLIST DATA
LMAGSI03        EQUALS  LMORBM03                        # COMMON DOWNLIST DATA
LMAGSI04        EQUALS  LMORBM04                        # COMMON DOWNLIST DATA
LMAGSI05        EQUALS  LMORBM05                        # COMMON DOWNLIST DATA

# ------------------------------------------------------

DNTABLE         GENADR  LMCSTADL                        # LM COAST AND ALIGN DOWNLIST
                GENADR  LMAGSIDL                        # LM AGS INITIALIZATION/UPDATE DOWNLIST
                GENADR  LMRENDDL                        # LM RENDEZVOUS AND PRE-THRUST DOWNLIST
                GENADR  LMORBMDL                        # LM ORBITAL MANEUVERS DOWNLIST
                GENADR  LMDSASDL                        # LM DESCENT AND ASCENT DOWNLIST
```

```
                GENADR  LMLSALDL                        # LM LUNAR SURFACE ALIGN DOWNLIST

# -----------------------------------------------------
```

```
# PROGRAM NAME:  AGS INITIALIZATION (R47)
#
# WRITTEN BY:  RHODE/KILROY/FOLLETT
#
# MOD NO.:        0
# DATE:           23 MARCH 1967
# MOD BY:         KILROY
#
# MOD NO.:        1
# DATE:           28 OCTOBER 1967
# MOD BY:         FOLLETT
#
# FUNCT. DESC.:  (1) TO PROVIDE THE AGS ABORT ELECTRONICS ASSEMBLY (AEA) WITH THE LEM AND CSM STATE VECTORS
#                (POSITION,VELOCITY,TIME) IN LEM IMU COORDINATES BY MEANS OF THE LGC DIGITAL DOWNLINK.
#
#                (2) TO ZERO THE ICDU, LGC, AND AEA GIMBAL ANGLE COUNTER SIMULTANEOUSLY IN ORDER TO ESTABLISH A
#                COMMON ZERO REFERENCE FOR THE MEASUREMENT OF GIMBAL (EULER) ANGLES WHICH DEFINE LEM ATTITUDE
#                (3) TO ESTABLISH THE GROUND ELAPSED TIME OF AEA CLOCK ZERO.  (IF AN AEA CLOCK ZERO IS
#                REQUESTED DURING THIS PROGRAM
#
# LOG SECTION:   AGS INITIALIZATION
#
# CALLING SEQ:   PROGRAM IS ENTERED WHEN ASTRONAUT KEYS V47E ON DSKY.
#                R47 MAY BE CALLED AT ANY TIME EXCEPT WHEN ANOTHER EXTENDED VERB IS IN PROGRESS
#
# SUBROUTINES
# CALLED:
#
# NORMAL EXIT:   ENDEXT
#
# ALARM/ABORT:   ALARM -- BAD REFSMMAT -- CODE:220
#                OPERATOR ERROR IF V47 SELECTED DURING ANOTHER EXTENDED VERB.
#
# ERASABLES
# USED:          SAMPTIME        (2)     TIME OF :ENTER: KEYSTROKE
#                AGSK            (2)     GROUND ELAPSED TIME OF THE AEA CLOCK :ZERO:
#                AGSBUFF         (14D)   CONTAINS AGS INITIALIZATION DATA (SEE :OUTPUT: BELOW)
#                AGSWORD         (1)     PREVIOUS DOWNLIST SAVED HERE

                EBANK=  AGSBUFF

                BANK    40
                SETLOC  R47
                BANK

                COUNT*  $$/R47

AGSINIT         CAF     REFSMBIT
                MASK    FLAGWRD3                        # CHECK REFSMFLG.
                CCS     A
```

```
                TC      REDSPTEM                        # REFSMMAT IS OK
                TC      ALARM                           # REFSMMAT IS BAD
                OCT     220
                TC      ENDEXT

NEWAGS          EXTEND
                DCA     SAMPTIME                        # TIME OF THE :ENTER: KEYSTROKE
                DXCH    AGSK                            # BECOMES NEW AEA CLOCK :ZERO:

REDSPTEM        EXTEND
                DCA     AGSK
                DXCH    DSPTEMX
AGSDISPK        CAF     V06N16
                TC      BANKCALL                        # R1 = 00XXX. HRS., R2 = 000XX MIN.,
                CADR    GOMARKF                         # R3 = 0XX.XX SEC.
                TC      ENDEXT                          # TERMINATE RETURN
                TC      AGSVCALC                        # PROCEED RETURN
                CS      BIT6                            # IS ENTER VIA A V32
                AD      MPAC
                EXTEND
                BZF     NEWAGS                          # YES, USE KEYSTROKE TIME FOR NEW AGSK

                EXTEND                                  # NO, NEW AGSK LOADED VIA V25
                DCA     DSPTEMX                         # LOADED INTO DSPTEMX BY KEYING
                TC      REDSPTEM -1                     # V25E FOLLOWED BY HRS.,MINS.,SECS.
                                                        # DISPLAY THE NEW K

AGSVCALC        TC      INTPRET
                SET
                        NODOFLAG                        # DON'T ALLOW V37
                SET     EXIT
                        XDSPFLAG

                CAF     V06N16
                TC      BANKCALL
                CADR    EXDSPRET

                TC      INTPRET                         # EXTRAPOLATE LEM AND CSM STATE VECTORS
                RTB                                     # TO THE PRESENT TIME
                        LOADTIME                        # LOAD MPAC WITH TIME2,TIME1
                STCALL  TDEC1                           # CALCULATE LEM STATE VECTOR
                        LEMPREC
                CALL                                    # CALL ROUTINE TO CONVERT TO SM COORDS AND
                        SCALEVEC                        # PROVIDE PROPER SCALING
                STODL   AGSBUFF                         # (LEMPREC AND CSMPREC LEAVE TDEC1 IN TAT)
                        TAT                             # TAT = TIME TO WHICH RATT1 AND VATT1 ARE
                STCALL  TDEC1                           # COMPUTED (CSEC SINCE CLOCK START B-28).
                        CSMPREC                         # CALCULATE CSM STATE VECTOR FOR SAME TIME
                CALL
                        SCALEVEC
```

```
                STODL   AGSBUFF +6
                        TAT
                DSU     DDV                         # CALCULATE AND STORE THE TIME
                        AGSK
                        TSCALE
                STORE   AGSBUFF +12D
                EXIT

                CAF     LAGSLIST
                TS      DNLSTCOD

                CAF     20SEC                       # DELAY FOR 20 SEC WHILE THE AGS
                TC      BANKCALL                    # DOWNLIST IS TRANSMITTED
                CADR    DELAYJOB

                CA      AGSWORD
                TS      DNLSTCOD                    # RETURN TO THE OLD DOWNLIST
                CAF     IMUSEBIT
                MASK    FLAGWRD0                    # CHECK IMUSE FLAG.
                CCS     A
                TC      AGSEND                      # IMU IS BEING USED -- DO NOT ZERO
CKSTALL         CCS     IMUCADR                     # CHECK FOR IMU USAGE WHICH AVOIDS THE
                TCF     +3                          # IMUSE BIT:  I.E., IMU COMPENSATION.
                TCF     +6                          # FREE.  GO AHEAD WITH THE IMU ZERO.
                TCF     +1
 +3             CAF     TEN                         # WAIT .1 SEC AND TRY AGAIN.
                TC      BANKCALL
                CADR    DELAYJOB
                TCF     CKSTALL

 +6             TC      BANKCALL                    # IMU IS NOT IN USE
                CADR    IMUZERO                     # SET IMU ZERO DISCRETE FOR 320 MSECS.
                TC      BANKCALL                    # WAIT 3 SEC FOR COUNTERS TO INCREMENT
                CADR    IMUSTALL
                TC      AGSEND
AGSEND          TC      DOWNFLAG                    # ALLOW V37
                ADRES   NODOFLAG

                CAF     V50N16
                TC      BANKCALL
                CADR    GOMARK3
                TCF     ENDEXT
                TCF     ENDEXT
                TC      ENDEXT

SCALEVEC        VLOAD   MXV
                        VATT1
                        REFSMMAT
                VXSC    VSL2
                        VSCALE
```

```
                VAD     VAD                     # THIS SECTION ROUNDS THE VECTOR, AND
                        AGSRND1                 # CORRECTS FOR THE FACT THAT THE AGS
                        AGSRND2                 # IS A 2 S COMPLEMENT MACHINE WHILE THE
                RTB                             # LGC IS A 1 S COMPLEMENT MACHINE.
                        VECSGNAG
                STOVL   VATT1
                        RATT1
                MXV     VXSC
                        REFSMMAT
                        RSCALE
                VSL8    VAD                     # AGAIN THIS SECTION ROUNDS.  TWO VECTORS
                        AGSRND1                 # ARE ADDED TO DEFEAT ALSIGNAG IN THE
                VAD     RTB                     # CASE OF A HIGH-ORDER ZERO COUPLED WITH
                        AGSRND2                 # A LOW ORDER NEGATIVE PART.
                        VECSGNAG
                LXA,1
                        VATT1
                SXA,1   LXA,1
                        MPAC +1
                        VATT1 +2
                SXA,1   LXA,1
                        MPAC +4
                        VATT1 +4
                SXA,1   RVQ
                        MPAC +6

LAGSLIST        =       ONE
V01N14          VN      0114
V50N00A         VN      5000
V00N25          EQUALS  OCT31
V06N16          VN      0616
V00N34          EQUALS  34DEC
V50N16          VN      5016
TSCALE          2DEC    100 B-10                # CSEC TO SEC SCALE FACTOR
20SEC           DEC     2000
RSCALE          2DEC    3.280839 B-3            # METERS TO FEET SCALE FACTOR
VSCALE          2DEC    3.280839 E2 B-9         # METERS/CS TO FEET/SEC SCALE FACTOR
AGSRND1         2OCT    0000060000
                2OCT    0000060000
                2OCT    0000060000
AGSRND2         2OCT    0000037777
                2OCT    0000037777
```

```
                2OCT    0000037777
                SBANK=  LOWSUPER                    # FOR SUBSEQUENT LOW 2CADRS.
```

```
                BANK    10
                SETLOC  FRANDRES
                BANK

                EBANK=  LST1

                COUNT*  $$/START        # FRESH AND RESTART
SLAP1           INHINT                  # FRESH START.  COMES HERE FROM PINBALL.
                TC      STARTSUB        # SUBROUTINE DOES MOST OF THE WORK

STARTSW         TCF     SKIPSIM         # PATCH....TCF STARTSIM FOR SIMULATION
STARTSIM        CAF     BIT14
                TC      FINDVAC
SIM2CADR        OCT     77777           # PATCH 2CADR (AND EBANK DESIGNATION) OF
                OCT     77777           # SIMULATION START ADDRESS.

SKIPSIM         CA      DSPTAB +11D     # TURN OFF ALL DSPTAB +11D LAMPS
                MASK    BITS4&6         # EXCEPT THE GIMBAL LOCK & NO ATT ONLY ON
                AD      BIT15           # REQUESTED FRESH START.
                TS      DSPTAB +11D

                CA      BIT12           # INITIALIZE DOWNLINK EARASABLE MEMORY
                TS      DUMPCNT         # DUMP FOR ONE PASS

                CA      ZERO
                TS      ERCOUNT
                TS      FAILREG
                TS      FAILREG +1
                TS      FAILREG +2
                TS      REDOCTR

                CS      PRIO12
                TS      DSRUPTSW

DOFSTART        CAF     BIT14           # INSURE ENGINE IS OFF.
                EXTEND
                WRITE   DSALMOUT
                CS      ZERO
                TS      THRUST

DOFSTRT1        CAF     FOUR
                TS      RCSFLAGS        # INITIALIZE ATTITUDE ERROR DISPLAYS.
                CA      PRIO30
                TS      RESTREG         # SUPER BANK PRIORITY FOR DISPLAYS.

                CA      ZERO
                TS      ABDELV          # DAP INITIALIZATION
                TS      NVSAVE
                TS      EBANKTEM
```

```
                TS      CH5MASK
                TS      CH6MASK
                TS      PVALVEST        # FOR RCS FAILURE MONITOR
                TS      ERESTORE        # ***** MUST NOT BE REMOVED FROM DOFSTART
                TS      SMODE           # ***** MUST NOT BE REMOVED FROM DOFSTART
                TS      DNLSTCOD        # SELECT P00 DOWNLIST
                TS      AGSWORD         # ALLOW AGS INITIALIZATION
                TS      UPSVFLAG        # ZERO UPDATE STATE VECTOR REQUEST FLAGWRD
                EXTEND
                WRITE   CHAN5           # TURN OFF RCS JETS.
                EXTEND
                WRITE   CHAN6           # TURN OFF RCS JETS.
                EXTEND
                WRITE   CHAN12
                EXTEND
                WRITE   CHAN13
                EXTEND
                WRITE   CHAN14
                CS      DSPTAB +11D
                MASK    BITS4&6
                CCS     A
                TC      +4
                CA      BITS4&6
                EXTEND                  # THE IMU WAS IN COARSE ALIGN IN GIMBAL
                WOR     CHAN12          # LOCK, SO PUT IT BACK INTO COARSE ALIGN.
 +4             TC      MR.KLEAN

                CS      ZERO
                TS      MODREG

                CAF     IM30INIF        # FRESH START IMU INITIALIZATION
                TS      IMODES30

                CAF     MAXDB
                TS      DB
                CAF     FOUR
                TS      RATEINDX        # INITIALZE KALCMANU RATE
                CA      BOOLSTRT
                TS      DAPBOOLS
                CAF     EBANK6
                TS      EBANK
                EBANK=  HIASCENT

                CA      STIKSTRT
                TS      STIKSENS
                CA      RATESTRT
                TS      -RATEDB
                CAF     FULLAPS         # INITIALIZE MAXIMUM ASCENT MASS FOR USE
                TS      HIASCENT        #   BY 1/ACCS UNTIL THE PAD LOAD IS DONE.
                CA      77001OCT        #     LOAD DAP FILTER GAINS PAD LOAD.
```

```
                TS      DKTRAP          #       TO BEST PRESENT ESTIMATE OF GOODIES
                TS      LMTRAP          # .14 DEG
                CA      60DEC
                TS      DKKAOSN
                TS      LMKAOSN         # 6 SEC GAIN FOR ALPHA
                CA      ZERO
                TS      LMOMEGAN        # UNITY GAIN
                CA      TEN
                TS      DKOMEGAN        # 1 SEC GAIN FOR OMEGA
                CAF     BIT8            # SET DOCKED DB TO 1.4 DEG.  MAY OVERWRITE
                TS      DKDB            #       WITH PAD LOAD.
                CAF     IM33INIT
                AD      BIT6            # KEEP BOTH DAP AND ERROR-NEEDLES DISPLAY
                TS      IMODES33        #       OFF UNTIL ICDU ZERO IS FINISHED.

                EXTEND                  # INITIALIZE SWITCHES ONLY ON FRESH START.
                DCA     SWINIT
                DXCH    STATE
                CA      SWINIT +2
                TS      STATE +2
                CA      REFSMBIT        # DO NOT ALTER REFSMFLG ON FRESH START.
                MASK    STATE +3
                AD      SWINIT +3
                TS      STATE +3
                EXTEND
                DCA     SWINIT +4
                DXCH    STATE +4
                EXTEND
                DCA     SWINIT +6
                DXCH    STATE +6
                CA      SURFFBIT        # DO NOT ALTER  SURFFLAG ON FRESH START.
                AD      CMOONBIT        #               CMOONFLG
                AD      LMOONBIT        #               LMOONFLG
                MASK    STATE +8D
                AD      SWINIT +8D
                TS      STATE +8D
                CA      SWINIT +9D
                TS      STATE +9D
                CA      APSFLBIT        # DO NOT ALTER APSFLAG ON FRESH START.
                MASK    STATE +10D
                AD      SWINIT +10D
                TS      STATE +10D
                CAF     SWINIT +11D
                TS      STATE +11D

ENDRSTRT        TC      POSTJUMP        # NOW IN ANOTHER BANK.
                CADR    DUMMYJOB +2     # PICKS UP AT RELINT.   (DON'T ZERO NEWJOB)

MR.KLEAN        INHINT
```

```
                EXTEND
                DCA     NEG0
                DXCH    -PHASE2
POOKLEAN        EXTEND
                DCA     NEG0
                DXCH    -PHASE4
V37KLEAN        EXTEND
                DCA     NEG0
                DXCH    -PHASE1
                EXTEND
                DCA     NEG0
                DXCH    -PHASE3
                EXTEND
                DCA     NEG0
                DXCH    -PHASE5
                EXTEND
                DCA     NEG0
                DXCH    -PHASE6
                TC      Q
```

```
# COMES HERE FROM LOCATION 4000, GOJAM, RESTART ANY PROGRAMS WHICH MAY HAVE BEEN RUNNING AT THE TIME.

                EBANK=  LST1
GOPROG          INCR    REDOCTR         # ADVANCE RESTART COUNTER.

                LXCH    Q
                EXTEND
                ROR     SUPERBNK
                DXCH    RSBBQ
                CA      DSPTAB +11D
                MASK    BIT4
                EXTEND
                BZF     +4
                AD      BIT6            # SET ERROR COUNTER ENABLE
                EXTEND
                WOR     CHAN12          # ISS WAS IN COARSE ALIGN SO GO BACK TO
BUTTONS         TC      LIGHTSET

# ERASCHK TEMPORARILY STORES THE CONTENST OF TWO ERASABLE LOCATIONS, X
# AND X+1 INTO SKEEP5 AND SKEEP6.  IT ALSO STORES X INTO SKEEP7 AND
# ERESTORE.  IF ERASCHK IS INTERRUPTED BY A RESTART, C(ERESTORE) SHOULD
# EQUAL C(SKEEP7), AND SHOULD BE A + NUMBER LESS THAN 2000 OCT.  OTHERWISE
# C(ERESTORE) SHOULD EQUAL +0.

                CAF     HI5
                MASK    ERESTORE
                EXTEND
                BZF     +2              # IF ERESTORE NOT = +0 OR +N LESS THAN 2K,
                TCF     NONAVKEY +3     # DO FRESH START -- E MEMORY MIGHT BE BAD
                CS      ERESTORE
                EXTEND
                BZF     DORSTART        # = +0 CONTINUE WITH RESTART.
                AD      SKEEP7
                EXTEND
                BZF     +2              # = SKEEP7, RESTORE E MEMORY.
                TCF     NONAVKEY +3     # DO FRESH START -- E MEMORY MIGHT BE BAD
                CA      SKEEP4
                TS      EBANK           # EBANK OF E MEMORY THAT WAS UNDER TEST.
                EXTEND                  # (NOT DXCH SINCE THIS MIGHT HAPPEN AGAIN)
                DCA     SKEEP5
                INDEX   SKEEP7
                DXCH    0000            # E MEMORY RESTORED
                CA      ZERO
                TS      ERESTORE
DORSTART        TC      STARTSUB        # DO INITIALIZATION AFTER ERASE RESTORE.

SETINFL         CS      INTFLBIT
                MASK    FLGWRD10
                TS      FLGWRD10
```

```
                CA      9,6,4           # LEAVE PROG ALARM, GIMBAL LOCK, NO ATT
                MASK    DSPTAB +11D     # LAMPS INTACT ON HARDWARE RESTART
                AD      BIT15
                XCH     DSPTAB +11D
                CAF     IFAILINH        # LEAVE IMU FAILURE INHIBITS INTACT ON
                MASK    IMODES30        # HARDWARE RESTART, RESET ALL FAILURE
                AD      IM30INIR        # CODES.
                TS      IMODES30

                CA      AGSWORD         # BE SURE OF CORRECT DOWNLIST
                TS      DNLSTCOD

                CA      BIT4            # TURN ON THROTTLE COUNTER
                EXTEND
                WOR     CHAN14          # TURN ON THRUST DRIVE
                CS      FLAGWRD5
                MASK    ENGONBIT
                CCS     A
                TCF     +5
                CAF     BIT13
                EXTEND
                WOR     DSALMOUT        # TURN ENGINE ON
                TCF     GOPROG3
 +5             CAF     BIT14
                EXTEND
                WOR     DSALMOUT        # TURN ENGINE OFF
                TCF     GOPROG3

ENEMA           INHINT
                TC      STARTSB1
                TCF     GOPROG2A
GOPROG2         TC      STARTSB2
GOPROG2A        TC      LIGHTSET
                CS      RSFLGBTS        # CLEAR BITS 7 AND 14.
                MASK    FLGWRD10
                TS      FLGWRD10

GOPROG3         CAF     NUMGRPS         # VERIFY PHASE TABLE AGREEMENTS
PCLOOP          TS      MPAC +5
                DOUBLE
                EXTEND
                INDEX   A
                DCA     -PHASE1         # COMPLEMENT INTO A, DIRECT INTO L.
                EXTEND
                RXOR    LCHAN           # RESULT MUST BE -0 FOR AGREEMENT.
                CCS     A
                TCF     PTBAD           # RESTART FAILURE.
                TCF     PTBAD
                TCF     PTBAD
```

```
                CCS     MPAC +5         # PROCESS ALL RESTART GROUPS.
                TCF     PCLOOP

                TS      MPAC +6         # SET TO +0.
                TC      MMDSPLAY        # DISPLAY MAJOR MODE

                INHINT                  # RELINT DONE IN MMDSPLAY

                CS      DIDFLBIT        # CLEAR DIDFLAG IN ORDER TO FORCE R10 TO
                MASK    FLAGWRD1        # RE-INITIALIZE ITSELF IF IT HAD BEEN
                TS      FLAGWRD1        # OPERATION AT THE TIME OF THE RESTART.

                CS      RODFLBIT        # CLEAR RODFLAG.  IF P66 IS IN OPERATION
                MASK    FLAGWRD1        #        IT WILL RE-INITIALIZE ITSELF AND
                TS      FLAGWRD1        #        CONTINUE.

                CS      P21FLBIT        # CLEAR P21 FLAG SO THAT P21 WILL COMPUTE
                MASK    FLAGWRD0        # NEW BASE STATE VECTORS.
                TS      FLAGWRD0

                CAF     NUMGRPS         # SEE IF ANY GROUPS RUNNING.
NXTRST          TS      MPAC +5
                DOUBLE
                INDEX   A
                CCS     PHASE1
                TCF     PACTIVE         # PNZ -- GROUP ACTIVE.
                TCF     PINACT          # +0 -- GROUP NOT RUNNING.

PACTIVE         TS      MPAC
                INCR    MPAC            # ABS OF PHASE.
                INCR    MPAC +6         # INDICATE GROUP DEMANDS PRESENT.
                CA      RACTCADR
                TC      SWCALL          # MUST RETURN TO SWRETURN.

PINACT          CCS     MPAC +5         # PROCESS ALL RESTART GROUPS.
                TCF     NXTRST

                CCS     MPAC +6         # NO, CHECK PHASE ACTIVITY FLAG
                TCF     ENDRSTRT        # PHASE ACTIVE
                CAF     BIT15           # IS MODE -0
                MASK    MODREG
                EXTEND
                BZF     GOTOPOOH        # NO
                TCF     ENDRSTRT        # YES
PTBAD           TC      ALARM           # SET ALARM TO SHOW PHASE TABLE FAILURE.
                OCT     1107

                TCF     DOFSTRT1
#********* ****** ******
```

```
# DO NOT USE GOPROG2 OR ENEMA WITHOUT CONSULTING POOH PEOPLE.

OCT10000        =       BIT13
OCT30000        =       PRIO30
OCT7777         OCT     7777
STIKSTRT        DEC     0.825268        # 20 D/S MAXIMUM COMPANDED RATE
RATESTRT        DEC     -218
RACTCADR        CADR    RESTARTS
BOOLSTRT        OCT     21312
77001OCT        OCT     77001           # .14 DEG SCALED AT 4.5 DEG
60DEC           DEC     60
RSFLGBTS        OCT     20100
MAXDB           OCTAL   03434           # 5 DEG ATTITUDE DEADBAND, SCALED AT 45.

LIGHTSET        CAF     BIT5            # CHECK FOR MARK REJECT AND ERROR RESET
                EXTEND
                RAND    NAVKEYIN
                EXTEND
                BZF     NONAVKEY        # NO MARK REJECT
                EXTEND
                READ    MNKEYIN         # CHECK IF KEYS 2M AND 5M ON
                AD      -ELR            # MAIN DSKY KEYCODE (BITS 1-5)
                EXTEND
                BZF     +2

NONAVKEY        TC      Q

                TC      STARTSUB
                TCF     DOFSTART
 +3             TC      STARTSUB
                TCF     DOFSTRT1        # DO FRESH START BUT DON'T TOUCH ENGINE
```

```
# INITIALIZATION COMMON TO BOTH FRESH START AND RESTART.

                EBANK=  AOSQ
STARTSUB        CAF     LDNPHAS1        # SET POINTER SO NEXT 20MS DOWNRUPT WILL
                TS      DNTMGOTO        # CAUSE THE CURRENT DOWNLIST TO BE
                                        # INTERRUPTED AND START SENDING FROM THE
                                        # BEGINNING OF THE CURRENT DOWNLIST.
                CAF     BIT6
                EXTEND
                RAND    CHAN33
                AD      RMODINIT
                TS      RADMODES

STARTSB1        CAF     POSMAX
                TS      TIME3
                AD      MINUS2
                TS      TIME4
                AD      NEGONE
                TS      TIME5

                CAF     EBANK6
                TS      EBANK

                CS      BIT13           # CAUSE DAPIDLER TO CALL 1/ACCS
                MASK    RCSFLAGS
                TS      RCSFLAGS        # ZERO BIT 13
                CAF     POSMAX          # DISABLE TIME6 CLOCK.  JUST IN CASE A T6
                TS      T6NEXT          #       RUPT IS ALREADY IN THE PRIORITY CHAIN,
                EXTEND                  #       ENSURE THAT ITS INPUTS WILL RENDER IT
                WAND    CHAN13          #       INEFFECTUAL.
                CAF     ZERO
                TS      NXT6ADR
                TS      NEXTP

                CS      ACCSOKAY
                MASK    DAPBOOLS
                TS      DAPBOOLS

                EXTEND                  # SET T5RUPT FOR DAPIDLER PROGRAM.
                DCA     IDLEADR
                DXCH    T5ADR

STARTSB2        CAF     OCT30001        # DURING SOFTWARE RESTART, DO NOT DISTURB
                EXTEND                  # ENGINE ON, OFF AND ISS WARNING.
                WAND    DSALMOUT

                CS      READRBIT        # CLEAR READRFLG FOR R29
                MASK    FLAGWRD3
                TS      FLAGWRD3
```

```
                CS      FLAGWRD3        # DURING SOFTWARE RESTART, CLEAR TURNON,
                MASK    NR29FBIT        # REPOSITION, CDU ZERO AND REMODE BITS
                EXTEND                  # IN RADMODES, SINCE TASKS ASSOCIATED
                BZF     +2              # WITH THESE BITS HAVE BEEN KILLED
                CAF     BIT10           # ALSO IF R29 HAD BEEN REQUESTED.
                AD      OCT32001        # (NOR29FLG = 0) CLEAR BIT 10 RADMODES
                COM                     # TO MAKE R29 FORGET IT HAD STARTED
                MASK    RADMODES        # DESIGNATING
                TS      RADMODES
                CAF     OCT27470        # DURING SOFTWARE RESTART, DO NOT DISTURB
                EXTEND                  # IMU FLAGS.  (COARSE ALIGN ENABLE, ZERO
                WAND    CHAN12          # IMU CDUS, ENABLE IMU COUNTER) AND GIMBAL
                                        # TRIM DRIVES.  LEAVE RR LOCKON ENABLE
                                        # ALONE.

                CS      NORRMBIT        # ENABLE R25.
                MASK    FLAGWRD5
                TS      FLAGWRD5

                CS      R77FLBIT        # CLEAR R77FLAG
                MASK    FLAGWRD5
                TS      FLAGWRD5
                CAF     OCT74160        # DURING SOFTWARE RESTART, DO NOT DISTURB
                EXTEND                  # TELEMETRY FLAGS, RESET TRAP FLAGS, AND
                WAND    CHAN13          # ENABLE T6RUPT FLAG.

                CAF     BIT12           # REENABLE RUPT10 (RUPT QUICKLY
                EXTEND                  #       RESUMES EXCEPT DURING P64)
                WOR     CHAN13

                CAF     BIT6            # DURING SOFTWARE RESTART, DO NOT DISTURB
                EXTEND                  # GYRO ENABLE FLAG.
                WAND    CHAN14

                EBANK=  LST1
                CAF     STARTEB
                TS      EBANK           # SET FOR E3

                CAF     NEG1/2          # INITIALIZE WAITLIST DELTA-TS.
                TS      LST1 +7
                TS      LST1 +6
                TS      LST1 +5
                TS      LST1 +4
                TS      LST1 +3
                TS      LST1 +2
                TS      LST1 +1
                TS      LST1

                CS      ENDTASK
                TS      LST2
```

```
                TS      LST2 +2
                TS      LST2 +4
                TS      LST2 +6
                TS      LST2 +8D
                TS      LST2 +10D
                TS      LST2 +12D
                TS      LST2 +14D
                TS      LST2 +16D
                CS      ENDTASK +1
                TS      LST2 +1
                TS      LST2 +3
                TS      LST2 +5
                TS      LST2 +7
                TS      LST2 +9D
                TS      LST2 +11D
                TS      LST2 +13D
                TS      LST2 +15D
                TS      LST2 +17D

                CS      ZERO            # MAKE ALL EXECUTIVE REGISTER SETS
                TS      PRIORITY        # AVAILABLE.
                TS      PRIORITY +12D
                TS      PRIORITY +24D
                TS      PRIORITY +36D
                TS      PRIORITY +48D
                TS      PRIORITY +60D
                TS      PRIORITY +72D
                TS      PRIORITY +84D

                TS      DSRUPTSW
                TS      NEWJOB          # SHOWS NO ACTIVE JOBS.

                CAF     VAC1ADRC        # MAKE ALL VAC AREAS AVAILABLE.
                TS      VAC1USE
                AD      LTHVACA
                TS      VAC2USE
                AD      LTHVACA
                TS      VAC3USE
                AD      LTHVACA
                TS      VAC4USE
                AD      LTHVACA
                TS      VAC5USE

                CAF     TEN
DSPOFF          TS      MPAC            # R1, R2, R3
                CS      BIT12
                INDEX   MPAC
                TS      DSPTAB
                CCS     MPAC
                TCF     DSPOFF
```

```
                TS      DELAYLOC
                TS      DELAYLOC +1
                TS      DELAYLOC +2
                TS      R1SAVE
                TS      INLINK
                TS      DSPCNT
                TS      CADRSTOR
                TS      REQRET
                TS      CLPASS
                TS      DSPLOCK
                TS      MONSAVE         # KILL MONITOR
                TS      MONSAVE1
                TS      VERBREG
                TS      NOUNREG
                TS      DSPLIST
                TS      MARKSTAT
                TS      EXTVBACT        # MAKE EXTENDED VERBS AVAILABLE
                TS      IMUCADR
                TS      OPTCADR
                TS      RADCADR
                TS      ATTCADR
                TS      LGYRO
                TS      FLAGWRD4        # KILL INTERFACE DISPLAYS
                CAF     NOUTCON
                TS      NOUT

                CS      ONE
                TS      SAMPLIM
                CAF     BIT6
                MASK    IMODES33        # LEAVE BIT 6 UNCHANGED
                AD      IM33INIT        # NO PIP OR TM FAILS.  BIT6=0 IN THIS WORD.
                TS      IMODES33

                CAF     LESCHK          # SELF CHECK GO-TO REGISTER.
                TS      SELFRET

                CS      VD1
                TS      DSPCOUNT

                TC      Q
                EBANK=  AOSQ
IDLEADR         2CADR   DAPIDLER

IFAILINH        OCT     435
LDNPHAS1        GENADR  DNPHASE1
LESCHK          GENADR  SELFCHK
VAC1ADRC        ADRES   VAC1USE
OCT32001        OCT     32001
LTHVACA         DEC     44
```

```
OCT27470        OCT     27470
OCT74160        OCT     74160
OCT30001        OCT     30001
STARTEB         EQUALS  EBANK3
NUMGRPS         EQUALS  FIVE
-ELR            OCT     -22             # -ERROR LIGHT RESET KEY CODE.
IM30INIF        OCT     37411           # INHIBITS IMU FAIL FOR 5 SEC AND PIP ISSW
IM30INIR        OCT     37000
IM33INIT        =       PRIO16          # NO PIP OR TM FAIL SIGNALS.
9,6,4           OCT     450
RMODINIT        OCT     00102
SWINIT          OCT     0
                OCT     0
                OCT     0
                OCT     02000           # BIT 11 = NOR29FLG
                OCT     0
                OCT     0
                OCT     0
                OCT     00100

                OCT     0
                OCT     0
                OCT     0
                OCT     40000           # BIT 15 = LRBYPASS.
```

```
# PROGRAM NAME          GOTOPOOH                ASSEMBLY SUNDANCE
# LOG SECTION           FRESH START AND RESTART
#
# FUNCTIONAL DESCRIPTION
#
#       FLASH V 37 ON DSKY MM CHANGE REQUEST
#
# INPUT/OUTPUT INFORMATION
#
#       A. CALLING SEQUENCE                     TC GOTOPOOH
#       B. ERASABLE INITIALIZATION              NONE
#       C. OUTPUT FLASH V 37 ON DSKY
#       D. DEBRIS                               L
#
# PROGRAM ANALYSIS
#
#       A. SUBROUTINES CALLED                   PRIODSPR, LINUS
#       B. NORMAL EXIT                          TCF ENDOFJOB
#       C. ALARM AND ABORT EXITS                NONE

                BLOCK   03
                SETLOC  FFTAG5
                BANK

                COUNT*  $$/POO
GOTOPOOH        CAF     OCT33                   # 4.33 SPOT FOR GOPOOFIX
                TS      L
                COM
                DXCH    -PHASE4

                TC      POSTJUMP
                CADR    GOPOOFIX
OCT24           MM      20
OCT31           MM      25

                BANK    20
                SETLOC  VERB37
                BANK

                COUNT*  $$/POO                  # VERB 37 AND POO IN BANK 4.
GOPOOFIX        TC      DOWNFLAG                # ALLOW X-AXIS OVERRIDE
                ADRES   XOVINFLG

                TC      DOWNFLAG                # INSURE THAT ULLAGE IS OFF
                ADRES   ULLAGFLG
```

```
                TC      CLEARMRK +2             # RELEASE MARK DISPLAY SYSTEM.
                CAF     V37N99
                TC      BANKCALL
                CADR    GOFLASH
                TCF     -3
                TCF     -4
                TCF     -5

V37N99          VN      3799
```

```
# PROGRAM NAME          V37                     ASSEMBLY SUNDANCE
#
# LOG SECTION           FRESH START AND RESTART
#
# FUNCTIONAL DESCRIPTION
#
#       1. CHECK IF NEW PROGRAM ALLOWED.  IF BIT 1 OF FLAGWRD2 (NODOFLAG) IS SET, AN ALARM 1520 IS CALLED.
#       2. CHECK FOR VALIDITY OF PROGRAM SELECTED.  IF AN INVALID PROGRAM IS SELECTED, THE OPERATOR ERROR LIGHT IS
#          SET AND CURRENT ACTIVITY, IF ANY, CONTINUE.
#       3. SERVICER IS TERMINATED IF IT HAS BEEN RUNNING.
#       4. INSTALL IS EXECUTED TO AVOID INTERRUPTING INTEGRATION.
#       5. THE ENGINE IS TURNED OFF AND THE DAP IS INITIALIZED FOR COAST.
#       6. TRACK AND UPDATE FLAGS ARE SET TO ZERO.
#       7. DISPLAY SYSTEM IS RELEASED.
#       8. THE FOLLOWING ARE PERFORMED FOR EACH OF THE THREE CASES.
#               A. PROGRAM SELECTED IS P00
#                       1. RENDEZVOUS AND P25 FLAGS ARE RESET.  (KILL P20 AND P25)
#                       2. STATINT1 IS SCHEDULED BY SETTING RESTART GROUP 2.
#                       3. MAJOR MODE 00 IS STORED IN THE MODE REGISTER (MODREG).
#                       4. SUPERBANK 3 IS SELECTED.
#                       5. NODOFLAG IS RESET.
#                       6. ALL RESTART GROUPS EXCEPT GROUP2 ARE CLEARED. CONTROL IS TRANSFERRED TO RESTART PROGRAM (GOPROG2)
#                          WHICH CAUSES ALL CURRENT ACTIVITY TO BE DISCONTINUED AND A 9 MINUTE INTEGRATION CYCLE TO BE
#                          INITIATED.
#               B. PROGRAM SELECTES IS P20 OR P25.
#                       1. IF THE CURRENT MAJOR MODE IS THE SAME AS THE SELECTED NEWPROGRAM.  THE PROGRAM IS RE-INITIALIZED
#                          VIA V37XEQ, ALL RESTART GROUPS, EXCEPT GROUP 4 ARE CLEARED.
#                       2. IF THE CURRENT MAJOR MODE IS NOT EQUAL TO THE NEW REQUEST, A CHECK IS MADE TO SEE IF THE REQUEST-
#                          ED MAJOR MODE HAS BEEN RUNNING THE BACKGROUND,
#                          AND IF IT HAS, NO NEW PROGRAM IS SCHEDULED, THE EXISTING
#                          P20 OR P25 IS RESTARTED TO CONTINUE, AND ITS MM IS SET.
#                       3. CONTROL IS TRANSFERRED TO GOPROG2.
#               C. PROGRAM SELECTED IS NEITHER P00, P20, NOR P25
#                       1. V37XEQ IS SCHEDULED (AS A JOB) BY SETTING RESTART GROUP 4
#                       2. ALL CURRENT ACTIVITY EXCEPT RENDEZVOUS AND TRACKING IS DISCONTINUED BY CLEARING ALL RESTART
#                          GROUPS.  IF THE RENDEZVOUS OR THE P25 FLAG IS ON, GROUP 2 IS NOT CLEARED, ALLOWING THESE PROGRAMS
#                          TO CONTINUE.
#
# INPUT/OUTPUT INFORMATION
#
#       A. CALLING SEQUENCE
#               CONTROL IS DIRECTED TO V37 BY THE VERBFAN ROUTINE.
#               VERBFAN GOES TO C(VERBTAB+C(VERBREG)). VERB 37 = MMCHANG.
#               MMCHANG EXECUTES A `TC POSTJUMP', CADR V37.
#
#       B. ERASABLE INITIALIZATION              NONE
#
#       C. OUTPUT
```

```
#                 MAJOR MOD CHANGE
#
#        D. DEBRIS
#                 MMNUMBER, MPAC +1, MINDEX, BASETEMP +C(MINDEX), FLAGWRD0, FLAGWRD1, FLAGWRD2, MODREG, GOLOC -1,
#                 GOLOC, GOLOC +1, GOLOC +2, BASETEMP, -PHASE2, PHASE2, -PHASE4
#
# PROGRAM ANALYSIS
#
#        A. SUBROUTINES CALLED
#                 ALARM, RELDSP, PINBRNCH, INTSTALL, ENGINOF2, ALLCOAST, V37KLEAN, GOPROG2, FALTON, FINDVAC, SUPERSW,
#                 DSPMM
#
#        B. NORMAL EXIT                           TC ENDOFJOB
#
#        C. ALARMS                                1520 (MAJOR MODE CHANGE NOT PERMITTED)

V37               TS       MMNUMBER               # SAVE MAJOR MODE
                  CAF      PRIO30                 # RESTART AT PINBALL PRIORITY
                  TS       RESTREG

                  CA       IMODES30               # IS IMU BEING INITIALIZED
                  MASK     BIT6
                  CCS      A
                  TCF      CANTROD

                  CS       MMNUMBER               # IS P70 REQUESTED?
                  AD       DEC70
                  EXTEND
                  BZF      SETUP70                # YES
                  AD       ONE                    # IS P71 REQUESTED?
                  EXTEND
                  BZF      SETUP71                # YES

                  CA       MMNUMBER               # IS NEW REQUEST P00
                  EXTEND
                  BZF      ISSERVON               # YES, CHECK SERVICER STATUS

                  CS       FLAGWRD2               # NO, IS NODO V37 FLAG SET
                  MASK     NODOBIT
                  CCS      A
                  TCF      CHECKTAB               # NO
CANTROD           TC       ALARM
                  OCT      1520

V37BAD            TC       RELDSP                 # RELEASES DISPLAY FROM ASTRONAUT

                  TC       POSTJUMP               # BRING BACK LAST NORMAL DISPLAY IF THERE
                  CADR     PINBRNCH               # WAS ONE.  OTHERWISE DO AN EOJ.

CHECKTAB          CA       NOV37MM                # INDEX FOR MM TABLES.
```

```
AGAINMM         TS      MPAC +1
                NDX     MPAC +1
                CA      PREMM1                  # OBTAIN WHICH MM THIS IS FOR
                MASK    LOW7
                COM
                AD      MMNUMBER
                CCS     A
                CCS     MPAC +1                 # IF GR, SEE IF ANY MORE IN LIST
                TCF     AGAINMM                 # YES, GET NEXT ONE
                TCF     V37NONO                 # LAST TIME OR PASSED MM

                CA      MPAC +1
                TS      MINDEX                  # SAVE INDEX FOR LATER

ISSERVON        CS      FLAGWRD7                # V37 FLAG SET -- I.E., IS SERVICER GOING
                MASK    V37FLBIT
                CCS     A
                TCF     CANV37                  # NO

                TC      DOWNFLAG                # YES, TURN OFF THE AVERAGE FLAG AND
                ADRES   AVEGFLAG                # WAIT FOR SERVICER TO RETURN TO CANV37

                CAF     V37RETAD
                TS      OUTROUTE

                TCF     ENDOFJOB

V37RET          CS      FLAGWRD0                # IS P20 OR P22 RUNNING?
                MASK    RNDVZBIT
                CCS     A
                TCF     +2                      # NO. CHECK FOR P25.
                TCF     2.7SPT                  # YES. DO 2.7SPOT
                CS      FLAGWRD0                # IS P25 RUNNING?
                MASK    P25FLBIT
                CCS     A
2.0SPT          CA      OCT37667
2.11SPT         AD      BIT5
2.7SPT          AD      OCT40072
                TC      PHSCHNGA

CANV37          CAF     ZERO
                EXTEND
                WRITE   SUPERBNK

                CAF     ROOAD
                TS      TEMPFLSH

                TC      PHASCHNG
                OCT     14
```

```
ROO             TC      INTPRET

                CALL                            # WAIT FOR INTEGRATION TO FINISH
                        INTSTALL
DUMMYAD         EXIT

                TC      DOWNFLAG
                ADRES   3AXISFLG                # RESET 3-AXIT FLAG

                CAF     LRBYBIT                 # CLEAN UP THE R12 FLAGWORD.
                TS      FLGWRD11

                TC      DOWNFLAG                # INSURE THAT THE R04FLAG IS CLEAR.
                ADRES   R04FLAG

                TC      DOWNFLAG                # INSURE MUNFLAG IS CLEAR.
                ADRES   MUNFLAG

                TC      DOWNFLAG                # ALLOW X-AXIS OVERRIDE.
                ADRES   XOVINFLG
                CCS     MMNUMBER                # IS THIS A POOH REQUEST
                TCF     NOUVEAU                 # NO, PICK UP NEW PROGRAM

POOH            TC      RELDSP                  # RELEASE DISPLAY SYSTEM

                CAF     PRIO5                   # SET VARIABLE RESTART PRIORITY FOR
                TS      PHSPRDT2                # P00 INTEGRATION.

                TC      CLRADMOD                # CLRADMOD DOES AN INHINT.

                CS      NODOBIT                 # TURN OFF NODOFLAG.
                MASK    FLAGWRD2
                TS      FLAGWRD2

                CA      FIVE                    # SET RESTART FOR STATEINT1
                TS      L
                COM
                DXCH    -PHASE2

                CS      OCT700                  # TURN OFF P20, P25, IMU IN USE FLAG
                MASK    FLAGWRD0
                TS      FLAGWRD0                #                         REMDFLG

                CAF     DNLADP00

SEUDOP00        TS      DNLSTCOD                # SET UP APPROPRIATE DOWNLIST CODE
                TS      AGSWORD                 #   (CURRENT LIST WILL BE COMPLETED BEFORE
                                                #     NEW ONE IS STARTED)
                TC      IBNKCALL
                CADR    ENGINOF1
```

```
                TC      IBNKCALL                # INSURE ALLCOAST.
                CADR    ALLCOAST                # DOES A RESTORDB.

                CS      OCT120                  # TURN OFF TRACK, UPDATE FLAGS
                TS      EBANKTEM
                MASK    FLAGWRD1
                TS      FLAGWRD1

                TC      IBNKCALL                # KILL GROUPS 1,3,5,6
                CADR    V37KLEAN

                CCS     MMNUMBER                # IS IT POOH
                TCF     RENDVOO                 # NO
GOMOD           TC      IBNKCALL                # REDUNDANT EXCEPT FOR GROUP 4
                CADR    POOKLEAN

                CA      MMNUMBER
                TS      MODREG

GOGOPROG        TC      POSTJUMP
                CADR    GOPROG2

RENDVOO         CS      MODREG                  # IS CURRENT PROGRAM 22
                AD      OCT26
                EXTEND
                BZF     RESET22                 # YES -- CLEAR RENDEZVOUS FLAG

                CS      MMNUMBER                # IS NE PROGRAM P22
                AD      OCT26
                EXTEND
                BZF     RESET22

                AD      NEG2                    # IS NEW PROGRAM = P20 OR P25
                EXTEND
                BZF     RENDNOO                 # YES
                AD      FIVE                    # 25
                EXTEND
                BZF     RENDNOO                 # YES

                CA      OCT500                  # NO, IS EITHER P20 OR P25 RUNNING
                MASK    FLAGWRDO
                CCS     A
                TCF     POOFIZZ                 # YES, LEAVE GROUP 2 TO PICK UP P20 OR P25

RESET22         CS      OCT700                  # CLEAR RENDEZVOUS, P25
                MASK    FLAGWRDO                # AND IMU IN USE FLAGS
                TS      FLAGWRDO
                TC      CLRADMOD
```

```
KILL2           EXTEND                          # NO, KILL 2
                DCA     NEGO
                DXCH    -PHASE2

POOFIZZ         CAF     V37QCAD                 # RESTART POINT FOR V37XEQ
                TS      TEMPFLSH

                TCF     GOGOPROG

RENDNOO         CS      MODREG
                AD      OCT24
                EXTEND
                BZF     KILL2                   # P20 OR P25 ON TOP OF P20 OR P25 --

                AD      FIVE
                EXTEND
                BZF     KILL2

                CA      OCT500
                MASK    FLAGWRD0
                AD      MMNUMBER
                COM
                AD      P20REG                  # IS IT 20 AND IS RENDEZVOUS FLAG ON
                EXTEND
                BZF     STATQUO                 # YES
                AD      OCT305                  # IS IT 25 AND IS P25 BIT ON
                EXTEND
                BZF     STATQUO                 # YES, LEAVE AS IS
                TCF     KILL2

STATQUO         CS      FLAGWRD1                # SET TRACKFLAG
                MASK    OCT120                  #       UPDATE FLAG
                ADS     FLAGWRD1

                TCF     GOMOD

NOUVEAU         CAF     OCT500                  # IS P20 OR P25 FLAG SET
                MASK    FLAGWRD0
                CCS     A
                TCF     +3                      # YES
                TC      DOWNFLAG                # NO, RESET IMUINUSE FLAG
                ADRES   IMUSE

                INDEX   MINDEX
                CAF     DNLADMM1                # OBTAIN APPROPIRATE DOWNLIST ADDRESS

                INHINT
                TCF     SEUDOPOO

V37NONO         TC      FALTON                  # COME HERE IF MM REQUESTED DOESN'T EXIST
```

```
                TCF     V37BAD

OCT00010        EQUALS  BIT4
OCT500          OCT     500                     # BITS 7 AND 9
OCT305          OCT     305
OCT26           OCT     26
P20REG          OCT     124

V37XEQ          INHINT
                INDEX   MINDEX
                CAF     PREMM1                  # OBTAIN PRIO, EBANK, AND MM
                TS      MMTEMP
                TS      CYR                     # SHIFT RIGHT TO BITS 14-10

                CA      CYR
                MASK    PRIO37
                TS      PHSPRDT4                # PRESET GROUP 4 RESTART PRIORITY
                TS      NEWPRIO                 # STORE PRIO FOR SPVAC

                CA      MMTEMP                  # OBTAIN EBANK -- BITS 8, 9, 10 OF MMTEMP.
                EXTEND
                MP      BIT8
                MASK    LOW3
                TS      L

                INDEX   MINDEX
                CAF     FCADRMM1
                TS      BASETEMP                # MAKE BBCON BY ADDING HI5 OR FCADR
                MASK    HI5
                ADS     L

                CA      BASETEMP                # OBTAIN GENADR PORTION OF 2CADR.
                MASK    LOW10
                AD      BIT11

                TC      SPVAC

V37XEQC         CA      MMTEMP                  # UPON RETURN FROM FINDVAC PLACE THE
                MASK    LOW7                    # NEW MM IN MODREG (THE LOW 7 BITS OF
                TC      NEWMODEA                # PHSPRDT1)

                TC      RELDSP                  # RELEASE DISPLAY
                TC      ENDOFJOB                # AND EXIT

NEG7            EQUALS  OCT77770

MMTEMP          EQUALS  PHSPRDT3
BASETEMP        EQUALS  TBASE4
V37QCAD         CADR    V37XEQ +3
ROOAD           CADR    DUMMYAD
```

```
V37RETAD        CADR    V37RET
OCT37667        OCT     37667
OCT40072        OCT     40072
OCT700          OCT     700

SETUP71         CAF     THREE
SETUP70         TS      Q
                EXTEND
                DCA     P70CADR
                AD      Q
                DTCB

DEC70           DEC     70
                EBANK=  R
P70CADR         2CADR   P70

# FOR VERB 37 TWO TABLES ARE MAINTAINED.  EACH TABLE HAS AN ETRY FOR EACH
# MAJOR MODE THAT CAN BE STARTED FROM THE KEYBOARD.  THE ENTRIES ARE PUT
# INTO THE TABLE WITH THE ENTRY FOR THE HIGHEST MAJOR MODE COMING FIRST,
# TO THE LOWEST MAJOR MODE WHICH IS THE LAST ENTRY IN EACH TABLE.
#
# THE FCADRMM TABLE CONTAINS THE FCADR OF THE STARTING JOB OF
# THE MAJOR MODE.  FOR EXAMPLE,
#
#       FCADRMM1        FCADR   P79             # START OF P 79
#                       FCADR   PROG18          # START OF P 18
#                       FCADR   P01             # START OF P 01
#
# NOTE:         THE FIRST ENTRY MUST BE LABELED FCADRMM1.
# -----

FCADRMM1        FCADR   P79
                FCADR   P78
                FCADR   P76
                FCADR   P75
                FCADR   P74
                FCADR   P73
                FCADR   P72
                FCADR   LANDJUNK
                FCADR   P63LM
                FCADR   P57
                FCADR   PROG52
                FCADR   P51
                FCADR   P47LM
                FCADR   P42LM
                FCADR   P41LM
                FCADR   P40LM
                FCADR   P39
                FCADR   P38
```

```
                FCADR   P35
                FCADR   P34
                FCADR   P33
                FCADR   P32
                FCADR   P31
                FCADR   P30
                FCADR   PROG25
                FCADR   PROG22
                FCADR   PROG21
                FCADR   PROG20
                FCADR   P12LM
                FCADR   P06

# THE PREMM TABLE CONTAINS THE E-BANK, MAJOR MODE, AND PRIORITY
# INFORMATION, IT IS IN THE FOLLOWING FORM,
#
#       PPP PPE EEM MMM MMM
#
#       WHERE THE       7 M BITS CONTAIN THE MAJOR MODE NUMBER
#                       3 E BITS CONTAIN THE E-BANK NUMBER
#                       5 P BITS CONTAIN THE PRIORITY AT WHICH THE JOB IS
#                           TO BE STARTED
#
#       FOR EXAMPLE,
#
#               PREMM1          OCT     67213           # PRIORITY      33
#                                                       # E-BANK        5
#                                                       # MAJOR MODE    11
#                               OCT     25437           # PRIORITY      12
#                                                       # E-BANK        6
#                                                       # MAJOR MODE    31
#
# NOTE:         THE FIRST ENTRY MUST BE LABELED PREMM1

PREMM1          OCT     27717           # MM 79         EBANK 7         PRIO 13
                OCT     27716           # MM 78         EBANK 7         PRIO 13
                OCT     27714           # MM 76         EBANK 7         PRIO 13
                OCT     27713           # MM 75         EBANK 7         PRIO 13
                OCT     27712           # MM 74         EBANK 7         PRIO 13
                OCT     27711           # MM 73         EBANK 7         PRIO 13
                OCT     27710           # MM 72         EBANK 7         PRIO 13
                OCT     27704           # MM 68         EBANK 7         PRIO 13
                OCT     27677           # MM 63         EBANK 7         PRIO 13
                OCT     27271           # MM 57         EBANK 5         PRIO 13
                OCT     27264           # MM 52         EBANK 5         PRIO 13
                OCT     27263           # MM 51         EBANK 5         PRIO 13
                OCT     27657           # MM 47         EBANK 7         PRIO 13
                OCT     27652           # MM 42         EBANK 7         PRIO 13
                OCT     27651           # MM 41         EBANK 7         PRIO 13
                OCT     27650           # MM 40         EBANK 7         PRIO 13
                OCT     27647           # MM 39         EBANK 7         PRIO 13
                OCT     27646           # MM 38         EBANK 7         PRIO 13
```

```
                OCT     27643           # MM 35         EBANK 7         PRIO 13
                OCT     27642           # MM 34         EBANK 7         PRIO 13
                OCT     27641           # MM 33         EBANK 7         PRIO 13
                OCT     27640           # MM 32         EBANK 7         PRIO 13
                OCT     27637           #
                OCT     27636           # MM 30         EBANK 7         PRIO 13
                OCT     27631           # MM 25         EBANK 7         PRIO 13
                OCT     27626           # MM 22         EBANK 7         PRIO 13
                OCT     27625           # MM 21         EBANK 7         PRIO 13
                OCT     27624           # MM 20         EBANK 7         PRIO 13
                OCT     27614           # MM 12         EBANK 7         PRIO 13
                OCT     27006           # MM 06         EBANK 4         PRIO 13

# NOTE:         THE FOLLOWING CONSTANT IS THE NUMBER OF ENTRIES IN EACH OF
# -----         THE ABOVE LISTS-1 (I.E., THE NUMBER OF MAJOR MODES (EXCEPT P00)
#               THAT CAN BE CALLED FROM THE KEYBOARD MINUS ONE)

NOV37MM         DEC     29              # MM'S -1

DNLADMM1        ADRES   RENDEZVU        # P79
                ADRES   RENDEZVU        # P78
                ADRES   RENDEZVU
                ADRES   RENDEZVU        # P75
                ADRES   RENDEZVU        # P74
                ADRES   RENDEZVU        # P73
                ADRES   RENDEZVU        # P72
                ADRES   DESASCNT        # P68
                ADRES   DESASCNT        # P63
                ADRES   LUNRSALN        # P57
                ADRES   COSTALIN        # P52
                ADRES   COSTALIN        # P51
                ADRES   ORBMANUV        # P47
                ADRES   ORBMANUV        # P42
                ADRES   ORBMANUV        # P41
                ADRES   ORBMANUV        # P40
                ADRES   RENDEZVU        # P39
                ADRES   RENDEZVU        # P38
                ADRES   RENDEZVU        # P35
                ADRES   RENDEZVU        # P34
                ADRES   RENDEZVU        # P33
                ADRES   RENDEZVU        # P32
                ADRES   RENDEZVU        # P31LM
                ADRES   RENDEZVU        # P30
                ADRES   RENDEZVU        # P25
                ADRES   LUNRSALN        # P22
                ADRES   RENDEZVU        # P21
                ADRES   RENDEZVU        # P20
                ADRES   DESASCNT        # P12
                ADRES   COSTALIN        # P06
DNLADP00        =       ZERO
COSTALIN        =       0
```

```
AGSUPDAT        =       1
RENDEZVU        =       2
ORBMANUV        =       3
DESASCNT        =       4
LUNRSALN        =       5

                BANK    13
                SETLOC  INTINIT
                BANK

                COUNT*  $$/INTIN

                EBANK=  RRECTCSM

# THIS ROUTINE DOES THE P00 INTEGRATION

STATEUP         SET     BOF             # EXTRAPOLATE CM STATE VECTOR
                        VINTFLAG
                        SURFFLAG        # ALSO 6X6 W-MATRIX IF LM ON LUNAR
                        DOINT           #       SURFACE AND W-MATRIX VALID
                BOF     SET             #       FOR RENDEZVOUS NAVIGATION.
                        RENDWFLG
                        DOINT
                        DIMOFLAG
DOINT           CLEAR   CALL
                        PRECIFLG        # ENGAGES 4-TIME STEP LOGIC IN INTEGRATION
                        INTEGRV         # WHEN MODREG = 0

                BON     DLOAD
                        SURFFLAG
                        NO-INT
                        TETCSM
                STCALL  TDEC1
                        INTSTALL
                CLEAR   CALL            # EXTRAPOLATE LM STATE VECTOR
                        VINTFLAG
                        SETIFLGS
                BOF                     # ALSO 9X9 W-MATRIX IF W IS VALID
                        RENDWFLG
                        DOINT2
                SET     SET
                        DIMOFLAG
                        D6OR9FLG
DOINT2          SET     CALL
                        PRECIFLG        # DISENGAGE 4 TIME STEP LOGIC IN INTEG.
                        INTEGRV
NO-INT          CLRGO
                        NODOFLAG
                        ENDINT
```

```
# THISVINT IS CALLED BY MIDTOAV1 AND 2

THISVINT        CLEAR   RVQ
                        VINTFLAG
```

```
# RESTART TABLES
# ------------------
#
# THERE ARE TWO FORMS OF RESTART TABLES FOR EACH GROUP.  THEY ARE KNOWN AS THE EVEN RESTART TABLES AND THE ODD
# RESTART TABLES.  THE ODD TABLES HAVE ONLY ONE ENTRY OF THREE LOCATIONS WHILE THE EVEN TABLES HAVE TWO ENTRIES
# EACH USING THREE LOCATIONS. THE INFORMATION AS TO WHETHER IT IS A JOB, WAITLIST, OR A LONGCALL IS GIVEN BY THE
# WAY THINGS ARE PUT INTO THE TABLES.
#      A JOB HAS ITS PRIORITY STORED IN PRDTTAB OF THE CORRECT PHASE SPOT - A POSITIVE PRIORITY INDICATES A
# FINDVAC JOB, A NEGATIVE PRIORITY A NOVAC.  THE 2CADR OF THE JOB IS STORED IN THE CADRTAB.
# FOR EXAMPLE,
#
# 5.7SPOT       OCT     23000
#               2CADR   SOMEJOB
#
# A RESTART OF GROUP 5 WITH PHASE SEVEN WOULD THEN CAUSE SOMEJOB TO BE RESTARTED AS A FINDVAC WITH PRIORITY 23.
#
# 5.5SPOT       OCT     -23000
#               2CADR   ANYJOB
#
# HERE A RESTART OF GROUP 5 WITH PHASE 7 WOULD CAUSE ANYJOB TO BE RESTARTED AS A NOVAC WITH PRIORITY 23.
# A LONGCALL HAS ITS GENADR OF ITS 2CADR STORED NEGATIVELY AND ITS BBCON STORED POSITIVELY.  IN ITS PRDTTAB IS
# PLACED THE LOCATION OF A DP REGISTER THAT CONTAINS THE DELTA TIME THAT LONGCALL HAD BEEN ORIGINALLY STARTED
# WITH.  EXAMPLE,
#
# 3.6SPOT       GENADR  DELTAT
#              -GENADR  LONGTASK
#               BBCON   LONGTASK
#
#               OCT     31000
#               2CADR   JOBAGAIN
#
# THIS WOULD START UP LONGTASK AT THE APPROPRIATE TIME, OR IMMEDIATELY IF THE TIME HAD ALREADY PASSED. IT SHOULD
# BE NOTED THAT IF DELTAT IS IN A SWITCHED E BANK, THIS INFORMATOIN SHOULD BE IN THE BBCON OFTHE 2CADR OF THE
# TASK.  FROM ABOVE, WE SEE THAT THE SECOND PART OF THIS PHASE WOULD BE STARTED AS A JOB WITH A PRIORITY OF 31.
#
# WAITLIST CALLS ARE IDENTIFIED BY THE FACT THAT THEIR 2CADR IS STORED NEGATIVELY. IF PRDTTAB OF THE PHASE SPOT
# IS POSITIVE, THEN IT CONTAINS THE DELTA TIME, IF PRDTTAB IS NEGATIVE THEN IT IS THE -GENADR OF AN ERASABLE
# LOCATION CONTAINING THE DELTA TIME, THAT IS, THE TIME IS STORED INDIRECTLY.  IT SHOULD BE NOTED AS ABOVE, THAT
# IF THE TIME IS STORED INDIRECTLY, THE BBCON MUST CONTAIN THE NECESSARY E BANK INFORMATION IF APPLICABLE.  WITH
# WAITLIST WE HAVE ONE FURTHER OPTION, IF -0 IS STORED IN PRDTTAB, IT WILL CAUSE AN IMMEDIATE RESTART OF THE
# TASK.  EXAMPLES,
#
#               OCT     77777           # THIS WILL CAUSE AN IMMEDIATE RESTART
#              -2CADR   ATASK           # OF THE TASK :ATASK:
#
#               DEC     200             # IF THE TIME OF THE 2 SECONDS SINCE DUMMY
#              -2CADR   DUMMY           # WAS PUT ON THE WAITLIST IS UP, IT WILL BEGIN
#                                       # IN 10 MS, OTHERWISE IT WILL BEGIN WHEN
#                                       # IT NORMALLY WOULD HAVE BEGUN.
```

```
#               -GENADR  DTIME            # WHERE DTIME CONTAINS THE DELTA TIME
#               -2CADR   TASKTASK         # OTHERWISE THIS IS AS ABOVE
#
# ***** NOW THE TABLES THEMSELVES *****

                BANK     01
                SETLOC   RESTART
                BANK

PRDTTAB         EQUALS   12000            # USED TO FIND THE PRIORITY OR DELTATIME
CADRTAB         EQUALS   12001            # THIS AND THE NEXT RELATIVE LOC CONTAIN
                                          # RESTART 2CADR

                COUNT*   $$/RSTAB         # TABLES IN BANK 1.
SIZETAB         TC       1.2SPOT -12006
                TC       1.3SPOT -12004
                TC       2.2SPOT -12006
                TC       2.3SPOT -12004
                TC       3.2SPOT -12006
                TC       3.3SPOT -12004
                TC       4.2SPOT -12006
                TC       4.3SPOT -12004
                TC       5.2SPOT -12006
                TC       5.3SPOT -12004
                TC       6.2SPOT -12006
                TC       6.3SPOT -12004
1.2SPOT         OCT      21000            # A DUMMY EXAMPLE TO BE REPLACED AS SOON
                EBANK=   STATE
                2CADR    ENDOFJOB         # AS THERE IS A LEGITIMATE 1.2SPOT

                DEC      100
                EBANK=   STATE
                2CADR    TASKOVER

# ANY MORE GROUP 1.EVEN RESTART VALUES SHOULD GO HERE

1.3SPOT         -GENADR  SAVET-30
                EBANK=   DVCNTR
                -2CADR   ULLGTASK

# ANY MORE GROUP 1.ODD RESTART VALUES SHOULD GO HERE

2.2SPOT         EQUALS   1.2SPOT
# ANY MORE GROUP 2.EVEN RESTART VALUES SHOULD GO HERE

2.3SPOT         GENADR   600SECS
                -GENADR  STATEINT
                EBANK=   RRECTCSM
                BBCON    STATEINT
```

```
2.5SPOT         OCT     05000
                EBANK=  RRECTCSM
                2CADR   STATINT1

2.7SPOT         DEC     1500
                EBANK=  LOSCOUNT
               -2CADR   P20LEMC1

2.11SPOT        OCT     14000
                EBANK=  P21TIME
                2CADR   P25LEM1

2.13SPOT        OCT     10000
                EBANK=  LOSCOUNT
                2CADR   RELINUS

2.15SPOT        OCT     26000
                EBANK=  LOSCOUNT
                2CADR   R22RSTRT

2.17SPOT        OCT     77777
                EBANK=  VGPREV
               -2CADR   REDO2.17

2.21SPOT        DEC     25
                EBANK=  DVCNTR
               -2CADR   R10,R11

# ANY MORE GROUP 2.ODD RESTART VALUES SHOULD GO HERE.

3.2SPOT         EQUALS  1.2SPOT
# ANY MORE GROUP 3.EVEN RESTART VALUES SHOULD GO HERE

3.3SPOT        -GENADR  ZOOMTIME
                EBANK=  DVCNTR
               -2CADR   ZOOM

3.5SPOT         OCT     20000
                EBANK=  TTOGO
                2CADR   S40.13

# ANY MORE GROUP 3.ODD RESTART VALUES SHOULD GO HERE

4.2SPOT         DEC     2500
                EBANK=  TTOGO
               -2CADR   TIG-5

                OCT     77777
                EBANK=  TTOGO
```

```
                -2CADR    REDO4.2

# ANY MORE GROUP 4.EVEN RESTART VALUES SHOULD GO HERE

4.3SPOT          OCT      25000
                 EBANK=   DVCNTR
                 2CADR    GOABORT

4.5SPOT          DEC      50
                 EBANK=   TTOGO
                -2CADR    ULLAGOFF

4.7SPOT          DEC      500
                 EBANK=   DVCNTR
                -2CADR    TIG-0

4.11SPOT        -GENADR   TGO +1
                 EBANK=   DVCNTR
                -2CADR    ENGOFTSK

4.13SPOT         OCT      12000
                 EBANK=   TRKMKCNT
                 2CADR    POSTBURN

4.15SPOT         DEC      500
                 EBANK=   TTOGO
                -2CADR    TIG-30

4.17SPOT         OCT      77777
                 EBANK=   DVCNTR
                -2CADR    TIG-5

4.21SPOT         OCT      13000
                 EBANK=   STAR
                 2CADR    R51.1 +1

4.23SPOT         OCT      77777
                 EBANK=   DVCNTR
                -2CADR    IGNITION

4.25SPOT         GENADR   SAVET-30
                -GENADR   TIG-35
                 EBANK=   SAVET-30
                 BBCON    TIG-35

4.27SPOT         OCT      52777
                 EBANK=   DVCNTR
                 2CADR    P70A
```

```
4.31SPOT        OCT     52777
                EBANK=  DVCNTR
                2CADR   P71A

4.33SPOT        OCT     46777
                EBANK=  DVCNTR
                2CADR   GOPOOFIX

4.35SPOT        OCT     46777
                EBANK=  DVCNTR
                2CADR   GOPOODOO

4.37SPOT        OCT     52777
                EBANK=  WHICH
                2CADR   COMFAIL

# ANY MORE 4.ODD RESTART VALUES SHOULD GO HERE.

5.2SPOT         OCT     22000
                EBANK=  DVCNTR
                2CADR   NORMLIZE

                DEC     200
                EBANK=  DVCNTR
               -2CADR   REREADAC

5.4SPOT         DEC     200
                EBANK=  DVCNTR
               -2CADR   REREADAC

                OCT     20000
                EBANK=  DVCNTR
                2CADR   SERVICER

# ANY MORE GROUP 5.EVEN RESTART VALUES SHOULD GO HERE

5.3SPOT         DEC     200
                EBANK=  DVCNTR
               -2CADR   REREADAC

5.5SPOT         OCT     77777
                EBANK=  DVCNTR
               -2CADR   REDO5.5

5.7SPOT         OCT     77777
                EBANK=  DVCNTR
```

```
                        -2CADR   BIBIBIAS

# ANY MORE GROUP 5.ODD RESTART VALUES SHOULD GO HERE

6.2SPOT         EQUALS  1.2SPOT
6.3SPOT         DEC     100
                EBANK=  TIG
               -2CADR   CLOKTASK

6.5SPOT         OCT     30000           # PROTECT INCREMENTING OF TIME2,TIME1 BY
                EBANK=  TEPHEM          # P27(UPDATE PROGRAM) VIA V70 OR V73.
                2CADR   TIMEDIDR

6.7SPOT         OCT     17000
                EBANK=  VGPREV
                2CADR   REDO6.7
```

```
                BANK    12
                SETLOC  AOTMARK1
                BANK

                EBANK=  XYMARK
                COUNT*  $$/MARK

AOTMARK         INHINT
                CCS     MARKSTAT        # SEE IF AOTMARK BUSY
                TC      +2              # MARK SYSTEM BUSY -- DO ALARM
                TC      EXTVBCHK
                TC      POODOO
                OCT     00105

EXTVBCHK        CAF     SIX             # SEE IF EXT. VERB WORKING
                MASK    EXTVBACT
                CCS     A
                TCF     MKABORT         # YES -- ABORT

                CAF     BIT2            # NO -- DISALLOW SOME EXTENDED VERB ACTION
                ADS     EXTVBACT        # BIT2 RESET IN ENDMARK
MKVAC           CCS     VAC1USE         # LOOK FOR A VAC AREAD -- DO ABORT IF
                TCF     MKVACFND        # NONE AVAILABLE
                CCS     VAC2USE
                TCF     MKVACFND
                CCS     VAC3USE
                TCF     MKVACFND
                CCS     VAC4USE
                TCF     MKVACFND
                CCS     VAC5USE
                TCF     MKVACFND
                DXCH    BUF2
                TC      BAILOUT1        # ALL VAC AREAS OCCUPIED -- ABORT.
                OCT     01207

MKVACFND        AD      TWO
                TS      MARKSTAT        # STORE VAC ADR IN LOW 9 OF MARKSTAT

                CAF     ZERO
                INDEX   MARKSTAT
                TS      0 -1            # ZERO IN VACUSE REG TO SHOW VAC OCCUPIED

                CAF     PRIO15
                TC      FINDVAC         # SET UP JOB FOR GETDAT
                EBANK=  XYMARK
                2CADR   GETDAT

                RELINT
                TCF     SWRETURN
```

```
MKABORT         DXCH    BUF2
                TC      BAILOUT1        # CONFLICT WITH EXTENDED VERB
                OCT     01211

MKRELEAS        CAF     ZERO
                XCH     MARKSTAT        # SET MARKSTAT TO ZERO
                MASK    LOW9            # PICK UP VAC AREA AOR
                CCS     A
                INDEX   A
                TS      0               # SHOW MKVAC AREA AVAILABLE
                CAF     ONE
                TC      IBNKCALL
                CADR    GOODEND         # GO WAKE UP CALLING JOB
```

```
KILLAOT         CAF     ZERO
                TS      EXTVBACT        # TERMINATE AOTMARK -- ALLOW EXT VERB
                TC      GOTOPOOH
GETDAT          CS      MARKSTAT        # SET BIT12 TO DISCOURAGE MARKRUPT
                MASK    BIT12           #       BIT12 RESET AT GETMARK
                ADS     MARKSTAT

                CAF     V01N71          # DISPLAY DETENT AND STAR CODE
                TC      BANKCALL
                CADR    GOMARKF

                TCF     KILLAOT         # V34 -- DOES GOTOPOOH
                TCF     DODAT           # V33 -- PROCEED -- USE THIS STAR FOR MARKS
ENTERDAT        TCF     GETDAT          # ENTER -- REDISPLAY STAR CODE

DODAT           CAF     HIGH9           # PICK DETENT CODE FROM BITS7-9 OF AOTCODE
                MASK    AOTCODE         # AND SEE IF CODE 1 TO 6
                EXTEND
                MP      BIT9
                TS      XYMARK          # STORE DETENT

                EXTEND
                BZMF    GETDAT          # COAS CALIBRATION CODE - NO GOOD HERE

                AD      NEG7            # SEE IF DETENT 7 FOR COAS
                EXTEND
                BZF     CODE7

                TCF     CODE1TO6

CODE7           CAF     V06N87*         # CODE 7, COAS SIGHTING, GET OPTIC AXIS
                TC      BANKCALL        # AZ AND EL OF SIGHTING DEVICE FROM ASTRO
                CADR    GOMARKF

                TCF     KILLAOT         # V34 -- DOES GOTOPOOH
                TCF     +2              # PROCEED
                TCF     CODE7           # ON ENTER, RECYCLE
                EXTEND
                DCA     AZ              # PICK UP AZ AND EL IN SP 25 COMP
                INDEX   FIXLOC
                DXCH    8D              # STORE IN 8D AND 9D OF LOCAL VAC
                CAF     ZERO            # BACKUP SYSTEM TO BE USED
                TCF     COASCODE        # ZERO APPARENT ROTATION

CODE1TO6        INDEX   XYMARK          # INDEX AOT POSITION BY DET CODE
                CA      AOTEL -1
                INDEX   FIXLOC
                TS      9D              # STORE ELEVATION IN VAC+9D

                INDEX   XYMARK          # INDEX DET CODE 1,2 OR 3
```

```
                CA      AOTAZ -1
                INDEX   FIXLOC
                TS      8D              # STORE AZIMUTH IN VAC +8D

                CA      AOTAZ +1        # COMPENSATION FOR APPARENT ROTATION OF
                EXTEND                  # AOT FIELD OF VIEW IN LEFT AND RIGHT
                INDEX   FIXLOC          # DETENTS IS STORED IN VAC +10D IN SP
                MSU     8D              # PRECISION ONE'S COMPLEMENT
COASCODE        INDEX   FIXLOC
                TS      10D             # ROT ANGLE

                TC      INTPRET         # COMPUTE X AND Y PLANE VECTORS
```

```
# THE OPTAXIS SUBROUTINE COMPUTES THE X AND Y MARK PLANE VECS AND
# ROTATES THEM THRU THE APPARENT FIELD OF VIEW ROTATION UNIQUE TO AOT
# OPTAXIS USES OANB TO COMPUTE THE OPTIC AXIS
#
#       INPUT --        AZIMUTH ANGLE IN SINGLE PREC AT CDU SCALE IN 8D OF JOB VAC
#                       ELEVATION ANGLE IN SINGLE PREC AT CDU SCALE IN 9D OF JOB VAC
#                       ROTATION ANGLE IN SINGLE PREC IS COMP SCALED BY PI IN 10D OF VAC
#
#       OUTPUT --       OPTIC AXIS VEC IN NG COORDS IN SCAXIS
#                       X-MARK PLANE 1/4VEC IN NB COORDS AT 18D OF JOB VAC
#                       Y-MARK PLANE 1/4VEC IN NB COORDS AT 12D OF JOB VAC

OPTAXIS         CALL                            # GO COMPUTE OA AN X AND Y PLANE VECS
                        OANB
                SLOAD   SR1                     # LOAD APP ROTATION IN ONES COMP
                        10D                     # RESCALE BY 2PI
                PUSH    SIN                     # 1/2SIN(ROT) 0-1
                PDDL    COS
                PUSH    VXSC                    # 1/2COS(ROT) 2-3
                        18D
                PDDL    VXSC                    # 1/4COS(ROT)UYP 4-9
                        0
                        24D                     # 1/4SIN(ROT)UXP
                BVSU    STADR                   # UP 4-9
                STODL   12D                     # YPNB=1/4(COS(ROT)UYP-SIN(ROT)UXP)
                VXSC    PDDL                    # UP 2-3 UP 0-1 FOR EXCHANGE
                        24D                     # 1/4COS(ROT)UXP        PUSH 0-5
                VXSC    VAD                     # 1/4SIN(ROT)UYP
                        18D                     # UP 0-5
                STADR
                STOVL   18D                     # XPNB=1/4(COS(ROT)UXP+SIN(ROT)UYP)
                        LO6ZEROS                # INITIALIZE AVE STAR VEC ACCUMULATOR
                STORE   STARAD +6
                EXIT
                TCF     GETMKS
```

```
# THE OANB SUBROUTINE COMPUTES THE OPTIC AXIS OF THE SIGHTING INSTRUMENT
# FROM AZIMUTH AND ELEVATION INPUT FROM THE ASTRONAUT.
#
#       INPUT --        AZIMUTH ANGLE IN SINGLE PREC 2'S COMP IN 8D OF JOB VAC
#                       ELEVATION ANGLE IN SINGLE PREC 2'S COMP IN 9D OF VAC
#
#       OUTPUT --       OPTIC AXIS IN NB COORDS. IN SCAXIS
#                       X-PLANE 1/2VEC IN NB COORDS AT 24D OF VAC
#                       Y-PLANE 1/2VEC IN NB COORDS AT 18D OF VAC

                BANK    05
                SETLOC  AOTMARK2
                BANK

                COUNT*  $$/MARK

OANB            SETPD   STQ
                        0
                        GCTR            # STORE RETURN
                SLOAD   RTB
                        9D              # PICK UP SP ELV
                        CDULOGIC
                PUSH    COS
                PDDL    SIN             # 1/2COS(ELV)   PD 0-1
                STADR
                STODL   SCAXIS          # OAX=1/2SIN(ELV)
                        8D
                RTB
                        CDULOGIC
                PUSH    COS
                STORE   20D             # STORE UYP(Y)  20-21
                PDDL    SIN             # 1/2COS(AZ)    PD 2-3
                PUSH    DCOMP           # PUSH 1/2S IN (AZ)     4-5
                STODL   22D             # STORE UYP(Z)  22-23
                        LO6ZEROS
                STODL   18D             # STORE UYP(X)  18-19
                DMP     SL1
                        0
                STODL   SCAXIS +2       # OAY=1/2COS(ELV)SIN(AZ)
                DMP     SL1             # UP    2-3
                STADR                   # UP    0-1
                STOVL   SCAXIS +4       # OAZ=1/2COS(ELV)COS(AZ)
                        18D             # LOAD UYP VEC
                VXV     UNIT
                        SCAXIS          # UXP VEC=UYP X OA
                STORE   24D             # STORE UXP
                GOTO
                        GCTR
```

```
# SURFSTAR COMPUTES A STAR VECTOR IN SM COORDINATES FOR LUNAR
# SURFACE ALIGNMENT AND EXITS TO AVEIT TO AVERAGE STAR VECTORS.
#
#       GIVEN   X-MARK PLANE 1/4 VEC IN NB AT 18D OF LOCAL VAC
#               Y-MARK PLANE 1/4 VEC IN NB AT 12D OF LOCAL VAC
#               CURSOR SP 2COMP AT POSITION 1 OF INDEXED MARKVAC
#               SPIRAL SP 2COMP AT POSITION 3 OF INDEXED MARKVAC
#               CDUY,Z,X AT POSITIONS 0,2,4 OF INDEXED MARKVAC

                BANK    15
                SETLOC  P50S
                BANK
                COUNT*  $$/R59

SURFSTAR        VLOAD*
                        0,1             # PUT X-MARK CDUS IN CDUSPOT FOR TRG*NBSM
                STORE   CDUSPOT
                SLOAD*  RTB
                        1,1             # PICK UP YROT
                        CDULOGIC
                STORE   24D             # STORE CURSOR FOR SPIRAL COMP (REVS)
                BZE
                        YZCHK           # IF YROT ZERO -- SEE IF SROT ZERO
JUSTZY          PUSH    COS
                PDDL    SIN             # 1/2COS(YROT)  0-1
                VXSC    PDDL            # UP 0-1        1/8SIN(YROT)UXP 0-5
                        18D
                VXSC    VSU             # UP    0-5
                        12D             # UYP
                UNIT    VXV
                        SCAXIS
                UNIT    PUSH
                SLOAD*  RTB
                        3,1             # PICK UP SPIRAL
                        CDULOGIC
                STORE   26D             # STORE SPIRAL (REVS)
                DSU     DAD
                        24D
                        ABOUTONE
                DMP
                        DP1/12
                STORE   26D             # SEP=(360 + SPIRAL -CURSOR)/12
                SIN     VXSC            # UP    0-5
                VSL1    PDDL            # 1/2SIN(SEP)(UPP X OA) 0-5
                        26D
                COS     VXSC
                        SCAXIS
                VSL1    VAD             # UP    0-5
JUSTOA          UNIT    CALL
                        TRG*NBSM
                STCALL  24D             # STAR VEC IN SM
                        AVEIT           # GO AVERAGE
```

```
         ABOUTONE        2DEC    .99999999

         DP1/12          EQUALS  DEG30            # .08333333
                         BANK    7
                         SETLOC  AOTMARK1
                         BANK
                         COUNT*  $$/MARK
         YZCHK           SLOAD*  BZE              # YROT ZERO AND IF SROT ZERO FORCE STAR
                                 3,1              # ALONG OPTIC AXIS
                                 YSZERO
                         DLOAD   GOTO
                                 24D
                                 JUSTZY           # SROT NOT ZERO -- CONTINUE NORMALLY
         YSZERO          VLOAD   GOTO
                                 SCAXIS
                                 JUSTOA
```

```
# THE GETMKS ROUTINE INITIALIZES THE SIGHTING MARK PROCEDURE

GETMKS          CAF     ZERO            # INITIALIZE MARK ID REGISTER AND MARK CNT
                TS      XYMARK
                TS      MARKCNTR
                CAF     LOW9            # ZERO BITS10 TO 15 RETAINING MKVAC ADR
                MASK    MARKSTAT
                TS      MARKSTAT
                CAF     MKVB54*         # DISPLAY VB54 INITIALLY
PASTIT          TC      BANKCALL
                CADR    GOMARK4

                TCF     KILLAOT         # V34 -- DOES GOTOPOOH
                TCF     MARKCHEX        # VB33 -- PROCEED, GOT MARKS, COMPUTE LOS
                TCF     GETDAT          # ENTER -- RECYCLE TO V01N71

MARKCHEX        CS      MARKSTAT        # SET BIT12 TO DISCOURAGE MARKRUPT
                MASK    BIT12
                ADS     MARKSTAT
                MASK    LOW9
                TS      XYMARK          # JAM MARK VAC ADR IN XYMARK FOR AVESTAR
                CAF     ZERO
                TS      MKDEX           # SET MKDEX ZERO FOR LOS VEC CNTR
                CA      MARKSTAT
                MASK    PRIO3           # SEE IF LAST MK PART COMPLETE
                TS      L
                CAF     PRIO3           # BITS10 AND 11
                EXTEND
                RXOR    LCHAN
                EXTEND
                BZF     AVESTAR         # LAST PAIR COMPLETE -- TO COMPUTE LOS
CNTCHK          CCS     MARKCNTR        # NO PAIR SHOWING -- SEE IF PAIR IN HOLD
                TCF     +2              # PAIR BURIED -- DECREMENT COUNTER
                TCF     MKALARM         # NO PAIR -- ALARM
                TS      MARKCNTR        # STORE DECREMENTED COUNTER

AVESTAR         CAF     BIT12           # INITIALIZE MKDEX FOR STAR LOS COUNTER
                ADS     MKDEX           # MKDEX WAS INITIALIZED ZERO IN MARKCHEX
                CS      MARKCNTR
                EXTEND
                MP      SIX             # GET C(L) = -6 MARKCNTR
                CS      XYMARK
                AD      L               # ADD -- MARK VAC ADR SET IN MARKCHEX
                INDEX   FIXLOC
                TS      X1              # JAM -- CDU ADR OF X-MARK IN X1

                CA      FIXLOC          # SET PD POINTER TO ZERO
                TS      PUSHLOC

                TC      INTPRET
```

```
                BON     VLOAD*
                        SURFFLAG        # IF ON SURFACE COMPUTE VEC AT SURFSTAR
                        SURFSTAR
                        1,1             # PUT Y-MARK CDUS IN CDUSPOT FOR TRG*NBSM
                STOVL   CDUSPOT
                        12D             # LOAD Y-PLANE VECTOR IN NG
                CALL
                        TRG*NBSM        # CONVERT IT TO STABLE MEMBER
                PUSH    VLOAD*
                        0,1             # PUT X-MARK CDUS IN CDUSPOT FOR TRG*NBSM
                STOVL   CDUSPOT
                        18D             # LOAD X-PLANE VECTOR IN NB
                CALL
                        TRG*NBSM        # CONVERT IT TO STABLE-MEMBER
                VXV     UNIT            # UNIT(XPSM * YPSM)
                STADR
                STORE   24D

AVEIT           SLOAD   PDVL            # N(NUMBER OF VECS) IN 0-1
                        MKDEX
                        24D             # LOAD CURRENT VECTOR
                VSR3    V/SC
                        0
                STODL   24D             # VEC/N
                        0
                DSU     DDV
                        DP1/8           # (N-1)/N
                VXSC    VAD
                        STARAD +6       # ADD VEC TO PREVIOUSLY AVERAGED VECTOR
                        24D             # (N-1)/N AVESTVEC + VEC/N
                STORE   STARAD +6       # AVERAGE STAR VECTOR
                STORE   STARSAV2
                EXIT
                CCS     MARKCNTR        # SEE IF ANOTHER MARK PAIR IN MKVAC
                TCF     AVESTAR -1      # THERE IS -- GO GET IT -- DECREMENT COUNTER
ENDMARKS        CAF     FIVE            # NO MORE MARKS -- TERMINATE AOTMARK
                INHINT
                TC      WAITLIST
                EBANK=  XYMARK
                2CADR   MKRELEAS

                TC      ENDMARK

MKALARM         TC      ALARM           # NOT A PAIR TO PROCESS -- DO GETMKS
                OCT     111
                TCF     GETMKS

V01N71          VN      171
V06N87*         VN      687
```

```
# MARKRUPT IS ENTERED FROM INTERRUPT LEAD-INS AND PROCESSES CHANNEL 16
# CAUSED BY X,Y MARK OR MARK REJECT OR BY THE RATE OF DESCENT SWITCH

MARKRUPT        TS      BANKRUPT
                CA      CDUY            # STORE CDUS AND TIME NOW -- THEN SEE IF
                TS      ITEMP3          # WE NEED THEM
                CA      CDUZ
                TS      ITEMP4
                CA      CDUX
                TS      ITEMP5
                EXTEND
                DCA     TIME2
                DXCH    ITEMP1
                XCH     Q
                TS      QRUPT

                CAF     OCT34           # SEE IF X OR Y MARK OR MKREJECT
                EXTEND
                RAND    NAVKEYIN
                CCS     A
                TCF     +2              # ITS A LIVE ONE -- SEE IF ITS WANTED
                TCF     SOMEKEY         # ITS SOME OTHER KEY

                CAF     BIT12           # ARE WE ASKING FOR A MARK
                MASK    MARKSTAT
                CCS     A
                TC      RESUME          # DON'T WANT MARK OR MKREJECT -- DO NOTHING

                CCS     MARKSTAT        # ARE MARKS BEING ACCEPTED
                TCF     FINDKEY         # THEY ARE -- WHICH ONE IS IT
                TC      ALARM           # MARKS NOT BEING ACCEPTED -- DO ALARM
                OCT     112
                TC      RESUME

FINDKEY         CAF     BIT5            # SEE IF MARK REJECT.
                EXTEND
                RAND    NAVKEYIN
                CCS     A
                TCF     MKREJ           # IT'S A MARK REJECT

                CAF     BIT4            # SEE IF Y MARK
                EXTEND
                RAND    NAVKEYIN
                CCS     A

                TCF     YMKRUPT         # IT'S A Y MARK

                CAF     BIT3            # SEE IF X MARK
                EXTEND
                RAND    NAVKEYIN
```

```
                CCS     A
                TCF     XMKRUPT         # IT'S A X MARK

SOMEKEY         CAF     OCT140          # NOT MARK OR MKREJECT -- SEE IF DESCENT BITS
                EXTEND
                RAND    NAVKEYIN
                EXTEND
                BZF     +3              # IF NO BITS

                TC      POSTJUMP        # IF DESCENT BITS
                CADR    DESCBITS

                TC      ALARM           # NO INBITS IN CHANNEL 16.
                OCT     113

                TC      RESUME

XMKRUPT         CAF     ZERO
                TS      RUPTREG1        # SET X MARK STORE INDEX TO ZERO
                CAF     BIT10
                TCF     +4
YMKRUPT         CAF     ONE
                TS      RUPTREG1        # SET Y MARK STORE INDEX TO ONE
                CAF     BIT11
                TS      XYMARK          # SET MARK IDENTIFICATION

                TC      MARKTYPE        # SEE IF SURFACE MARK
                TCF     SURFSTOR        # SURFACE MARK -- JUST STORE CDUS

                CAF     BIT14           # GOT A MARK -- SEE IF MARK PARI MADE
                MASK    MARKSTAT
                EXTEND
                BZF     VERIFYMK        # NOT A PAIR, NORMAL PROCEDURE
                CS      MARKCNTR        # GO A PAIR, SEE IF ANOTHER CAN BE MADE
                AD      FOUR            # IF SO, INCREMENT POINTER, CLEAR BITS 10,11
                EXTEND
                BZMF    5MKALARM        # HAVE FIVE MARK PAIRS -- DON'T ALLOW MARK
                INCR    MARKCNTR        # OK FOR ANOTHER PAIR, INCR POINTER
                CS      PRIO23          # CLEAR BITS 10,11,14 FOR NEXT PAIR
                MASK    MARKSTAT
                TS      MARKSTAT

VERIFYMK        CA      XYMARK
                MASK    MARKSTAT
                CCS     A
                TCF     +2              # THIS MARK NOT DESIRED
                TCF     VACSTOR         # MARK DESIRED -- STORE CDUS
                TC      ALARM
                OCT     114
                TC      RESUME          # RESUME -- DISPLAY UNCHANGED -- WAIT FOR ACTION
```

```
5MKALARM        TC      ALARM           # ATTEMPTING TO MAKE MORE THAN 5 MK PAIRS
                OCT     107
                TC      MARKTYPE        # SEE IF SURFACE MARK
                TCF     DSPV6N79        # IT IS
                TC      RESUME          # DON'T CHANGE DISPLAY -- DO NOTHING
```

```
MKREJ           TC      MARKTYPE        # SEE IF SURFACE
                TCF     SURFREJ         # SURFACE -- JUST CHECK MARK COUNTER

                CAF     PRIO3           # INFLIGHT -- SEE IF MARKS MADE
                MASK    MARKSTAT
                CCS     A
                TCF     REJECT          # MARKS MADE -- REJECT ONE
REJALM          TC      ALARM           # NO MARK TO REJECT -- BAD PROCEDURE -- ALARM
                OCT     115
                TC      RESUME          # DESIRED ACTION DISPLAYED

REJECT          CS      PRIO30          # ZERO BIT14, SHOW REJ., SEE IF MARK SINCE
                MASK    MARKSTAT        # LAST REJECT
                AD      BIT13
                XCH     MARKSTAT
                MASK    BIT13
                CCS     A
                TCF     REJECT2         # ANOTHER REJECT SET BIT 10+11 TO ZERO

                CS      XYMARK          # MARK MADE SINCE REJECT -- REJECT MARK IN 1D
RENEWMK         MASK    MARKSTAT
                TS      MARKSTAT
                TCF     REMARK          # GO REQUEST NEW MARK ACTION

REJECT2         CS      PRIO3           # ON SECOND REJECT -- DISPLAY VB53 AGAIN
                TCF     RENEWMK

SURFREJ         CCS     MARKCNTR        # IF MARK DECREMENT COUNTER
                TCF     +2
                TCF     REJALM          # NO MARKS TO REJECT -- ALARM
                TS      MARKCNTR
                TC      RESUME
```

```
# MARKTYPE TESTS TO SEE IF LEM ON LUNAR SURFACE.  IF IT IS RETURN TO LOC+1

MARKTYPE        CS      FLAGWRD8        # SURFFLAG ******** TEMPORARY ******
                MASK    BIT8
                CCS     A
                INCR    Q               # IF SURFACE MARK RETURN TO LOC +1
                TC      Q               # IF INFLIGHT MARK RETURN TO LOC +2

SURFSTOR        CAF     ZERO            # FOR SURFACE MARK ZERO MARK KIND INDEX
                TS      RUPTREG1

                CS      MARKSTAT        # SET BITS10,11 TO SHOW SURFACE MARK
                MASK    PRIO3           # FOR MARKCHEX
                ADS     MARKSTAT

VACSTOR         CAF     LOW9
                MASK    MARKSTAT        # STORE MARK VAC ADR IN RUPTREG2
                TS      RUPTREG2
                EXTEND
                DCA     ITEMP1          # PICK UP MARKTIME
                DXCH    TSIGHT          # STORE LAST MARK TIME
                CA      MARKCNTR        # 6 X MARKCNTR FOR STORE INDEX
                EXTEND
                MP      SIX
                XCH     L               # GET INDEX FROM LOW ORDER PART
                AD      RUPTREG2        # SET CDU STORE INDEX TO MARKVAC
                ADS     RUPTREG1        # INCREMENT VAC PICKUP BY MARK FOR FLIGHT
                TS      MKDEX           # STORE HERE IN CASE OF SURFACE MARK
                CA      ITEMP3
                INDEX   RUPTREG1
                TS      0               # STORE CDUY
                CA      ITEMP4
                INDEX   RUPTREG1
                TS      2               # STORE CDUZ
                CA      ITEMP5
                INDEX   RUPTREG1
                TS      4               # STORE CDUX
                TC      MARKTYPE        # IF SURFACE MARK -- JUST DO SURFJOB
                TCF     SURFJOB

                CAF     BIT13           # CLEAR BIT13 TO SHOW MARK MADE
                AD      XYMARK          # SET MARK ID IN MARKSTAT
                COM
                MASK    MARKSTAT
                AD      XYMARK
                TS      MARKSTAT
                MASK    PRIO3           # SEE IF X, Y MARK MADE
                TS      L
```

```
                CA      PRIO3
                EXTEND
                RXOR    LCHAN
                CCS     A
                TCF     REMARK          # NOT PAIR YET, DISPLAY MARK ACTION
                CS      MARKSTAT        # MARK PAIR COMPLETE -- SET BIT14
                MASK    BIT14
                ADS     MARKSTAT
                TCF     REMARK          # GO DISPLAY V54
```

```
REMARK          CAF     PRIO3           # BITS 10 AND 11
                MASK    MARKSTAT
                EXTEND
                MP      BIT6            # SHIFT MARK IDS TO BE 0 TO 3 FOR INDEX
                TS      MKDEX           # STORE VERB INDEX
SURFJOB         CAF     PRIO15
                TC      NOVAC           # ENTER JOB TO CHANGE DISPLAY TO
                EBANK=  XYMARK          # REQUEST NEXT ACTION
                2CADR   CHANGEVB

                TC      RESUME

CHANGEVB        TC      MARKTYPE
                TCF     DSPV6N79        # SURFACE -- DISPLAY V 06 N 79
                INDEX   MKDEX           # INFLIGHT -- PICK UP MARK VB INDEX
                CAF     MKVB54
                TC      PASTIT          # PASTE UP NEXT MK VERB DISPLAY

# THE FOUR MKVBS ARE INDEXED -- THEIR ORDER CANNOT BE CHANGED

MKVB54          VN      5471            # MAKE X OR Y MARK
MKVB53          VN      5371            # MAKE Y MARK
MKVB52          VN      5271            # MAKE X MARK
MKVB54*         VN      5471            # MAKE X OR Y MARK
DP1/8           2DEC    .125

OCT34           OCT     34
V06N71          VN      671
V06N79*         VN      679
```

```
# ROUTINE TO REQUEST CURSOR AND SPIRAL MEASUREMENTS
                COUNT*  $$/R59

DSPV6N79        CAF     V06N79*         # CURSOR -- SPIRAL DISPLAY
                TC      BANKCALL
                CADR    GOMARKF

                TCF     KILLAOT         # V34 -- DOES GOTOPOOH
                TCF     SURFEND         # V33 -- PROCEED, END MARKING
                CAF     BIT6            # IF V32(OCT40) IN MPAC DO RECYCLE
                MASK    MPAC            # OTHERWISE IT IS LOAD VB ENTER SO
                CCS     A               # RE-DISPLAY V06N79
                TCF     SURFAGAN        # VB32 -- RECYCLE
                TCF     DSPV6N79        # ENTER

SURFEND         CS      BIT14           # SET BIT14 TO SHOW MARK END
                MASK    MARKSTAT
                AD      BIT14
                TS      MARKSTAT

SURFAGAN        CA      CURSOR
                INDEX   MKDEX           # HOLDS VAC AREA POINTER FOR SURF MARKING
                TS      1               # STORE CURSOR SP 2COMP
                CA      SPIRAL
                INDEX   MKDEX
                TS      3               # STORE SPIRAL

                CS      MARKSTAT        # IF BIT 14 SET -- END MARKING
                MASK    BIT14
                EXTEND
                BZF     MARKCHEX
                CA      MARKCNTR        # THIS IS RECYCLE -- SEE IF 5 MARKS ALREADY
                AD      ONE
                COM
                AD      FIVE
                EXTEND
                BZMF    5MKALARM        # CAN'T RECYCLE -- TOO MANY MARKS -- ALARM
                INCR    MARKCNTR        # OF FOR RECYCLE -- INCR COUNTER
                TCF     GETMKS +3       # GO DISPLAY MARK VB
```

```
                BANK     7
                SETLOC   EXTVERBS
                BANK

                EBANK=   OGC

                COUNT*   $$/EXTVB

# FAN-OUT

GOEXTVB         INDEX    MPAC               # VERB-40 IS IN MPAC
                TC       LST2FAN            # FAN AS BEFORE.

LST2FAN         TC       VBZERO             # VB40 ZERO (USED WITH NOUN 20 OR 72 ONLY)
                TC       VBCOARK            # VB41 COARSE ALIGN (USED WITH NOUN 20 OR
                                            #                                72 ONLY)
                TC       IMUFINEK           # VB42 FINE ALIGN IMU
                TC       IMUATTCK           # VB43 LOAD IMU ATTITUDE ERROR METERS.
                TC       RRDESEND           # VB44 TERMINATE CONTINUOUS DESIGNATE
                TC       ALM/END            # VB45 SPARE
                TC       ALM/END            # VB46 SPARE
                TC       V47TXACT           # VB47 AGS INITIALIZATION
                TC       DAPDISP            # VB48 LOAD A/P DATA
                TCF      CREWMANU           # VB49 START AUTOMATIC ATTITUDE MANEUVER
                TC       GOLOADLV           # VB50 PLEASE PERFORM
                TC       ALM/END            # VB51 SPARE
                TC       GOLOADLV           # VB52 PLEASE MARK X -- RETICLE.
                TC       GOLOADLV           # VB53 PLEASE MARK Y -- RETICLE.
                TC       GOLOADLV           # VB54 PLEASE MARK X OR Y RETICLE
                TC       ALINTIME           # VB55 ALIGN TIME
                TC       TRMTRACK           # VB56 TERMINATE TRACKING -- P20 + P25
                TC       LRON               # VB57 PERMIT LANDING RADAR UPDATES
                TC       LROFF              # VB58 INHIBIT LANDING RADAR UPDATES
                TC       ALM/END            # VB59 SPARE
                TC       LRPOS2K            # VB60 COMMAND LR TO POSITION 2.
                TC       DAPATTER           # VB61 DISPLAY DAP ATTITUDE ERROR
                TC       TOTATTER           # VB62 DISPLAY TOTAL ATTITUDE ERROR
                TC       R04                # VB63 SAMPLE RADAR ONCE PER SECOND
                TC       VB64               # VB64 CALCULATE, DISPLAY S-BAND ANT ANGLES
                TC       SNUFFOUT           # VB65 DISABLE U,V JETS DURING DPS BURNS.
                TC       ATTACHED           # VB66 ATTACHED   MOVE THIS TO OTHER STATE
                TC       V67                # VB67 W MATRIX MONITOR
                TC       ALM/END            # VB68 SPARE
VERB69          TC       VERB69             # VB69 FORCE A HARDWARE RESTART
                TC       V70UPDAT           # VB70 UPDATE LIFTOFF TIME.
                TC       V71UPDAT           # VB71 UNIVERSAL UPDATE -- BLOCK ADDRESS.
                TC       V72UPDAT           # VB72 UNIVERSAL UPDATE -- SINGLE ADDRESS.
                TC       V73UPDAT           # VB73 UPDATE AGC TIME (OCTAL).
                TC       DNEDUMP            # VB74 INITIALIZE DOWN-TELEMETRY PROGRAM
                                            #        FOR ERASABLE DUMP.
                TC       OUTSNUFF           # VB75 ENABLE U,V JETS DURING DPS BURNS.
```

```
                TC      MINIMP          # VB76 MINIMUM IMPULSE MODE
                TC      NOMINIMP        # VB77 RATE COMMAND MODE
                TC      R77             # VB78 START LR SPURIOUS RETURN TEST
                TC      R77END          # VB79 TERMINATE LR SPURIOUS RETURN TEST
                TC      LEMVEC          # VB80 UPDATE LEM STATE VECTOR
                TC      CSMVEC          # VB81 UPDATE CSM STATE VECTOR
                TC      V82PERF         # VB82 REQUEST ORBIT PARAM DISPLAY (R30)
                TC      V83PERF         # VB83 REQUEST REND PARAM DISPLAY (R31)
                TC      ALM/END         # VB84 SPARE
                TC      VERB85          # VB85 DISPLAY RR LOS AZ AND ELEV
                TC      ALM/END         # VB86 SPARE
                TC      ALM/END         # VB87 SPARE
                TC      ALM/END         # VB88 SPARE
                TC      V89PERF         # VB89 ALIGN XORZ LEM AXIS ALONG LOS (R63)
                TC      V90PERF         # VB90 OUT OF PLANE RENDEZVOUS DISPLAY
                TC      GOSHOSUM        # VB91 DISPLAY BANK SUM.
                TC      SYSTEST         # VB92 OPERAT IMU PERFORMANCE TEST.
                TC      WMATRXNG        # VB93 CLEAR RENDWFLG
                TC      ALM/END         # VB94 SPARE
                TC      UPDATOFF        # VB95 NO STATE VECTOR UPDATE ALLOWED
                TC      VERB96          # VB96 INTERRUPT INTEGRATION AND GO TO P00
                TC      GOLOADLV        # VB97 PLEASE VERIFY ENGINE FAILURE
                TC      ALM/END         # VB98 SPARE
                TC      GOLOADLV        # VB99 PLEASE ENABLE ENGINE

# END OF EXTENDED VERB FAN

TESTXACT        CCS     EXTVBACT        # ARE EXTENDED VERBS BUSY
                TC      ALM/END         # YES, TURN ON OPERATOR LIGHT
                CA      FLAGWRD4        # ARE PRIORITY DISPLAYS USING DSKY
                MASK    OC24100
                CCS     A
                TC      ALM/END         # YES
                CAF     OCT24           # SET 3, AND 5
SETXTACT        TS      EXTVBACT        # NO.  SET FLAG TO SHOW EXT VERB DISPLAY
                                        # SYSTEM BUSY

                CA      Q
                TS      MPAC +1

                CS      TWO             # BLANK EVERYTHING EXCEPT MM AND VERB
                TC      NVSUB
                TC      +1
                TC      MPAC +1

XACTALM         TC      FALTON          # TURN ON OPERATOR ERROR LIGHT.
                TC      ENDEXT          # RELEASE MARK AND EXT. VERB DISPLAY SYS.

TERMEXTV        EQUALS  ENDEXT
```

```
ENDEXTVB        EQUALS  ENDEXT

XACTO           CAF     ZERO            # RELEASE MARK AND EXT. VERB DISPLAY SYS.
                TC      SETXTACT

ALM/END         TC      FALTON          # TURN ON OPERATOR ERROR LIGHT
GOPIN           TC      POSTJUMP
                CADR    PINBRNCH

CHKPOOH         CA      MODREG          # CHECK FOR P00 OR P00-.
                EXTEND
                BZF     TCQ
                TC      ALM/END

OC24100         OCT     24100
```

```
# VBZERO          VERB 40          DESCRIPTION
#
#       1.        REQUIRE NOUN 20 (ICDU ANGLES) OR NOUN 72 (RCDU ANGLES).
#       2.        FOR N20, CHECK IMUCADR IN AN EFFORT TO AVOID A 1210 RESTART.
#                 FOR N72, CHECK IF EITHER RADAR IS IN USE.
#       3.        EXECUTE THE CDU ZERO.
#       4.        STALL UNTIL THE ZERO IS DONE.
#       5.        DON'T DIFFERENTIATE BETWEEN A BAD OR GOOD RETURN.
#       6.        EXIT, RE-ESTABLISHING THE INTERRUPTED DISPLAY (IF ANY).

VBZERO          TC      OP/INERT
                TC      IMUZEROK        # RETURN HERE IF NOUN = ICDU(20)
                TC      RRZEROK         # RETURN HERE IF NOUN = RCDU(72)
IMUZEROK        TC      CKMODCAD
                TC      BANKCALL        # KEYBOARD REQ FOR ISS CDUZERO
                CADR    IMUZERO

                TC      BANKCALL        # STALL
                CADR    IMUSTALL
                TC      +1

                TC      GOPIN           # IMUZERO

RRZEROK         TC      RDRUSECK
                TC      BANKCALL
                CADR    RRZERO

RWAITK          TC      BANKCALL
                CADR    RADSTALL
                TCF     +1
                TC      GOPIN           # RRZERO

# LRPOS2K       VERB 60                 DESCRIPTION
#       COMMAND LANDING RADAR TO POSITION 2
#
#       1.      EXIT WITH OP ERROR IF SOMEONE IS USING EITHER RADAR.
#       2.      ALARM WITH CODE 523 IF POS 2 IS NOT INDICATED WITHIN
#               THE PRESCRIBED TIME.
#       3.      RE-ESTABLISH THE DISPLAYS.

LRPOS2K         TC      RDRUSECK
                TC      BANKCALL        # COMMAND LR TO POSITION 2
                CADR    LRPOS2
                TC      BANKCALL
                CADR    RADSTALL
                TC      LRP2ALM
                TC      GOPIN

LRP2ALM         TC      ALARM
                OCT     523
                TC      GOPIN
```

```
# V61   VERB 61, DISPLAY DAP ATTITUDE ERRORS ON FDAI ATTITUDE ERROR NEEDLES.

DAPATTER        TC      DOWNFLAG
                ADRES   NEEDLFLG
                TC      GOPIN

# V62   VERB 62, DISPLAY TOTAL ATTITUDE ERRORS ON FDAI ATTITUDE ERROR NEEDLES.

TOTATTER        TC      UPFLAG
                ADRES   NEEDLFLG
                TC      GOPIN
```

```
# VBCOARK        VERB 41         DESCRIPTION
#       COARSE ALIGN IMU OR RADAR
#
#       1.      REQUIRE NOUN 20 OR NOUN 72 OR TURN ON OPERATOR ERROR.
#       2.      REQUIRE EXT VERB DISPLAY SYS AVAILABLE OR TURN ON OPERATOR ERROR LIGHT AND GO TO PINBRNCH.
#                               CASE 1, NOUN 20 (ICDU ANGLES)
#       3.      SET EXT VERB DISPLAY ACTIVE FLAG.
#       4.      DISPLAY FLASHING V25,N22 (LOAD NEW ICDU ANGLES).
#               RESPONSES
#               A.      TERMINATE
#                       1.      RELEASE EXT VERB DISPLAY SYSTEM
#               B.      PROCEED
#                       1.      COARSE ALIGN TO THE EXISTING THETAD'S (ICORK2).
#               C.      ENTER
#                       1.      COARSE ALIGN TO THE LOADED THETAD'S (ICORK2).
# ICORK2
#       1.      RE-DISPLAY VERB 41.
#       2.      EXECUTE IMUCCARS (IMU COARSE ALIGN).
#       3.      EXECUTE IMUSTALL (ALLOW TIME FOR DATA TRANSFER).
#       4.      RELEASE EXT VERB DISPLAY SYSTEM.
#                               CASE 2, NOUN 72 (RCDU ANGLES)
#               EXIT WITH OP ERROR IF SOMEONE IS USING EITHER RADAD.
#       5.      DISPLAY FLASHING V24, N73 (LOAD NEW RR TRUNION ANGLE AND NEW SHAFT ANGLE).
#               RESPONSES
#               A.      TERMINATE
#                       1.      RELEASE EXT VERB DISPLAY SYS.
#               B.      PROCEED OR ENTER
#                       1.      EXECUTE AURLOKON (ASK OPERATOR FOR LOCK-ON REQUIREMENTS).
#                       2.      RE-DISPLAY VERB 41.
#                       3.      SCHEDULE RRDESK2 WITH PRIORITY 20.
#                       4.      RELEASE EXT VERB DISPLAY SYS.
# AURLOKON
#       1.      FLASH V04 N12 R1 = 00006 R2 = 00002
#               RESPONSES
#               A.      TERMINATE
#               B.      PROCEED
#                       1.      RESET LOCK-ON SWITCH
#                       2.      SET CONTINUOUS DESIGNATE FLAG
#                       3.      DISABLE R25
#               C.      V22 E 1 E, R1 = 00001, PROCEED
#                       1.      SET LOCK-ON SWITCH

VBCOARK         TC      OP/INERT
                TC      IMUCOARK                # RETURN HERE IF NOUN = ICDU (20)
                TC      RRDESNBK                # RETURN HERE IF NOUN = RCDU (72)

# RETURNS TO L+1 IF IMU OR L+2 IF RR.

OP/INERT        CS      OCT24
                AD      NOUNREG
                EXTEND
```

```
                BZF     TCQ                     # IF = 20.

                AD      RRIMUDIF                # -52
                EXTEND
                BZF     Q+1

                TC      ALM/END                 # ILLEGAL.

RRIMUDIF        DEC     -52                     # THE IMU
IMUCOARK        TC      CKMODCAD
                TC      TESTXACT                # COARSE ALIGN FROM KEYBOARD.
                CAF     VNLODCDU                # CALL FOR THETAD LOAD
                TC      BANKCALL
                CADR    GOXDSPF
                TC      TERMEXTV
                TCF     +1

ICORK2          CAF     IMUCOARV                # RE-DISPLAY COARSE ALIGN VERB.
                TC      BANKCALL
                CADR    EXDSPRET

                TC      BANKCALL                # CALL MODE SWITCHING PROG
                CADR    IMUCOARS

                TC      BANKCALL                # STALL
                CADR    IMUSTALL
                TC      ENDEXTVB
                TC      ENDEXTVB

VNLODCDU        VN      2522
IMUCOARV        VN      4100
```

```
# DESIGNATE TO DESIRED GIMBAL ANGLES.

RRDESNBK        TC      RDRUSECK
                TC      TESTXACT
                CA      RNDVZBIT                # IS P20 RUNNING?
                MASK    FLAGWRD0
                CCS     A
                TCF     XACTALM                 # OPERADOR ERROR IF IN P20
                CS      OCT41000                # TERMINATE PRESENT DESIGNATION
                INHINT                          # RELINT DONE IN GOXDSPF
                MASK    RADMODES
                TS      RADMODES

                CAF     VNLDRCDU                # ASK FOR GIMBAL ANGLES.
                TC      BANKCALL
                CADR    GOXDSPF
                TC      TERMEXTV
                TCF     -4                      # V33

                TC      BANKCALL                # ASK OP FOR LOCK ON REQUIREMENTS.
                CADR    AURLOKON

                CAF     OPTCOARV                # RE-DISPLAY OUR OWN VERB
                TC      BANKCALL
                CADR    EXDSPRET

                CAF     PRIO20
                TC      FINDVAC
                EBANK=  LOSCOUNT
                2CADR   RRDESK2

                TCF     TERMEXTV                # FREES DISPLAY

VNLDRCDU        VN      2473
OPTCOARV        EQUALS  IMUCOARV                # DIFFERENT NOUNS.

RRDESK2         TC      BANKCALL
                CADR    RRDESNB

                TC      +1                      # DUMMY NEEDED SINCE DESRETRN DOES INCR
                CA      PRIORITY
                MASK    LOW9
                CCS     A
                INDEX   A
                TS      A                       # RELEASE THIS JOB'S VAC AREA.
                COM                             # INSURE ENDOFJOB DOES A NOVAC END (BZMF).
                ADS     PRIORITY
                TC      BANKCALL                # WAIT FOR COMPLETION OF DESIGNATE
                CADR    RADSTALL
```

```
                TC      +2                      # BADEND -- NO LOCKON OR OUT OF LIMITS
                TC      ENDOFJOB                # GOODEND -- LOCKON ACHIEVED
                TC      ALARM
                OCT     503                     # TURN ON ALARM LIGHT -- 503 DESIGNATE FAIL

                TC      ENDOFJOB

RRDESEND        CCS     RADMODES                # TERMINATE CONTINOUS DESIGNATE ONLY
                TCF     GOPIN
                TCF     GOPIN
                TCF     +1
                CS      OCT41000                # BEGDES GOES TO ENDRADAR
                INHINT                          # RELINT DONE IN DOWNFLAG
                MASK    RADMODES
                TS      RADMODES
                TC      CLRADMOD
                CAF     1SEC
                TC      BANKCALL
                CADR    DELAYJOB
                TC      DOWNFLAG                # ENABLE R25 GIMBAL MONITOR
                ADRES   NORRMON
                TCF     GOPIN
OCT41000        OCT     41000                   # CONTINOUS DESIGNATE -- DESIGNATE
```

```
                BANK    23
                SETLOC  EXTVB1
                BANK
                COUNT*  $$/EXTVB

AURLOKON        TC      MAKECADR
                TS      DESRET
                CAF     TWO
                TS      OPTIONX +1
                CAF     SIX                     # OPTION CODE FOR V04N12
                TS      OPTIONX

 -5             CAF     V04N1272
                TC      BANKCALL                # R2    00001   LOCK-ON
                CADR    GOMARKFR
                TCF     ENDEXT                  # V34
                TCF     +5                      # V33
                TCF     -5                      # V32
                CAF     BIT3
                TC      BLANKET
                TC      ENDOFJOB

 +5             CA      OPTIONX +1
                MASK    BIT2
                CCS     A
                TCF     NOLOKON
                TC      UPFLAG
                ADRES   LOKONSW
                TCF     AURLKON1

NOLOKON         TC      DOWNFLAG                # IF NO LOCK-ON, SET BIT15 OF RADMADES TO
                ADRES   LOKONSW                 # INDICATE THAT CONTINUOUS DESIGNATION IS
                TC      UPFLAG                  # WANTED (TO BE TERMINATED BY V44.)
                ADRES   CDESFLAG
                TC      UPFLAG                  # SET NO RR ANGLE MONITOR FLAG.
                ADRES   NORRMON                 # DISABLE R25 RR GIMBAL MONITOR IN T4RUPT
AURLKON1        RELINT
                CA      DESRET
                TCF     BANKJUMP

V04N1272        VN      412
-LOKONFG        OCT     -20

                BANK    43
                SETLOC  EXTVERBS
                BANK
                COUNT*  $$/EXTVB

LRON            TC      UPFLAG          # PERMIT INCORPORATION OF LR DATA       V57
```

```
                ADRES   LRINH
                TCF     GOPIN

LROFF           TC      DOWNFLAG          # INHIBIT INCORPORATION OF LR DATA      V58
                ADRES   LRINH
                TCF     GOPIN

                EBANK=  OGC
```

```
# IMUFINEK      VERB 42         DESCRIPTION
#       FINE ALIGN IMU
#
#       1.      REQUIRE EXT VERB DISPLAY AVAILABLE AND SET BUSY FLAG OR TURN ON OPER ERROR AND GO TO PINBRNCH.
#       2.      DISPLAY FLASHING V25,N93....LOAD DELTA GYRO ANGLES....
#               RESPONSES
#               A.      TERMINATE
#                       1.      RELEASE EXT VERB DISPLAY SYSTEM.
#               B.      PROCEED OR ENTER
#                       1.      RE-DISPLAY VERB 42
#                       2.      EXECUTE IMUFINE (IMU FIVE ALIGN MODE SWITCHING).
#                       3.      EXECUTE IMUSTALL (ALLOW FOR DATA TRANSFER)
#                               A.      FAILED
#                                       1.      RELEASE EXT VERB DISPLAY SYSTEM.
#                               B.      GOOD
#                                       1.      EXECUTE IMUPULSE (TORQUE IRIGS).
#                                       2.      EXECUTE IMUSTALL AND RELEASE EXT VERB DISPLAY SYSTEM.

IMUFINEK        TC      CKMODCAD
                TC      TESTXACT                # FINE ALIGN WITH GYRO TORQUING.
                CAF     VNLODGYR                # CALL FOR LOAD OF GYRO COMMANDS
                TC      BANKCALL
                CADR    GOXDSPF
                TC      TERMEXTV
                TC      +1                      # PROCEED WITHOUT A LOAD

                CAF     IMUFINEV                # RE-DISPLAY OUR OWN VERB
                TC      BANKCALL
                CADR    EXDSPRET

                TC      BANKCALL                # CALL MODE SWITCH PROG
                CADR    IMUFINE

                TC      BANKCALL                # HIBERNATION
                CADR    IMUSTALL
                TC      ENDEXTVB

FINEK2          CAF     LGYROBIN                # PINBALL LEFT COMMANDS IN OGC REGISTERS
                TC      BANKCALL
                CADR    IMUPULSE

                TC      BANKCALL                # WAIT FOR PULSES TO GET OUT.
                CADR    IMUSTALL
                TC      ENDEXTVB
                TC      ENDEXTVB

LGYROBIN        ECADR   OGC
VNLODGYR        VN      2593
IMUFINEV        VN      4200

# GOLOADLV      VERB 50         DESCRIPTION
#       AND OTHER PLEASE
```

```
#       DO SOMETHING VERBS
#
# PLEASE PERFORM, MARK, CALIBRATE, ETC.
#
#       1.      PRESSING ENTER ON DSKY INDICATES REQUESTED ACTION HAS BEEN PERFORMED, AND THE PROGRAM DOES THE
#               SAME RECALL AS A COMPLETED LOAD.
#       2.      THE EXECUTION OF A VERB 33 (PROCEED WITHOUT DATA) INDICATES THE REQUESTED ACTION IS NOT DESIRED.

                SBANK=  PINSUPER        # FOR LOADLV1 AND SHOWSUM CADR'S

GOLOADLV        TC      FLASHOFF

                CAF     PINSUPBT
                EXTEND
                WRITE   SUPERBNK
                TC      POSTJUMP
                CADR    LOADLV1

# VERB 47 -- AGS INITIALIZATION -- R47.
#
# SEE LOG SECTION AGS INITIALIZATION FOR OTHER PERTINENT REMARKS.

V47TXACT        TC      TESTXACT        # NO OTHER EXTVERB.
                CAF     PRIO4
                TC      FINDVAC
                SBANK=  LOWSUPER
                EBANK=  AGSBUFF
                2CADR   AGSINIT

                TC      ENDOFJOB

CKMODCAD        CA      MODECADR
                EXTEND
                BZF     TCQ
                TC      ALM/END         # SOMEBODY IS USING MODECADR SO EXIT
```

```
# ALINTIME      VERB 55         DESCRIPTION
#       REQUIRE P00 OR P00-.
#
#       1.      SET EXT VERB DISPLAY BUSY FLAG.
#       2.      DISPLAY FLASHING V25,N24 (LOAD DELTA TIME FOR AGC CLOCK.
#       3.      REQUIRE EXECUTION OF VERB 23.
#       4.      ADD DELTA TIME, RECEIVED FROM INPUT REGISTER, TO THE COMPUTER TIME.
#       5.      RELEASE EXT VERB DISPLAY SYSTEM

ALINTIME        TC      TESTXACT
                TC      POSTJUMP        # NO ROOM IN 43
                CADR    R33

                BANK    42
                SETLOC  SBAND
                BANK
                COUNT*  $$/R33

R33             CAF     PRIO7
                TC      PRIOCHNG
                CAF     VNLODDT
                TC      BANKCALL
                CADR    GOXDSPF
                TC      ENDEXT          # TERMINATE
                TC      ENDEXT          # PROCEED
                CS      DEC23           # DATA IN OR RESEQUENCE (UNLIKELY)
                AD      MPAC            # RECALL LEFT VERB IN MPAC
                EXTEND
                BZF     UPDATIME        # GO AHEAD WITH UPDATE ONLY IF RECALL
                TC      ENDEXT          #       WITH V23 (DATA IN).

UPDATIME        INHINT                  # DELTA TIME IS IN DSPTEM1, +1.
                CAF     ZERO
                TS      MPAC +2         # NEEDED FOR TP AGREE
                TS      L               # ZERO T1 + 2 WHILE ALIGNING.
                DXCH    TIME2
                DXCH    MPAC
                DXCH    DSPTEM2 +1      # INCREMENT
                DAS     MPAC

                TC      TPAGREE         # FORCE SIGN AGREEMENT.
                DXCH    MPAC            # NEW CLOCK.
                DAS     TIME2
                RELINT
UPDTMEND        TC      ENDEXT

DEC23           DEC     23              # V 23

VNLODDT         VN      2524            # V25N24 FOR LOAD DELTA TIME
```

```
# SET UP FOR RADAR SAMPLING.

                BANK    42
                SETLOC  EXTVERBS
                BANK

                EBANK=  RSTACK

                COUNT*  $$/R0477

R77             TC      RDRUSECK        # TRY TO AVOID THE 1210.
                CA      FLAGWRD3        # IS R04 RUNNING?
                MASK    R04FLBIT
                CCS     A
                TC      ALM/END         # YES.
                TC      UPFLAG
                ADRES   R77FLAG
                TCF     R04Z

R04             TC      RDRUSECK        # TRY TO AVOID THE 1210.
                TC      TESTXACT
                TC      UPFLAG
                ADRES   R04FLAG         # SET R04FLAG FOR ALARMS

R04Z            CAF     EBANK4
                TS      EBANK
                CAF     1SEC+1          # SAMPLE ONCE PER SECOND
                TS      RSAMPDT
                CAF     ZERO
                TS      RTSTLOC
                TS      RFAILCNT        # ZERO BAD SAMPLE COUNTER

                INHINT
                CS      LRPOSCAL        # INITIALIZE
                MASK    RADMODES        #       BIT 9   LR RANGE LOW SCALE =0
                TS      RADMODES        #       BIT 6   LR POS 1 =0
                CAF     LRPOSCAL        #       BIT 3   RR RANGE LOW SCALE =0
                EXTEND
                RAND    CHAN33
                ADS     RADMODES
                RELINT

                CS      FLAGWRD3        # CHECK R04FLAG         R04 =1          R77 =0
                MASK    R04FLBIT
                CCS     A
                TCF     R04K

                CAF     ONE             # INDICATES RENDEZVOUS DESIRED
                TS      OPTIONX +1
R04A            CAF     BIT3            # OPTION CODE FOR V04N12
```

```

                TS      OPTIONX
                CAF     V04N12X
                TC      BANKCALL        #       R2      00001   RENDEZVOUS RADAR
                CADR    GOMARKFR        #               00002   LANDING RADAR
                TCF     R04END          # V34
                TCF     +5              # V33
                TCF     R04A    +2      # R2
                CAF     BIT3
                TC      BLANKET
                TC      ENDOFJOB

                CA      OPTIONX +1      # SAVE DESIRED OPTION   RR =1   LR =2
                TS      RTSTDEX

R04X            CAF     SIX             # RR OR LR DESIRED
                MASK    RTSTDEX
                CCS     A
                TCF     R04L            # LANDING RADAR
                TS      RTSTBASE        # FOR RR        BASE =0, MAX =1

R04B            CAF     BIT2            # IS RR AUTO MODE DISCRETE PRESENT
                EXTEND
                RAND    CHAN33
                EXTEND
                BZF     R04C            # YES

                CAF     201R04          # REQUEST SELECTION OF RR AUTO MODE
                TS      DSPTEM1
                CAF     V50N25X
                TC      BANKCALL
                CADR    GOMARK4
                TCF     R04END          # V34
                TCF     R04B            # V33
                TCF     -7              # E

R04C            CAF     BIT14           # ENABLE RR AUTO TRACKER
                EXTEND
                WOR     CHAN12

                CAF     TWO
                TS      RTSTMAX         # FOR SEQUENTIAL STORAGE

                TC      WAITLIST
                SBANK=  PINSUPER
                EBANK=  RSTACK
                2CADR   RADSAMP

                RELINT

                CS      FLAGWRD3        # CHECK R04FLAG         R04 =1          R77 =0
                MASK    R04FLBIT
```

```
                CCS     A
                TCF     GOPIN           # R77

                CAF     SIX             # RR OR LR
                MASK    RTSTDEX
                CCS     A
                TCF     R04LR           # LR

R04RR           CAF     V16N72          # DISPLAY RR CDU ANGLES (1/SEC)
                TC      BANKCALL        #       R1 + XXX.XX DEG         TRUNNION
                CADR    GOMARKF         #       R2 + XXX.XX DEG         SHAFT
                TCF     R04END          # V34   R3   BLANK
                TCF     +2              # V33
                TCF     R04RR           # V32

                CAF     V16N78          # DISPLAY RR RANGE AND RANGE RATE (1/SEC)
                TC      BANKCALL        #       R1 +- XXX.XX NM         RANGE
                CADR    GOMARKF         #       R2 +- XXXXX. FPS        RANGE RATE
                TCF     R04END          # V34   R3    BLANK
                TCF     R04Y            # V33
                TCF     R04RR           # V32

R04LR           CAF     V16N66          # DISPLAY LR RANGE AND POSITON (1/SEC)
                TC      BANKCALL        #       R1 +- XXXXX, FT         LR RANGE
                CADR    GOMARKF         #       R2 +  0000X.            POS. NO.
                TCF     R04END          # V34   R3    BLANK
                TCF     +2              # V33
                TCF     R04LR           # V32

                CAF     V16N67          # DISPLAY LR VELX, VELY, VELZ (1/SEC)
                TC      BANKCALL        #       R1 +- XXXXX. FPS        LR V(X)
                CADR    GOMARKF         #       R2 +- XXXXX. FPS        LR V(Y)
                TCF     R04END          # V34   R3 +- XXXXX. FPS        LR V(Z)
                TCF     R04Y            # V33
                TCF     R04LR           # V32

R04Y            CAF     ZERO            # TO TERMINATE SAMPLING.
                TS      RSAMPDT
                CAF     2SECS           # WAIT FOR LAST RADARUP
                TC      BANKCALL
                CADR    DELAYJOB
                CAF     1SEC+1          # SAMPLE ONCE PER SECOND
                TS      RSAMPDT
                CAF     ZERO            # FOR STORING RESULTS
                TS      RTSTLOC
                CAF     SIX
                MASK    RTSTDEX
                CCS     A
                CS      ONE             # WAS LR
                AD      TWO             # WAS RR
```

```
                TCF     R04X -1

R04K            CAF     250MS+1         # SAMPLE 4 LR COMPONENTS PER SECOND.
                TS      RSAMPDT

R04L            CAF     TWO
                TS      RTSTBASE        # FOR LR        BASE =2, MAX =3
                CAF     SIX
                TCF     R04C +4
R04END          CAF     ZERO            # ZERO RSAMPDT
                TS      RSAMPDT         # TO TERMINATE SAMPLING
                CAF     BIT8            # WAIT 1.28 SECONDS FOR POSSIBLE
                TC      BANKCALL        # PENDING RUPT.
                CADR    DELAYJOB

                INHINT
                CS      BIT14           # DISABLE RR AUTO TRACKER.
                EXTEND
                WAND    CHAN12

                TC      DOWNFLAG
                ADRES   R04FLAG         # SIGNAL END OF R04.

                TC      ENDEXT

R77END          CAF     EBANK4          # TO TERMINATE SAMPLING
                TS      EBANK
                CAF     ZERO
                TS      RSAMPDT
                CAF     BIT6            # WAIT 320 MS FOR POSSIBLE
                TC      BANKCALL        # PENDING RUPT.
                CADR    DELAYJOB

                TC      DOWNFLAG
                ADRES   R77FLAG
                TCF     GOPIN

V16N72          VN      1672
V16N78          VN      1678
V16N66          VN      1666
V16N67          VN      1667
V04N12X         VN      412
V50N25X         VN      5025
201R04          OCT     00201
1SEC+1          DEC     101
250MS+1         EQUALS  CALLCODE
LRPOSCAL        OCT     444
```

```
RDRUSECK        CS      FLAGWRD3        # IS R29 ON?
                MASK    NR29FBIT
                CCS     A
                TC      ALM/END         # YES
                CA      FLAGWRD5        # IS R77 RUNNING?
                MASK    R77FLBIT
                CCS     A
                TC      ALM/END         # YES.
                CS      FLAGWRD7        # IS SERVICER RUNNING AND HENCE POSSIBLY
                MASK    V37FLBIT        # R12 USING THE LR?
                CCS     A
                TCF     CHECKRR         # NO
                CS      FLGWRD11        # YES, IS R12 ON?
                MASK    LRBYBIT
                CCS     A
                TC      ALM/END         # YES
CHECKRR         CS      FLAGWRD1        # IS THE TRACK FLAG SET AND HENCE POSSIBLY
                MASK    TRACKBIT        # P20 USING THE RR?
                CCS     A
                TCF     CHECKP22        # NO, CHECK FOR P22.

CKRNDBIT        CA      FLAGWRD0        # YES, BUT IS IT P25?
                MASK    RNDVZBIT
                CCS     A
                TC      ALM/END
CHECKP22        CS      MODREG
                AD      DEC22
                EXTEND
                BZF     ALM/END
                TC      Q

DEC22           DEC     22

                COUNT*  $$/EXTVB

VB64            TC      CHKPOOH         # DEMAND PROGRAM 00.
                TC      TESTXACT        # IF DISPLAY SYS. NOT BUSY MAKE IT BUSY.
                CAF     PRIO4
                TC      FINDVAC
                EBANK=  ALPHASB
                2CADR   SBANDANT        # CALC., DISPLAY S-BAND ANTENNA ANGLES.

                TC      ENDOFJOB
```

```
# IMUATTCK      VERB 43         DESCRIPTION
#       LOAD IMU ATTITUDE ERROR METERS
#
#       1.      REQUIRE POO OR FRESH START.
#       2.      REQUIRE COARSE ALIGN ENABLE AND ZERO ICDU BITS OFF.
#       3.      REQUIRE THAT NEEDLES BE OFF.
#       4.      REQUEST LOAD OF N22 (VALUES TO BE DISPLAYED).
#       5.      ON PROCEED OR ENTER RE-DISPLAY V43 AND SEND PULSES.

IMUATTCK        TC      CHKPOOH         # VB 76 -- LOAD IMU ATT. ERROR METERS

                CAF     BITS4&5         # SEE IF COARSE ALIGN ENABLE AND ZERO IMU
                EXTEND                  # CDUS BITS ARE ON
                RAND    CHAN12
                CCS     A
                TCF     ALM/END         # NOT ALLOWED IF IMU COARSE OR IMU ZERO ON

                CAF     BIT13-14        # BOTH BITS 13 AND 14 MUST BE 1
                EXTEND                  # INDICATING THE MODE SELECTED IS OFF.
                RXOR    CHAN31
                MASK    BIT13-14
                EXTEND
                BZF     +2              # NEEDLES IS OFF.
                TCF     ALM/END         # EXIT.  NEEDLES IS ON.

                TC      TESTXACT

                CAF     VNLODCDU
                TC      BANKCALL
                CADR    GOXDSPF
                TC      ENDEXT          # V34
                TC      +1
                CAF     V43K            # REDISPLAY OUR VERB.
                TC      BANKCALL
                CADR    EXDSPRET
                CAF     BIT6
                EXTEND
                WOR     CHAN12          # ENABLE ERROR COUNTERS.
                CAF     TWO
                TC      WAITLIST        # PUT OUT COMMANDS IN .32 SECONDS.
                EBANK=  THETAD
                2CADR   ATTCK2

                TCF     ENDEXT

                BANK    42
                SETLOC  PINBALL3        # SOMETHING IN B42.
                BANK

                COUNT*  $$/EXTVB
```

```
ATTCK2          CAF     TWO             # PUT OUT COMMANDS.
 +1             TS      Q               # CDU WILL LIMIT EXCESS DATA.
                INDEX   A
                CA      THETAD
                EXTEND
                MP      ATTSCALE
                INDEX   Q
                XCH     CDUXCMD
                CCS     Q
                TCF     ATTCK2 +1

                CAF     13,14,15
                EXTEND
                WOR     CHAN14
                TCF     TASKOVER        # LEAVE ERROR COUNTERS ENABLED.

ATTSCALE        DEC     0.1

                BANK    7
                SETLOC  EXTVERBS
                BANK

                COUNT*  $$/EXTVB

V43K            VN      4300

# V82PERF       VERB82          DESCRIPTION
#       REQUEST ORBIT PARAMETERS DISPLAY (R30)
#
#       1.      IF AVERAGE G IS OFF:
#                       FLASH DISPLAY V04N06.  R2 INDICATES WHICH SHIP'S STATE VECTOR IS
#                       TO BE UPDATED.  INITIAL CHOICE IS THIS SHIP (R2=1).  ASTRONAUT
#                       CAN CHANGE TO OTHER SHIP BY V22EXE, WHERE X NOT EQ I.
#                       SELECTED STATE VECTOR UPDATED BY THISPREC (OTHPREC).
#                       CALLS SR30.1 (WHICH CALLS TFFCONMU + TFFRP/RA) TO CALCULATE
#                       RPER (PERIGEE RADIUS), RAPO (APOGEE RADIUS), HPER (PERIGEE
#                       HEIGHT ABOVE LAUNCH PAD OR LUNAR LANDING SITE), HAPO (APOGEE
#                       HEIGHT AS ABOVE), TPER (TIME TO PERIGEE), TFF (TIME TO
#                       INTERSECT 300 KFT ABOVE PAD OR 35KFT ABOVE LANDING SITE).
#                       FLASH MONITOR V16N44 (HAPO, HPER, TFF).  TFF IS -59M59S IF IT WAS
#                       NOT COMPUTABLE, OTHERWISE IT INCREMENTS ONCE PER SECOND.
#                       ASTRONAUT HAS OPTION TO MONITOR TPER BY KEYING IN N 32 E.
#                       DISPLAY IS IN HMS, IS NEGATIVE (AS WAS TFF), AND INCREMENTS
#                       ONCE PER SECOND ONLY IF TFF DISPLAY WAS -59M59S.
#
#       2.      IF AVERAGE G IS ON:
#                       CALLS SR30.1 APPROX EVERY TWO SECS.  STATE VECTOR IS ALWAYS
#                       FOR THIS VEHICLE.  V82 DOES NOT DISTURB STATE VECTOR.  RESULTS
#                       OF SR30.1 ARE RAPO, RPER, HAPO, HPER, TPER, TFF.
#                       FLASH MONITOR V16N44 (HAPO, HPER, TFF).
#                       IF MODE IS P11, THEN CALL DELRSPL SO ASTRONAUT CAN MONITOR
#                       RESULTS BY N50E.  SPLASH COMPUTATION DONE ONCE PER TWO SECS.
```

```
V82PERF         TC      TESTXACT

                CAF     PRIO7           # LESS THAN LAMBERT.  R30,V82
                TC      PRIOCHNG
                EXTEND
                DCA     V82CON
                TC      SUPDXCHZ        # V82CALL IN DIFF SUPERBANK FROM V82PERF

                EBANK=  HAPO
V82CON          2CADR   V82CALL

# VB83PERF      VERB 83         DESCRIPTION
#       REQUEST RENDEZVOUS PARAMETER DISPLAY (R31)
#
#       1.      SET EXT VERB DISPLAY BUSY FLAG.
#       2.      SCHEDULE R31CALL WITH PRIORITY 5.
#               A.      DISPLAY
#                       R1      RANGE
#                       R2      RANGE RATE
#                       R3      THETA

V83PERF         TC      TESTXACT

                CAF     BIT2
                TC      WAITLIST
                EBANK=  TSTRT
                2CADR   R31CALL

                TC      ENDOFJOB

# VERB 89       DESCRIPTION     RENDEZVOUS FINAL ATTITUDE ROUTINE (R63)
#
# CALLED BY VERB 89 ENTER DURING P00.  PRIO 10 IS USED.  CALCULATES AND
# DISPLAYS FINAL FDAI BALL ANGLES TO POINT LM +X OR +Z AXIS AT CSM.
#
# 1. KEY IN V 89 E ONLY IF IN PROG 00.  IF NOT IN P00, OPERATOR ERROR AND
# EXIT R63, OTHERWISE CONTINUE.
#
# 2. IF IN P00, DO IMU STATUS CHECK ROUTINE (R02BOTH).  IF IMU ON AND ITS
# ORIENTATION KNOWN TO LGC,CONTINUE.
#
# 3. FLASH DISPLAY V 04 N 06.  R2 INDICATES WHICH SPACECRAFT AXIS IS TO
# BE POINTED AT CSM.  INITIAL CHOICE IS PREFERRED (+Z) AXIS (R2=1).
# ASTRONAUT CAN CHANGE TO (+X) AXIS (R2 NOT = 1) BY V 22 E 2 E.  CONTINUE
# AFTER KEYING IN PROCEED.
#
# 4. BOTH VEHICLE STATE VECTORS UPDATED BY CONIC EQS.
#
# 5. HALF MAGNITUDE UNIT LOS VECTOR (IN STABLE MEMBER COORDINATES) AND
```

```
# HALF MAGNITUDE UNIT SPACECRAFT AXIS VECTOR (IN BODY COORDINATES)
# PREPARED FOR VECPOINT.
#
# 6. GIMBAL ANGLES FROM VECPOINT TRANSFORMED INTO FDAI BALL ANGLES BY
# BALLANGS.  FLASH DISPLAY V 06 N 18 AND AWAIT RESPONSE.
#
# 7.    RECYCLE -- RETURN TO STEP 4.
#       TERMINATE -- EXIT R63.
#       PROCEED -- RESET 3AXISFLG AND CALL R60LEM FOR ATTITUDE MANEUVER.

V89PERF         TC      CHKPOOH
                TC      TESTXACT
                CAF     PRIO10
                TC      FINDVAC
                EBANK=  RONE
                2CADR   V89CALL

                TC      ENDOFJOB

# V90PERF       VERB 90         DESCRIPTION
#       REQUEST RENDEZVOUS OUT-OF-PLANE DISPLAY (R36)
#
#       1.      SET EXT VERB DISPLAY BUSY FLAG.
#       2.      SCHEDULE R36 CALL WITH PRIORITY 10
#               A.      DISPLAY
#                       TIME OF EVENT -- HOURS, MINUTES, SECONDS
#                       Y       OUT-OF-PLANE POSITION -- NAUTICAL MILES
#                       YDOT    OUT-OF-PLANE VELOCITY -- FEET/SECOND
#                       PSI     ANGLE BTW LINE OF SIGHT AND FORWARD
#                               DIRECTION VECTOR IN HORIZONTAL PLANE -- DEGREES

V90PERF         TC      TESTXACT
                CAF     PRIO7           # R36,V90
                TC      FINDVAC
                EBANK=  RPASS36
                2CADR   R36

                TCF     ENDOFJOB

# MINIMP        VERB 76         DESCRIPTION
#       MINIMUM IMPULSE MODE
#
#       1.      SET MINIMUM IMPULSE RHO MODE FLAG TO 1.

MINIMP          INHINT
                CS      DAPBOOLS
                MASK    PULSES          # PULSES = 1 INDICATES MIN IMP MODE
                ADS     DAPBOOLS
                TCF     GOPIN           # RETURN VIA PINBRNCH

# NOMINIMP      VERB 77         DESCRIPTION
#       RATE COMMAND MODE
```

```
#
#       1.      SET MINIMUM IMPULSE RHO MODE FLAG TO O.  (ZERO INDICATES NOT MINIMUM IMPULSE MODE.).
#       2.      MOVE CDUX, CDUY, CDUZ INTO CDUXD, CDUYD, CDUZD.

NOMINIMP        INHINT
                CS      PULSES
                MASK    DAPBOOLS
                TS      DAPBOOLS        # PULSES = NOT IN MINIMUM UMPULSE MODE
                TC      IBNKCALL
                CADR    ZATTEROR
                TC      GOPIN
```

```
# CREMANU       VERB 49         DESCRIPTION
#       START AUTOMATIC ATTITUDE MANEUVER
#
#       1.      REQUIRE PROGRAM 00 ACTIVE.
#       2.      SET EXT VERB DISPLAY BUSY FLAG.
#       3.      SCHEDULE R62DISP WITH PRIORITY 10.
#       4.      RELEASE EXT VERB DISPLAY.
#
#       R62DISP
#       1.      DISPLAY FLASHING V06,N22.
#               RESPONSES
#               A.      TERMINATE
#                       1.      GOTOPOOH
#               B.      PROCEED
#                       1.      SET 3AXISFLG TO INDICATE MANEUVER IS SPECIFIED BY 3 AXIS.
#                       2.      EXECUTE R60LEM (ATTITUDE MANEUVER).
#               C.      ENTER
#                       1.      REPEAT FLASHING V06,N22.

CREWMANU        TC      CHKPOOH         # DEMAND P00

                TC      TESTXACT

                CAF     PRIO10
                TC      FINDVAC
                EBANK=  BCDU
                2CADR   R62DISP

                TC      ENDOFJOB
```

```
# TRMTRACK      VERB 56         DESCRIPTION
#       TERMINATE TRACKING (P20 AND P25).
#
#       1.      KNOCK DOWN RENDEZVOUS, TRACK, AND UPDATE FLAGS.
#       2.      REQUIRE P20 OR P25 NOT RUNNING ALONE OR GO TO GOGOPOOH (REQUEST PROGRAM 00).
#       3.      SCHEDULE V56TOVAC WITH PRIORITY 30.
#
#       V56TOVAC
#       1.      EXECUTE INTSTALL (IF INTEGRATION IS RUNNING, STALL UNTIL IT IS FINISHED.).
#       2.      ZERO GROUP 2 TO HALT P20.
#       3.      TRANSFER CONTROL TO GOPROG2 (SOFTWARE RESTART).

TRMTRACK        CA      BITS9+7         # IS REND OR P25 FLAG ON
                MASK    FLAGWRD0
                EXTEND
                BZF     GOPIN           # NO

                TC      DOWNFLAG
                ADRES   RNDVZFLG

                TC      DOWNFLAG
                ADRES   P25FLAG

                TC      DOWNFLAG        # ENSURE SEARCH FLAG IS OFF
                ADRES   SRCHOPTN

                CA      TRACKBIT        # IS TRACK FLAG ON?
                MASK    FLAGWRD1
                EXTEND
                BZF     GOPIN

                TC      POSTJUMP
                CADR    TRMTRAK1

BITS9+7         OCT     500

                SETLOC  SBAND           # BANK 42
                BANK

                COUNT*  $$/EXTVB

TRMTRAK1        TC      DOWNFLAG
                ADRES   UPDATFLG        # UPDATE FLAG DOWN
                TC      DOWNFLAG
                ADRES   TRACKFLG        # TRACK FLAG DOWN
                TC      DOWNFLAG
                ADRES   IMUSE

                TC      INTPRET
                CALL
                        INTSTALL        # DON'T INTERRUPT INTEGRATION
```

```
                EXIT

                TC      PHASCHNG
                OCT     2               # KILL GROUP 2 TO HALT P20 ACTIVITY

                INHINT
                TC      IBNKCALL        # ZERO THE COMMANDED RATES TO STOP
                CADR    STOPRATE        # MANEUVER

                TC      IBNKCALL
                CADR    RESTORDB

                TC      CLRADMOD        # CLEAR BITS 10 + 15 OF RADMODES.

                CS      BIT14           # DISABLE LOCKON
                EXTEND
                WAND    CHAN12
                TC      POSTJUMP
                CADR    GOPROG2         # CAUSE RESTART.

# DNEDUMP       VERB 74         DESCRIPTION
#       INITIALZE DOWN-TELEMETRY PROGRAM FOR ERASABLE MEMORY DUMP.
#
#       1.      SET EXT VERB DISPLAY BUSY FLAG.
#       2.      REPLACE CURRENT DOWNLIST WITH ERASABLE MEMORY.
#       3.      RELEASE EXT VERB DISPLAY.

                SETLOC  EXTVERBS
                BANK

                COUNT*  $$/EXTVB

                EBANK=  400
DNEDUMP         CAF     LDNDUMPI
                TS      DNTMGOTO
                TC      GOPIN

V74             EQUALS  DNEDUMP
LDNDUMPI        REMADR  DNDUMPI

# LEMVEC        VERB 80         DESCRIPTION
#       UPDATE LEM STATE VECTOR
#               RESET VHUPFLG TC 0

LEMVEC          TC      DOWNFLAG
                ADRES   VEHUPFLG        # VB 80 -- VEHUPFLG DOWN INDICATES LEM

                TC      NOUPDOWN

# CSMVEC        VERB 81         DESCRIPTION
#       UPDATE CSM STATE VECTOR
```

```
#               SET VEHUPFLG TO 1

CSMVEC          TC      UPFLAG
                ADRES   VEHUPFLG        # VB 81 -- VEHUPFLG UP INDICATES CSM

NOUPDOWN        TC      DOWNFLAG
                ADRES   NOUPFLAG

                TCF     GOPIN

# UPDATOFF      VERB 95         DESCRIPTION
#       INHIBIT STATE VECTOR UPDATES BY INCORP
#               SET NOUPFLAG TO 1

UPDATOFF        TC      UPFLAG          # VB 95 SET NOUPFLAG
                ADRES   NOUPFLAG

                TC      GOPIN
```

```
# SYSTEST       VERB 92         DESCRIPTION
#       OPERATE IMU PERFORMANCE TEST.
#
#       1.      REQUIRE PROGRAM 00 OR TURN ON OPERATOR ERROR.
#       2.      SET EXT VERB BUSY FLAG.

                EBANK=  QPLACE

SYSTEST         TC      CHKPOOH         # DEMAND P00

                TC      TESTXACT

                CAF     PRIO22
                TC      FINDVAC
                EBANK=  QPLACE
                SBANK=  IMUSUPER
                2CADR   REDO

                TC      ENDOFJOB

# VERB 93       CLEAR RENDWFLG, CAUSES W-MATRIX TO BE RE-INITIALIZED.

WMATRXNG        INHINT
                CS      RENDWBIT
                MASK    FLAGWRD5
                TS      FLAGWRD5

                TC      GOPIN

GOSHOSUM        EQUALS  SHOWSUM

SHOWSUM         TC      CHKPOOH         # *
                TC      TESTXACT        # *
                CAF     PRIO7           # * ALLOW OTHER CHARINS.
                TC      PRIOCHNG        # *
                CAF     S+1             # *
                TS      SKEEP6          # * SHOWSUM OPTION
                CAF     S+ZERO          # *
                TS      SMODE           # * TURN OFF SELF-CHECK
                CA      SELFADRS        # *
                TS      SELFRET         # *
                TC      STSHOSUM        # * ENTER ROPECHK

SDISPLAY        LXCH    SKEEP2          # * BANK # FOR DISPLAY
                LXCH    SKEEP3          # * BUGGER WORD FOR DISPLAY
NOKILL          CA      ADRS1           # *
                TS      MPAC +2         # *
                CA      VNCON           # * 0501
                TC      BANKCALL        # *
                CADR    GOXDSPF         # *
                TC      +3              # *
                TC      NXTBNK          # *
```

```
                TC      NOKILL          # *
                CA      SELFADRS
                TS      SKEEP1

                TC      ENDEXT          # *
VNCON           VN      501             # *
ENDSUMS         CA      SKEEP6          # *
                EXTEND                  # *
                BZF     SELFCHK         # * ROPECHK, START SELFCHK AGAIN.
                TC      STSHOSUM        # * START SHOWSUM AGAIN.
```

```
# DAPDISP       VERB 48         DESCRIPTION
#       LOAD AUTO PILOT DATA
#
#       1.      REQUIRE EXT VERB DISPLAY AVAILABLE AND SET BUSY FLAG.
#       2.      EXECUTE DAPDATA1, DAPDATA2, AND DAPDATA3.
#       3.      RELEASE EXT VERB DISPLAY SYSTEM.

DAPDISP         TC      TESTXACT
                CAF     PRIO7           # RO3
                TC      PRIOCHNG
                TC      POSTJUMP
                CADR    DAPDATA1

                BANK    34
                SETLOC  LOADDAP
                BANK

                COUNT*  $$/R03

                SBANK=  LOWSUPER        # FOR SUBSEQUENT LOW 2CADR'S

DAPDATA1        CAF     BOOLSMSK        # SET DISPLAY ACCORDING TO DAPBOOLS BITS.
                MASK    DAPBOOLS        # LM
                TS      DAPDATR1        # LM
                CS      FLGWRD10        # SET BIT 14 TO BE COMPLEMENT OF APSFLAG.
                MASK    APSFLBIT
                CCS     A
                CAF     BIT14
                ADS     DAPDATR1
CHKDATA1        CAE     DAPDATR1        # IF BITS 13 AND 14 ARE BOTH ZERO, FORCE
                MASK    BIT13-14        #       A ONE INTO BIT 13.
                EXTEND
                BZF     FORCEONE
                CAE     DAPDATR1        # ENSURE THAT NO ILLEGAL BITS SET BY CREW.
MSKDATR1        MASK    DSPLYMSK
                TS      DAPDATR1
                CAF     V01N46          # LM
                TC      BANKCALL
                CADR    GOXDSPFR
                TCF     ENDEXT          # V34E TERMINATE
                TCF     DPDAT1          # V33E PROCEED
                TCF     CHKDATA1        # E     NEW DATA        CHECK AND REDISPLAY
                CAF     REVCNT          # BITS 2 & 3:  BLANKS R2 & R3.
                TC      BLANKET
                TCF     ENDOFJOB
FORCEONE        CAF     BIT13
                ADS     DAPDATR1
                TCF     MSKDATR1

DPDAT1          INHINT                  # INHINT FOR SETTING OF FLAG BITS AND MASS
                CS      APSFLBIT        #       ON BASIS OF DISPLAYED DAPDATR1.
                MASK    FLGWRD10
                TS      L               # SET APSFLAG TO BE COMPLEMENT OF BIT 14.
```

```
                CS      DAPDATR1
                MASK    BIT14
                CCS     A
                CAF     APSFLBIT
                AD      L
                TS      FLGWRD10
                CS      DAPDATR1        # SET BITS OF DAPBOOLS ON BASIS OF DISPLAY
                MASK    BIT13-14        #       MASK OUT CSMDOCKD (BIT 13) UNLESS BOTH
                CCS     A               #       13 AND 14 ARE SET.
                CS      CSMDOCKD
                AD      BOOLSMSK
                MASK    DAPDATR1
                TS      L
                CS      BOOLSMSK
                MASK    DAPBOOLS
                AD      L
                TS      DAPBOOLS
                MASK    CSMDOCKD        # LOAD MASS IN ACCORDANCE WITH CSMDOCKD.
                CCS     A               #       MASS IS USUALLY OKAY, SO DO
                CAE     CSMMASS         #       NOT TOUCH ITS LOW-ORDER PART.
                AD      LEMMASS
                TS      MASS
                CAE     DAPBOOLS
                MASK    ACC4OR2X        # 2 OR 4 JET X-TRANSLATION
                EXTEND                  # (BIT ACC4OR2X = 1 FOR 4 JETS)
                BZF     +5
                CS      BIT15
                MASK    FLAGWRD1        # CLEAR NJTSFLAG TO 0 FOR 4 JETS
                TS      FLAGWRD1
                TCF     +4
                CS      FLAGWRD1        # SET NJTSFLAG TO 1 FOR 2 JETS
                MASK    BIT15
                ADS     FLAGWRD1
                CA      DAPBOOLS        # SELECT DESIRED KALCMANU AUTOMATIC
                MASK    THREE           # MANEUVER RATE
                DOUBLE                  # RATEINDX HAS TO BE 0,2,4,6 SINCE RATES
                TS      RATEINDX        # ARE DP
                TC      POSTJUMP
                CADR    STIKLOAD

V01N46          VN      0146
DSPLYMSK        OCT     33113
BOOLSMSK        OCT     13113

                BANK    01
                SETLOC  LOADDAP1
                BANK

                COUNT*  $$/R03

STIKLOAD        CAF     EBANK6
```

```
                TS       EBANK
                EBANK=   STIKSENS
                CA       RHCSCALE        # SET STICK SENSITIVITY TO CORRESPOND TO A
                MASK     DAPBOOLS        # MAXIMUM COMMANDED RATE (AT 42 COUNTS) OF
                CCS      A               # 20 D/S (NORMAL) OR 4 D/S (FINE), SCALED
                CA       NORMAL          # AT 45 D/S.
                AD       FINE
                TS       STIKSENS
                CA       -0.6D/S
                TS       -RATEDB         # LM-ONLY BREAKOUT LEVEL IS .6 D/S.
                CA       CSMDOCKD        # IF CSM-DOCKED, DIVIDE STICK SENSITIVITY
                MASK     DAPBOOLS        # BY 10.  NORMAL SCALING IS THEN 2 D/S AND
                EXTEND                   # FINE SCALING IS 0.4 D/S
                BZF      +7              # BRANCH IF CSM IS NOT DOCKED.
                CA       STIKSENS
                EXTEND
                MP       1/10
                TS       STIKSENS
                CA       -0.3D/S         # CSM-DOCKED BREAKOUT LEVEL IS .3 D/S.
                TS       -RATEDB
                RELINT                   # PROCEED TO NOUN 47, MASS LOAD.

DAPDATA2        CAF      V0647
                TC       BANKCALL
                CADR     GOXDSPFR
                TCF      ENDR03          # V34E  TERMINATE. FIRST SET DB.  DO 1/ACCS
                TCF      DAPDAT2         # V33E  PROCEED
                TCF      DAPDATA2        #       LOAD NEW DATA AND RECYCLE
                CAF      BIT3            # BLANKS R3
                TC       BLANKET         #                LM
                TCF      ENDOFJOB
ENDR03          INHINT
                TC       IBNKCALL
                CADR     RESTORDB
                TCF      ENDEXT          # DOES RELINT

DAPDAT2         CS       FLGWRD10        # DETERMINE STAGE FROM APSFLAG
                MASK     APSFLBIT
                CCS      A
                CA       MINLMD
                AD       MINMINLM
                AD       LEMMASS         # LEMMASS MUST BE GREATER THAN EMPTY LEM
                EXTEND
                BZMF     DAPDATA2        # ASK FOR NEW MASSES
                CAE      DAPBOOLS
                MASK     CSMDOCKD
                EXTEND
                BZF      LEMALONE        # SKIP TEST ON CSMMASS IF NOT DOCKED.
                CS       MINCSM          # TEST CSM MASS
                AD       CSMMASS         # CSMMASS MUST BE GREATER THAN EMPTY CSM
```

```
                EXTEND
                BZMF    DAPDATA2        # ASK FOR NEW MASSES
                CAE     CSMMASS         # DOCKED:  MASS = CSMMASS + LEMMASS
LEMALONE        AD      LEMMASS         # LEM ALONE:  MASS = LEMMASS
                ZL
                DXCH    MASS
                INHINT
                TC      IBNKCALL        # SET DEADBANK AND COMPUTE MOMENTS OF
                CADR    RESTORDB        #       INERTIA.
                RELINT                  # PROCEED TO NOUN 48 (OR END).

DAPDATA3        CS      FLGWRD10
                MASK    APSFLBIT
                EXTEND                  # END ROUTINE IF LEM HAS STAGED.
                BZF     ENDEXT
                CAF     V06N48          # DISPLAY TRIM ANGLES AND REQUEST RESPONSE
                TC      BANKCALL
                CADR    GOXDSPFR
                TC      ENDEXT
                TCF     DPDAT3          # V33E GO DO TRIM (WAITLIST TO TRIMGIMB)
                TCF     -5              # LOAD NEW DATA AND RECYCLE
                CAF     BIT3
                TC      BLANKET         # BLANK R3
                TCF     ENDOFJOB
DPDAT3          CAF     BIT1            # GO TO TRIMGIMB VIA WAITLIST SO IT
                INHINT                  # CAN USE FIXDELAY AND VARDELAY
                TC      WAITLIST
                EBANK=  ROLLTIME
                2CADR   TRIMGIMB

                TCF     ENDOFJOB        # DOES A RELINT
TRIMDONE        CAF     V50N48
                TC      BANKCALL        # TRIM IS FINISHED; PLEASE TERMINATE R03
                CADR    GOMARK3R
                TC      ENDEXT          # V34E TERMINATE
                TC      ENDEXT
                TC      ENDEXT
                CAF     OCT24           # BIT5 TO CHANGE TO PERFORM, 3 TO BLANK 43
                TC      BLANKET
                TCF     ENDOFJOB

V0647           VN      0647
V06N48          VN      0648

V50N48          VN      5048
NORMAL          DEC     .660214
                                        # NORMAL SCALING IS 20 D/S
FINE            DEC     .165054         # FINE STICK SCALING (4 D/S).
1/10            DEC     .1              # FACTOR FOR CSM-DOCKED SCALING
-0.6D/S         DEC     -218
```

```
-0.3D/S         DEC      -109
```

```
# VERB 66       VEHICLES ARE ATTACHED. MOVE THIS VEHICLE STATE VECTOR TO
#               OTHER VEHICLE STATE VECTOR.
#
# USE SUBROUTINE GENTRAN.

                BANK    7
                SETLOC  EXTVERBS
                BANK

                COUNT*  $$/EXTVB

                EBANK=  RRECTHIS

ATTACHED        CAF     PRIO10
                TC      FINDVAC
                EBANK=  RRECTHIS

                2CADR   ATTACHIT

                TC      ENDOFJOB

ATTACHIT        TC      INTPRET
                CALL
                        INTSTALL
                SET     BON
                        MOONOTH
                        MOONTHIS
                        +3
                CLEAR
                        MOONOTH
                EXIT
                CAF     OCT51
                TC      GENTRAN
                ADRES   RRECTHIS        # OUR STATE VECTOR INTO OTHER VIA GENTRAN
                ADRES   RRECTOTH

                RELINT
                TC      INTPRET
                CALL                    # UPDATE R-OTHER, V-OTHER
                        PTOALEM
                LXA,2   CALL
                        PBODY
                        SVDWN1
                EXIT

                CAF     TCPINAD
                INDEX   FIXLOC
                TS      QPRET
                TC      POSTJUMP
                CADR    INTWAKE         # FREE INTEGRATION AND EXIT.
```

```
TCPIN           RTB
                        PINBRNCH

OCT51           OCT     51
TCPINAD         CADR    TCPIN

# VERB 96       SET QUITFLAT TO STOP INTEGRATION.
#
#       GO TO V37 WITH ZERO TO CAUSE P00.
#               STATEINT WILL CHECK QUITFLAG AND SKIP 1ST PASS,
#                       THUS ALLOWING A 10 MINUT PERIOD WITHOUT INTEGRATION.

VERB96          TC      UPFLAG          # QUITFLAG WILL CAUSE INTEGRATION TO EXIT
                ADRES   QUITFLAG        #       AT NEXT TIMESTEP

                CAF     ZERO
                TC      POSTJUMP
                CADR    V37             # GO TO P00

# VERB 67:      DISPLAY OF W MATRIX

V67             TC      TESTXACT
                CAF     PRIO5
                TC      FINDVAC
                EBANK=  WWPOS
                2CADR   V67CALL

                TC      ENDOFJOB

# VERB 65       DISABLE U,V JETS DURING DPS BURNS

SNUFFOUT        TC      UPFLAG
                ADRES   SNUFFER
                TC      GOPIN

# VERB 75       ENABLE U,V JETS DURING DPS BURNS

OUTSNUFF        TC      DOWNFLAG
                ADRES   SNUFFER
                TC      GOPIN

# VERB 85       DISPLAY RR LOS AZIMUTH AND ELEVATION.
#
# AZIMUTH IS THE ANGLE BETWEEN THE LOS AND THE X-Z NB PLANE, 0-90 DEG IN THE +Y HEMISPHERE,
# 360-270 DEG IN THE -Y HEMISPHERE.
#
# ELEVATION IS THE ANGLE BETWEEN +ZNB AND THE PROJECTION OF THE LOS INTO THE X-Z PLANE, 0-360 ABOUT +Y.

                EBANK=  RR-AZ
VERB85          TC      TESTXACT
```

```
                TC      POSTJUMP
                CADR    DSPRRLOS

                SETLOC  PINBALL1
                BANK

                COUNT*  $$/EXTVB

DSPRRLOS        CAF     PRIO5
                TC      FINDVAC
                EBANK=  RR-AZ
                2CADR   RRLOSDSP

                CAF     PRIO4
                TC      PRIOCHNG
                CAF     V16N56
                TC      BANKCALL
                CADR    GOMARKFR
                TC      B5OFF
                TC      B5OFF
                TC      B5OFF

                CAF     BIT3
                TC      BLANKET
                TC      ENDOFJOB

RRLOSDSP        EXTEND
                DCA     CDUT
                DXCH    MPAC
                TC      INTPRET
                CALL
                        RRNBMPAC        # GET RR LOS IN BODY AXIS.
                STORE   0D              # UNIT LOS
                STODL   6D
                        HI6ZEROS
                STOVL   8D
                        6D
                UNIT
                STORE   6D              # UNIT OF LOS PROJ IN X-Z PLANE
                DOT
                        UNITZ
                STOVL   COSTH           # 16D
                        UNITX
                DOT
                        6D
                STCALL  SINTH           # 18D
                        ARCTRIG
                BPL     DAD             # INSURE DISPLAY OF 0-360 DEG.
                        +2
                        DPPOSMAX        # INTRODUCES AND ERROR OF B-28 REVS.
```

```
                STOVL   RR-ELEV
                        OD
                DOT
                        UNITY
                STOVL   SINTH
                        OD
                DOT
                        6D
                STCALL  COSTH
                        ARCTRIG
                BPL     DAD             # INSURE DISPLAY OF 0-360 DEG.
                        +2
                        DPPOSMAX        # INTRODUCES AN ERROR OF B-28 REVS.
                STORE   RR-AZ
                EXIT
                CA      1SEC
                TC      BANKCALL
                CADR    DELAYJOB

                CA      BIT5
                MASK    EXTVBACT
                CCS     A
                TC      RRLOSDSP
                TC      ENDEXT

V16N56          VN      1656
```

```
# THE FOLLOWING REFERS TO THE NOUN TABLES
#
# COMPONENT CODE NUMBER          INTERPRETATION
#
#       00000                    1 COMPONENT
#       00001                    2 COMPONENT
#       00010                    3 COMPONENT
#       X1XXX                    BIT 4 = 1.  DECIMAL ONLY
#       1XXXX                    BIT 5 = 1.  NO LOAD
# END OF COMPONENT CODE NUMBER
#
# SF ROUTINE CODE NUMBER         INTERPRETATION
#
#       00000                    OCTAL ONLY
#       00001                    STRAIGHT FRACTIONAL
#       00010                    CDU DEGREES (XXX.XX)
#       00011                    ARITHMETIC SF
#       00100                    ARITH DP1       OUT (MULT BY 2EXP14 AT END)     IN (STRAIGHT)
#       00101                    ARITH DP2       OUT (STRAIGHT)                  IN (SL 7 AT END)
#       00110                    LANDING RADAR POSITION (+0000X)
#       00111                    ARITH DP3       OUT (SL 7 AT END)               IN (STRAIGHT)
#       01000                    WHOLE HOURS IN R1, WHOLE MINUES (MOD 60) IN R2,
#                                        SECONDS (MOD 60) 0XX.XX IN R3.  *** ALARMS IF USED WITH OCTAL
#       01001                    MINUTES (MOD 60) IN D1D2, D3 BLANK, SECONDS (MOD 60) IN D4D5
#                                        LIMITS TO 59B59 IF MAG EXCEEDS THIS VALUE.
#                                        ALARMS IF USED WITH OCTAL ******** IN (ALARM)
#       01010                    ARITH DP4       OUT (STRAIGHT)                  IN (SL 3 AT END)
#       01011                    ARITH1 SF       OUT (MULT BY 2EXP14 AT END)     IN (STRAIGHT)
#       01100                    2 INTEGERS IN D1D2, D4D5, D3 BLANK.
#                                        ALARMS IF USED WITH OCTAL ******** IN (ALARM)
#       01101                    360-CDU DEGREES (XXX.XX)
#
# END OF SF ROUTINE CODE NUMBERS

# SF CONSTANT CODE NUMBER        INTERPRETATION
#
#       00000                    WHOLE                           USE ARITH
#       00000                    DP TIME SEC (XXX.XX SEC)        USE ARITHDP1
#       00000                    LR POSITION (+0000X)            USE LR POSITION
#       00001                    SPARE
#       00010                    CDU DEGREES                     USE CDU DEGREES
#       00010                    360-CDU DEGREES                 USE 360-CDU DEGREES
#       00011                    DP DEGREES (90 XX.XXX DEG       USE ARITHDP3
#       00100                    DP DEGREES (360) XXX.XX DEG     USE ARITHDP4
#       00101                    DEGREES (180) XXX.XX DEG        USE ARITH
#       00101                    OPTICAL TRACKER AZIMUTH ANGLE (XXX.XXDEG)
#                                                                USE ARITHDP1
#       00110                    WEIGHT2 (XXXXX. LBS)            USE ARITH1
```

```
#       00111                   POSITION5 (XXX.XX NAUTICAL MILES)
#                                                               USE ARITHDP3
#       01000                   POSITION4 (XXXX.X NAUTICAL MILES)
#                                                               USE ARITHDP3
#       01001                   VELOCITY2 (XXXXX. FT/SEC)       USE ARITHDP4
#       01010                   VELOCITY3 (XXXX.X FT/SEC)       USE ARITHDP3
#       01011                   ELEVATION DEGREES (89.999 MAX)  USE ARITH
#       01100                   RENDEZVOUS RADAR RANGE (XXX.XX NAUT MI)
#                                                               USE ARITHDP1
#       01101                   RENDEZVOUS RADAR RANGE RATE (XXXXX.FT/SEC)
#                                                               USE ARITHDP1
#       01110                   LANDING RADAR ALTITUDE (XXXXX.FEET)
#                                                               USE ARITHDP1
#       01111                   INITIAL/FINAL ALTITUDE (XXXXX. FEET)
#                                                               USE ARITHDP1
#       10000                   ALTITUDE RATE (XXXXX.FT/SEC)    USE ARITH
#       10001                   FORWARD/LATERAL VELOCITY (XXXXX.FEET/SEC)
#                                                               USE ARITH
#       10010                   ROTATIONAL HAND CONTROLLER ANGLE RATES
#                                       XXXXX.DEG/SEC           USE ARITH
#       10011                   LANDING RADAR VELX (XXXXX.FEET/SEC)
#                                                               USE ARITHDP1
#       10100                   LANDING RADAR VELY (XXXXX.FEET/SEC)
#                                                               USE ARITHDP1
#       10101                   LANDING RADAR VELZ (XXXXX.FEET/SEC)
#                                                               USE ARITHDP1
#       10110                   POSITION7 (XXXX.X NAUT MI)      USE ARITHDP4
#       10111                   TRIM DEGREES2 (XXX.XX DEG)      USE ARITH
#       11000                   COMPUTED ALTITUDE (XXXXX. FEET)
#                                                               USE ARITHDP1
#       11001                   DP DEGREES (XXXX.X DEG)         USE ARITHDP3
#       11010                   POSITION9 (XXXX.X FT)           USE ARITHDP3
#       11011                   VELOCITY4 (XXXX.X FT/SEC)       USE ARITHDP2
#       11100                   RADIANS (XXX.XXX RADIANS)       USE ARITHDP4
#
# END OF SF CONSTANT CODE NUMBERS

# FOR GREATER THAN SINGLE PRECISION SCALES, PUT ADDRESS OF MAJOR PART INTO
# NOUN TABLES.

# OCTAL LOADS PLACE +0 INTO MAJOR PART, DATA INTO MINOR PART.

# OCTAL DISPLAYS SHOW MINOR PART ONLY.

# TO GET AT BOTH MAJOR AND MINOR PARTS (IN OCTAL), USE NOUN 01.

# A NOUN MAY BE DECLARED :DECIMAL ONLY: BY MAKING BIT4=1 OF ITS COMPONENT
# CODE NUMBER.  IF THIS NOUN IS USED WITH ANY OCTAL DISPLAY VERB, OR IF
# DATA IS LOADED IN OCTAL, IT ALARMS.


# IN LOADING AN :HOURS, MINUTES, SECONDS: NOUN, ALL 3 WORDS MUST BE
# LOADED, OR ALARM.
```

```
# ALARM IF AN ATTEMPT IS MADE TO LOAD :SPLIT MINUTES/SECONDS: (MMBSS).
# THIS IS USED FOR DISPLAY ONLY.
```

```
# THE FOLLOWING ROUTINES ARE FOR READING THE NOUN TABLES AND THE SF TABLES
# (WHICH ARE IN A SEPARATE BANK FROM THE REST OF PINBALL).  THESE READING
# ROUTINES ARE IN THE SAME BANK AS THE TABLES.  THEY ARE CALLED BY DXCH Z.

# LODNNTAB LOADS NNADTEM WITH THE NNADTAB ENTRY, NNTYPTEM WITH THE
# NNTYPTAB ENTRY.  IF THE NOUN IS MIXED, IDADITEM IS LOADED WITH THE FIRST
# IDADDTAB ENTRY, IDAD2TEM THE SECOND IDADDTAB ENTRY, IDAD3TEM THE THIRD
# IDADDTAB ENTRY, RUTMXTEM WITH THE RUTMXTAB ENTRY.  MIXBR IS SET FOR
# MIXED OR NORMAL NOUN.

                BANK    6
                SETLOC  PINBALL3
                BANK
                COUNT*  $$/NOUNS
LODNNTAB        DXCH    IDAD2TEM                # SAVE RETURN INFO IN IDAD2TEM, IDAD3TEM.
                INDEX   NOUNREG
                CAF     NNADTAB
                TS      NNADTEM
                INDEX   NOUNREG
                CAF     NNTYPTAB
                TS      NNTYPTEM
                CS      NOUNREG
                AD      MIXCON
                EXTEND
                BZMF    LODMIXNN                # NOUN NUMBER G/E FIRST MIXED NOUN
                CAF     ONE                     # NOUN NUMBER L/ FIRST MIXED NOUN
                TS      MIXBR                   # NORMAL.  +1 INTO MIXBR
                TC      LODNLV
LODMIXNN        CAF     TWO                     # MIXED.  +2 INTO MIXBR.
                TS      MIXBR
                INDEX   NOUNREG
                CAF     RUTMXTAB -40D           # FIRST MIXED NOUN = 40.
                TS      RUTMXTEM
                CAF     LOW10
                MASK    NNADTEM
                TS      Q                       # TEMP
                INDEX   A
                CAF     IDADDTAB
                TS      IDAD1TEM                # LOAD IDAD1TEM WITH FIRST IDADDTAB ENTRY
                EXTEND
                INDEX   Q                       # LOAD IDAD2TEM WITH 2ND IDADDTAB ENTRY
                DCA     IDADDTAB +1             # LOAD IDAD3TEM WITH 3RD IDADDTAB ENTRY.
LODNLV          DXCH    IDAD2TEM                # PUT RETURN INFO INTO A, L.
                DXCH    Z

MIXCON          =       OCT50                   # (DEC 40)

# GTSFOUT LOADS SFTEMP1, SFTEMP2 WTIH THE DP SFOUTAB ENTRIES.

GTSFOUT         DXCH    SFTEMP1                 # 2X(SFCONUM) ARRIVES IN SFTEMP1.
```

```
                EXTEND
                INDEX   A
                DCA     SFOUTAB
SFCOM           DXCH    SFTEMP1
                DXCH    Z

# GTSFIN LOADS SFTEMP1, SFTEMP2 WITH THE DP SFINTAB INTRIES.

GTSFIN          DXCH    SFTEMP1                 # 2X(SFCONUM) ARIVES IN SFTEMP1.
                EXTEND
                INDEX   A
                DCA     SFINTAB
                TCF     SFCOM

                                                # NN    NORMAL NOUNS
NNADTAB         OCT     00000                   # 00    NOT IN USE
                OCT     40000                   # 01    SPECIFY MACHINE ADDRESS (FRACTIONAL)
                OCT     40000                   # 02    SPECIFY MACHINE ADDRESS (WHOLE)
                OCT     40000                   # 03    SPECIFY MACHINE ADDRESS (DEGREES)
                ECADR   DSPTEM1                 # 04    ANGULAR ERROR/DIFFERENCE
                ECADR   DSPTEM1                 # 05    ANGULAR ERROR/DIFFERENCE
                ECADR   OPTION1                 # 06    OPTION CODE
                ECADR   XREG                    # 07    ECADR OF WORD TO BE MODIFIED
                                                #       ONES FOR BITS TO BE MODIFIED
                                                #       1 TO SET OR 0 TO RESET SELECTED BITS
                ECADR   ALMCADR                 # 08    ALARM DATA
                ECADR   FAILREG                 # 09    ALARM CODES
                OCT     77776                   # 10    CHANNEL TO BE SPECIFIED
                ECADR   TCSI                    # 11    TIG OF CSI (HRS,MIN,SEC)
                ECADR   OPTIONX                 # 12    OPTION CODE
                                                #       (USED BY EXTENDED VERBS ONLY)
                ECADR   TCDH                    # 13    TIG OF CDH (HRS,MIN,SEC)
                ECADR   DSPTEMX                 # 14    CHECKLIST
                                                #       (USED BY EXTENDED VERBS ONLY)
                OCT     77777                   # 15    INCREMENT MACHINE ADDRESS
                ECADR   DSPTEMX                 # 16    TIME OF EVENT (HRS,MIN,SEC)
                OCT     00000                   # 17    SPARE
                ECADR   FDAIX                   # 18    AUTO MANEUVER BALL ANGLES
                OCT     00000                   # 19    SPARE
                ECADR   CDUX                    # 20    ICDU ANGLES
                ECADR   PIPAX                   # 21    PIPAS
                ECADR   THETAD                  # 22    NEW ICDU ANGLES
                OCT     00000                   # 23    SPARE
                ECADR   DSPTEM2 +1              # 24    DELTA TIME FOR AGC CLOCK (HRS,MIN,SEC)
                ECADR   DSPTEM1                 # 25    CHECKLIST
                                                #       (USED WTIH PLEASE PERFORM ONLY)
                ECADR   DSPTEM1                 # 26    PRIO/DELAY, ADRES, BBCON
                ECADR   SMODE                   # 27    SELF TEST ON/OFF SWITCH
```

```
                OCT     00000                   # 28    SPARE
                OCT     00000                   # 29    SPARE
                OCT     0                       # 30    SPARE
                OCT     0                       # 31    SPARE
                ECADR   -TPER                   # 32    TIME TO PERIGEE (HRS,MIN,SEC)
                ECADR   TIG                     # 33    TIME OF IGNITION (HRS,MIN,SEC)
                ECADR   DSPTEM1                 # 34    TIME OF EVENT (HRS,MIN,SEC)
                ECADR   TTOGO                   # 35    TIME TO GO TO EVENT (HRS,MIN,SEC)
                ECADR   TIME2                   # 36    TIME OF AGC CLOCK (HRS,MIN,SEC)
                ECADR   TTPI                    # 37    TIG OF TPI (HRS,MIN,SEC)
                ECADR   TET                     # 38    TIME OF STATE BEING INTEGRATED
                OCT     00000                   # 39    SPARE

# END OF NNADTAB FOR NORMAL NOUNS

                                                # NN    MIXED NOUNS
                OCT     64000                   # 40    TIME TO IGNITION/CUTOFF
                                                #       VG
                                                #       DELTA V (ACCUMULATED)
                OCT     02003                   # 41    TARGET  AZIMUTH
                                                #               ELEVATION
                OCT     24006                   # 42    APOGEE
                                                #       PERIGEE
                                                #       DELTA V (REQUIRED)
                OCT     24011                   # 43    LATITUDE
                                                #       LONGITUDE
                                                #       ALTITUDE
                OCT     64014                   # 44    APOGEE
                                                #       PERIGEE
                                                #       TFF
                OCT     64017                   # 45    MARKS
                                                #       TTI OF NEXT BURN
                                                #       MGA
                OCT     00022                   # 46    AUTOPILOT CONFIGURATION
                OCT     22025                   # 47    LEM WEIGHT
                                                #       CSM WEIGHT
                OCT     22030                   # 48    GIMBAL PITCH TRIM
                                                #       GIMBAL ROLL TRIM
                OCT     24033                   # 49    DELTA R
                                                #       DELTA V
                                                #       RADAR DATA SOURCE CODE
                OCT     0                       # 50    SPARE
                OCT     22041                   # 51    S-BAND ANTENNA  PITCH
                                                #                       YAW
                OCT     00044                   # 52    CENTRAL ANGLE OF ACTIVE VEHICLE
                OCT     00000                   # 53    SPARE
                OCT     24052                   # 54    RANGE
                                                #       RANGE RATE
                                                #       THETA
                OCT     24055                   # 55    NO. OF APSIDAL CROSSINGS
```

```
                                        #       ELEVATION ANGLE
                                        #       CENTRAL ANGLE
                OCT     02060           # 56    RR LOS  AZIMUTH
                                        #               ELEVATION
                OCT     20063           # 57    DELTA R
                OCT     24066           # 58    PERIGEE ALT
                                        #       DELTA V TPI
                                        #       DELTA V TPF
                OCT     24071           # 59    DELTA VELOCITY LOS
                OCT     24074           # 60    HORIZONTAL VELOCITY
                                        #       ALTITUDE RATE
                                        #       COMPUTED ALTITUDE
                OCT     64077           # 61    TIME TO GO IN BRAKING PHASE
                                        #       TIME TO IGNITION
                                        #       CROSS RANGE DISTANCE
                OCT     64102           # 62    ABSOLUTE VALUE OF VELOCITY
                                        #       TIME TO IGNITION
                                        #       DELTA V (ACCUMULATED)
                OCT     24105           # 63    ABSOLUTE VALUE OF VELOCITY
                                        #       ALTITUDE RATE
                                        #       COMPUTED ALTITUDE
                OCT     64110           # 64    TIME LEFT FOR REDESIGNATION -- LPD ANGLE
                                        #       ALTITUDE RATE
                                        #       COMPUTED ALTITUDE
                OCT     24113           # 65    SAMPLED AGC TIME (HRS,MIN,SEC)
                                        #       (FETCHED IN INTERRUPT)
                OCT     62116           # 66    LR      RANGE
                                        #               POSITION
                OCT     04121           # 67    LRVX
                                        #       LRVY
                                        #       LRVZ
                OCT     64124           # 68    SLANT RANGE TO LANDING SIGHT
                                        #       TIME TO GO IN BRAKING PHASE
                                        #       LR ALTITUDE -- COMPUTED ALTITUDE
                OCT     00000           # 69    SPARE
                OCT     04132           # 70    AOT DETENT CODE/STAR CODE
                OCT     04135           # 71    AOT DETENT CODE/STAR CODE
                OCT     02140           # 72    RR      360 -- TRUNNION ANGLE
                                        #               SHAFT ANGLE
                OCT     02143           # 73    NEW RR  360 -- TRUNNION ANGLE
                                        #               SHAFT ANGLE
                OCT     64146           # 74    TIME TO IGNITION
                                        #       YAWAFTER VEHICLE RISE
                                        #       PITCH AFTER VEHICLE RISE
                OCT     64151           # 75    DELTA ALTITUDE CDH
                                        #       DELTA TIME (CDH-CSI OR TPI-CDH)
                                        #       DELTA TIME (TPI-CDH OR TPI-NOMTPI)
                OCT     24154           # 76    DESIRED HORIZONTAL VELOCITY
                                        #       DESIRED RADIAL VELOCITY
                                        #       CROSS-RANGE DISTANCE
```

```
                OCT     62157           # 77    TIME TO ENGINE CUTOFF
                                        #       VELOCITY NORMAL TO CSM PLANE
                OCT     02162           # 78    RR      RANGE
                                        #               RANGE RATE
                OCT     24165           # 79    CURSOR ANGLE
                                        #       SPIRAL ANGLE
                                        #       POSITION CODE
                OCT     02170           # 80    DATA INDICATOR
                                        #       OMEGA
                OCT     24173           # 81    DELTA V (LV)
                OCT     24176           # 82    DELTA V (LV)
                OCT     24201           # 83    DELTA V (BODY)
                OCT     24204           # 84    DELTA V (OTHER VEHICLE)
                OCT     24207           # 85    VG (BODY)
                OCT     24212           # 86    VG (LV)
                OCT     02215           # 87    BACKUP OPTICS LOS       AZIMUTH
                                        #                               ELEVATION
                OCT     24220           # 88    HALF UNIT SUN OR PLANET VECTOR
                OCT     24223           # 89    LANDMARK        LATITUDE
                                        #                       LONGITUDE/2
                                        #                       ALTITUDE
                OCT     24226           # 90    Y
                                        #       Y DOT
                                        #       PSI
                OCT     04231           # 91    ALTITUDE
                                        #       VELOCITY
                                        #       FLIGHT PATH ANGLE
                OCT     00000           # 92    SPARE
                OCT     04237           # 93    DELTA GYRO ANGLES
                OCT     00000           # 94    SPARE
                OCT     0               # 95    SPARE
                OCT     0               # 96    SPARE
                OCT     04253           # 97    SYSTEM TEST INPUTS
                OCT     04256           # 98    SYSTEM TEST RESULTS
                OCT     24261           # 99    RMS IN POSITION
                                        #       RMS IN VELOCITY
                                        #       RMS IN BIAS

# END OF NNADTAB FOR MIXED NOUNS

                                        # NN    NORMAL NOUNS
NNTYPTAB        OCT     00000           # 00    NOT IN USE
                OCT     04040           # 01    3COMP FRACTIONAL
                OCT     04140           # 02    3COMP WHOLE
                OCT     04102           # 03    3COMP CDU DEGREES
                OCT     00504           # 04    1COMP DPDEG(360)
                OCT     00504           # 05    1COMP DPDEG(360)
                OCT     04000           # 06    3COMP OCTAL ONLY
                OCT     04000           # 07    3COMP OCTAL ONLY
                OCT     04000           # 08    3COMP OCTAL ONLY
```

```
                OCT     04000                   # 09    3COMP OCTAL ONLY
                OCT     00000                   # 10    1COMP OCTAL ONLY
                OCT     24400                   # 11    3COMP HMS (DEC ONLY)
                OCT     02000                   # 12    2COMP OCTAL ONLY
                OCT     24400                   # 13    3COMP HMS (DEC ONLY)
                OCT     04140                   # 14    3COMP WHOLE
                OCT     00000                   # 15    1COMP OCTAL ONLY
                OCT     24400                   # 16    3COMP HMS (DEC ONLY)
                OCT     0                       # 17    SPARE
                OCT     04102                   # 18    3COMP CDU DEG
                OCT     00000                   # 19    SPARE
                OCT     04102                   # 20    3COMP CDU DEGREES
                OCT     04140                   # 21    3COMP WHOLE
                OCT     04102                   # 22    3COMP CDU DEGREES
                OCT     00000                   # 23    SPARE
                OCT     24400                   # 24    3COMP HMS (DEC ONLY)
                OCT     04140                   # 25    3COMP WHOLE
                OCT     04000                   # 26    3COMP OCTAL ONLY
                OCT     00140                   # 27    1COMP WHILE
                OCT     00000                   # 28    SPARE
                OCT     00000                   # 29    SPARE
                OCT     0                       # 30    SPARE
                OCT     0                       # 31    SPARE
                OCT     24400                   # 32    3COMP HMS (DEC ONLY)
                OCT     24400                   # 33    3COMP HMS (DEC ONLY)
                OCT     24400                   # 34    3COMP HMS (DEC ONLY)
                OCT     24400                   # 35    3COMP HMS (DEC ONLY)
                OCT     24400                   # 36    3COMP HMS (DEC ONLY)
                OCT     24400                   # 37    3COMP HMS (DEC ONLY)
                OCT     24400                   # 38    3COMP HMS (DEC ONLY)
                OCT     00000                   # 39    SPARE

# END OF NNTYPTAB FOR NORMAL NOUNS

                                                # NN    MIXED NOUNS
                OCT     24500                   # 40    3COMP   MIN/SEC, VEL3, VEL3
                                                #               (NO LOAD, DEC ONLY)
                OCT     00542                   # 41    2COMP   CDU DEG, ELEV DEG
                OCT     24410                   # 42    3COMP   POS4, POS4, VEL3
                                                #               (DEC ONLY)
                OCT     20204                   # 43    3COMP   DPDEG(360), DPDEG(360) POS4
                                                #               (DEC ONLY)
                OCT     00410                   # 44    3COMP   POS4, POS4, MIN/SEC
                                                #               (NO LOAD, DEC ONLY)
                OCT     10000                   # 45    3COMP   WHOLE, MIN/SEC, DPDEG(360)
                                                #               (NO LOAD, DEC ONLY)
                OCT     00000                   # 46    1COMP   OCTAL ONLY
                OCT     00306                   # 47    2COMP   WEIGHT2 FOR EACH
                                                #               (DEC ONLY)
                OCT     01367                   # 48    2COMP   TRIM DEG2 FOR EACH
```

```
                                                #               (DEC ONLY)
                OCT     00510                   # 49    3COMP   POS4, VEL3, WHOLE
                                                #               (DEC ONLY)
                OCT     0                       # 50    SPARE
                OCT     00204                   # 51    2COMP   DPDEG(360), DPDEG(360)
                                                #               (DEC ONLY)
                OCT     00004                   # 52    1COMP   DPDEG(360)
                OCT     00000                   # 53    SPARE
                OCT     10507                   # 54    3COMP   POS5, VEL3, DPDEG(360)
                                                #               (DEC ONLY)
                OCT     10200                   # 55    3COMP   WHOLE, DPDEG(360), DPDEG(360)
                                                #               (DEC ONLY)
                OCT     00204                   # 56    2COMP   DPDEG(360), DPDEG(360)
                OCT     00010                   # 57    1COMP   POS4
                                                #               (DEC ONLY)
                OCT     24510                   # 58    3COMP   POS4, VEL3, VEL3
                                                #               (DEC ONLY)
                OCT     24512                   # 59    3COMP   VEL3 FOR EACH
                                                #               (DEC ONLY)
                OCT     60512                   # 60    3COMP   VEL3, VEL3, COMP ALT
                                                #               (DEC ONLY)
                OCT     54000                   # 61    3COMP   MIN/SEC, MIN/SEC, POS7
                                                #               (NO LOAD, DEC ONLY)
                OCT     24012                   # 62    3COMP   VEL3, MIN/SEC, VEL3
                                                #               (NO LOAD, DEC ONLY)
                OCT     60512                   # 63    3COMP   VEL3, VEL3, COMP ALT
                                                #               (DEC ONLY)
                OCT     60500                   # 64    3COMP   2INT, VEL3, COMP ALT
                                                #               (NO LOAD, DEC ONLY)
                OCT     00000                   # 65    3COMP   HMS (DEC ONLY)
                OCT     00016                   # 66    2COMP   LANDING RADAR ALT, POSITION
                                                #               (NO LOAD, DEC ONLY)
                OCT     53223                   # 67    3COMP   LANDING RADAR VELX, Y, Z
                OCT     60026                   # 68    3COMP   POS7, MIN/SEC, COMP ALT
                                                #               (NO LOAD, DEC ONLY)
                OCT     00000                   # 69    SPARE
                OCT     0                       # 70    3COMP   OCTAL ONLY FOR EACH
                OCT     0                       # 71    3COMP   OCTAL ONLY FOR EACH
                OCT     00102                   # 72    2COMP   360-CDU DEG, CDU DEG
                OCT     00102                   # 73    2COMP   360-CDU DEG, CDU DEG
                OCT     10200                   # 74    3COMP   MIN/SEC, DPDEG(360), DPDEG(360)
                                                #               (NO LOAD, DEC ONLY)
                OCT     00010                   # 75    3COMP   POS4, MIN/SEC, MIN/SEC
                                                #               (NO LOAD, DEC ONLY)

                OCT     20512                   # 76    3COMP   VEL3, VEL3, POS4
                                                #               (DEC ONLY)
                OCT     00500                   # 77    2COMP   MIN/SEC, VEL3
                                                #               (NO LOAD, DEC ONLY)
                OCT     00654                   # 78    2 COMP  RR RANGE, RR RANGE RATE
                OCT     00102                   # 79    3COMP   CDU DEG, CDU DEG, WHOLE
```

```
                                        #               (DEC ONLY)
            OCT     00200               # 80    2COMP   WHOLE, DPDEG(360)
            OCT     24512               # 81    3COMP   VEL3 FOR EACH
                                        #               (DEC ONLY)
            OCT     24512               # 82    3COMP   VEL3 FOR EACH
                                        #               (DEC ONLY)
            OCT     24512               # 83    3COMP   VEL3 FOR EACH
                                        #               (DEC ONLY)
            OCT     24512               # 84    3COMP   VEL3 FOR EACH
                                        #               (DEC ONLY)
            OCT     24512               # 85    3COMP   VEL3 FOR EACH
                                        #               (DEC ONLY)
            OCT     24512               # 86    3COMP   VEL3 FOR EACH
                                        #               (DEC ONLY)
            OCT     00102               # 87    2COMP   CDU DEG FOR EACH
            OCT     0                   # 88    3COMP   FRAC FOR EACH
                                        #               (DEC ONLY)
            OCT     16143               # 89    3COMP   DPDEG(90), DPDEG(90), POS5
                                        #               (DEC ONLY)
            OCT     10507               # 90    3COMP   POS5, VEL3, DEPDEG(360)
                                        #               (DEC ONLY)
            OCT     10450               # 91    3COMP   POS4, VEL2, DPDEG(360)
            OCT     00000               # 92    SPARE
            OCT     06143               # 93    3COMP   DPDEG(90) FOR EACH
            OCT     00000               # 94    SPARE
            OCT     0                   # 95    SPARE
            OCT     0                   # 96    SPARE
            OCT     00000               # 97    3COMP   WHOLE FOR EACH
            OCT     00000               # 98    3COMP   WHOLE, FRAC, WHOLE
            OCT     71572               # 99    3COMP   POS9, VEL4, RADIANS
                                        #               (DEC ONLY)

# END OF NNTYPTAB FOR MIXED NOUNS

SFINTAB     OCT     00006               # WHOLE, DP TIME (SEC)
            OCT     03240
            OCT     00000               # SPARE
            OCT     00000
            OCT     00000               # CDU DEGREES, 360-CDU DEGREES
            OCT     00000               #       (SFCONS IN DEGINSF)
            OCT     10707               # DP DEGREES (90)
            OCT     03435               #       UPPED BY 1
            OCT     13070               # DP DEGREES (360) (POINT BETWN BITS 11-12)
            OCT     34345               #       UPPED BY 1
            OCT     00005               # DEGREES (180)
            OCT     21616
            OCT     26113               # WEIGHT2
            OCT     31713
            OCT     00070               # POSITION5
            OCT     20460
```

```
                OCT     01065                   # POSITION4
                OCT     05740
                OCT     11414                   # VELOCITY2     (POINT BETWN BITS 11-12)
                OCT     31463
                OCT     07475                   # VELOCITY3
                OCT     16051
                OCT     00001                   # ELEVATION DEGREES
                OCT     03434
                OCT     00047                   # RENDEZVOUS RADAR RANGE
                OCT     21135
                OCT     77766                   # RENDESVOUS RADAR RANGE RATE
                OCT     50711
                2DEC*   .9267840599 E5 B-28*    # LANDING RADAR ALTITUDE

                OCT     00002                   # INITIAL/FINAL ALTITUDE
                OCT     23224
                OCT     00014                   # ALTITUDE RATE
                OCT     06500
                OCT     00012                   # FORWARD/LATERAL VELOCITY
                OCT     36455
                OCT     04256                   # ROT HAND CONT ANGLE RATE
                OCT     07071
                2DEC*   -1.552795030 E5 B-28*   # LANDING RADAR VELX

                2DEC*   .8250825087 E5 B-28*    # LANDING RADAR VELY

                2DEC*   1.153668673 E5 B-28*    # LANDING RADAR VELZ

                OCT     04324                   # POSITION7
                OCT     27600
                OCT     00036                   # TRIM DEGREES2
                OCT     20440
                OCT     00035                   # COMPUTED ALTITUDE
                OCT     30400
                OCT     23420                   # DP DEGREES
                OCT     00000
                2DEC    30480 B-19              # POSITION 9

                2DEC    30.48 B-7               # VELOCITY4

                2DEC    100 B-8                 # RADIANS

                                                # END OF SFINTAB

SFOUTAB         OCT     05174                   # WHOLE, DP TIME (SEC)
                OCT     13261
                OCT     00000                   # SPARE
                OCT     00000
                OCT     00000                   # CDU DEGREES, 360-CDU DEGREES
```

```
                OCT     00000           #       (SFCONS IN DEGOUTSF, 360 CDUO)
                OCT     00714           # DP DEGREES (90) (POINT BETWN BITS 7-8)
                OCT     31463
                OCT     13412           # DP DEGREES (360)
                OCT     07534
                OCT     05605           # DEGREES (180)
                OCT     03656
                OCT     00001           # WEIGHT2
                OCT     16170
                OCT     00441           # POSITION5
                OCT     34306
                OCT     07176           # POSITION4     (POINT BETWN BITS 7-8)
                OCT     21603
                OCT     15340           # VELOCITY2
                OCT     15340
                OCT     01031           # VELOCITY3     (POINT BETWN BITS 7-8)
                OCT     21032
                OCT     34631           # ELEVATION DETREES
                OCT     23146
                OCT     00636           # RENDEZVOUS RADAR RANGE
                OCT     14552
                OCT     74552           # RENDEZVOUS RADAR RANGE RATE
                OCT     70307
                2DEC    1.079 E-5 B14   # LANDING RADAR ALTITUDE

                OCT     14226           # INITIAL/FINAL ALTITUDE
                OCT     31757
                OCT     02476           # ALTITUDE RATE
                OCT     05531
                OCT     02727           # FORWARD/LATERAL VELOCTY
                OCT     16415
                OCT     00007           # ROT HAND CONT ANGLE RATE
                OCT     13734
                2DEC    -.6440 E-5 B14  # LANDING RADAR VELX

                2DEC    1.212 E-5 B14   # LANDING RADAR VELY

                2DEC    .8668 E-5 B14   # LANDING RADAR VELZ

                OCT     34772           # POSITION7
                OCT     07016
                OCT     01030           # TRIM DEGREES2
                OCT     33675
                OCT     01046           # COMPUTED ALTITUDE
                OCT     15700
                OCT     00321           # DP DEGREES
                OCT     26706
                2DEC    17.2010499 B-7  # POSITION 9

                2DEC    .032808399      # VELOCITY4
```

```
                2DEC    .32                     # RADIANS

                                                # END OF SFOUTAB

                                                # NN    SF CONSTANT             SF ROUTINE
IDADDTAB        ECADR   TTOGO                   # 40    MIN/SEC                 M/S
                ECADR   VGDISP                  # 40    VEL3                    DP3
                ECADR   DVTOTAL                 # 40    VEL3                    DP3
                ECADR   DSPTEM1                 # 41    CDU DEG                 CDU
                ECADR   DSPTEM1 +1              # 41    ELEV DEG                ARTH
                OCT     0                       # 41    SPARE COMPONENT
                ECADR   HAPO                    # 42    POS4                    DP3
                ECADR   HPER                    # 42    POS4                    DP3
                ECADR   VGDISP                  # 42    VEL3                    DP3
                ECADR   LAT                     # 43    DPDEG(360)              DP4
                ECADR   LONG                    # 43    DPDEG(360               DP4
                ECADR   ALT                     # 43    POS4                    DP3
                ECADR   HAPOX                   # 44    POS4                    DP3
                ECADR   HPERX                   # 44    POS4                    DP3
                ECADR   TFF                     # 44    MIN/SEC                 M/S
                ECADR   TRKMKCNT                # 45    WHOLE                   ARTH
                ECADR   TTOGO                   # 45    MIN/SEC                 M/S
                ECADR   +MGA                    # 45    DPDEG(360)              DP4
                ECADR   DAPDATR1                # 46    OCTAL ONLY              OCT
                OCT     0                       # 46    SPARE COMPONENT
                OCT     0                       # 46    SPARE COMPONENT
                ECADR   LEMMASS                 # 47    WEIGHT2                 ARTH1
                ECADR   CSMMASS                 # 47    WEIGHT2                 ARTH1
                OCT     0                       # 47    SPARE COMPONENT
                ECADR   PITTIME                 # 48    TRIM DEG2               ARTH
                ECADR   ROLLTIME                # 48    TRIM DEG2               ARTH
                OCT     0                       # 48    SPARE COMPONENT
                ECADR   R22DISP                 # 49    POS4                    DP3
                ECADR   R22DISP +2              # 49    VEL3                    DP3
                ECADR   WHCHREAD                # 49    WHOLE                   ARTH
                OCT     0                       # 50    SPARE
                OCT     0                       # 50    SPARE
                OCT     0                       # 50    SPARE
                ECADR   ALPHASB                 # 51    DPDEG(360)              DP4
                ECADR   BETASB                  # 51    DPDEG(360)              DP4
                OCT     0                       # 51    SPARE COMPONENT
                ECADR   ACTCENT                 # 52    DPDEG(360)              DP4
                OCT     00000                   # 52    SPARE COMPONENT
                OCT     00000                   # 52    SPARE COMPONENT
                OCT     00000                   # 53    SPARE
                OCT     00000                   # 53
                OCT     00000                   # 53
                ECADR   RANGE                   # 54    POS5                    DP1
```

```
                ECADR   RRATE                   # 54    VEL3                    DP3
                ECADR   RTHETA                  # 54    DPDEG(360)              DP4
                ECADR   NN                      # 55    WHOLE                   ARTH
                ECADR   ELEV                    # 55    DPDEG(360)              DP4
                ECADR   CENTANG                 # 55    DPDEG(360)              DP4
                ECADR   RR-AZ                   # 56    DPDEG(360)              DP4
                ECADR   RR-ELEV                 # 56    DPDEG(360)              DP4
                OCT     0                       # 56    SPARE COMPONENT
                ECADR   DELTAR                  # 57    POS4                    DP3
                OCT     0                       # 57    SPARE COMPONENT
                OCT     0                       # 57    SPARE COMPONENT
                ECADR   POSTTPI                 # 58    POS4                    DP3
                ECADR   DELVTPI                 # 58    VEL3                    DP3
                ECADR   DELVTPF                 # 58    VEL3                    DP3
                ECADR   DVLOS                   # 59    VEL3                    DP3
                ECADR   DVLOS +2                # 59    VEL3                    DP3
                ECADR   DVLOS +4                # 59    VEL3                    DP3
                ECADR   VHORIZ                  # 60    VEL3                    DP3
                ECADR   HDOTDISP                # 60    VEL3                    DP3
                ECADR   HCALC                   # 60    COMP ALT                DP1
                ECADR   TTFDISP                 # 61    MIN/SEC                 M/S
                ECADR   TTOGO                   # 61    MIN/SEC                 M/S
                ECADR   OUTOFPLN                # 61    POS7                    DP4
                ECADR   ABVEL                   # 62    VEL3                    DP3
                ECADR   TTOGO                   # 62    MIN/SEC                 M/S
                ECADR   DVTOTAL                 # 62    VEL3                    DP3
                ECADR   ABVEL                   # 63    VEL3                    DP3
                ECADR   HDOTDISP                # 63    VEL3                    DP3
                ECADR   HCALC1                  # 63    COMP ALT                DP1
                ECADR   FUNNYDSP                # 64    2INT                    2INT
                ECADR   HDOTDISP                # 64    VEL3                    DP3
                ECADR   HCALC                   # 64    COMP ALT                DP1
                ECADR   SAMPTIME                # 65    HMS (MIXED ONLY TO KEEP CODE 65) HMS
                ECADR   SAMPTIME                # 65    HMS                     HMS
                ECADR   SAMPTIME                # 65    HMS                     HMS
                ECADR   RSTACK +6               # 66    LANDING RADAR ALT       DP1
                OCT     0                       # 66    LR POSITION             LRPOS
                OCT     0                       # 66    SPARE COMPONENT
                ECADR   RSTACK                  # 67    LANDING RADAR VELX      DP1
                ECADR   RSTACK +2               # 67    LANDING RADAR VELY      DP1
                ECADR   RSTACK +4               # 67    LANDING RADAR VELZ      DP1
                ECADR   RANGEDSP                # 68    POS7                    DP4
                ECADR   TTFDISP                 # 68    MIN/SEC                 M/S
                ECADR   DELTAH                  # 68    COMP ALT                DP1
                OCT     00000                   # 69    SPARE
                OCT     00000                   # 69
                OCT     00000                   # 69
                ECADR   AOTCODE                 # 70    OCTAL ONLY              OCT
                ECADR   AOTCODE +1              # 70    OCTAL ONLY              OCT
                ECADR   AOTCODE +2              # 70    OCTAL ONLY              OCT
```

```
                ECADR   AOTCODE                 # 71    OCTAL ONLY              OCT
                ECADR   AOTCODE +1              # 71    OCTAL ONLY              OCT
                ECADR   AOTCODE +2              # 71    OCTAL ONLY              OCT
                ECADR   CDUT                    # 72    360-CDU DEG             360-CDU
                ECADR   CDUS                    # 72    CDU DEG                 CDU
                OCT     0                       # 72    SPARE COMPONENT
                ECADR   TANG                    # 73    360-CDU DEG             360-CDU
                ECADR   TANG +1                 # 73    CDU DEG                 CDU
                OCT     0                       # 73    SPARE COMPONENT
                ECADR   TTOGO                   # 74    MIN/SEC                 M/S
                ECADR   YAW                     # 74    DPDEG(360)              DP4
                ECADR   PITCH                   # 74    DPDEG(360)              DP4
                ECADR   DIFFALT                 # 75    POS4                    DP3
                ECADR   T1TOT2                  # 75    MIN/SEC
                ECADR   T2TOT3                  # 75    MIN/SEC                 M/S
                ECADR   ZDOTD                   # 76    VEL3                    DP3
                ECADR   RDOTD                   # 76    VEL3                    DP3
                ECADR   XRANGE                  # 76    POS4                    DP3
                ECADR   TTOGO                   # 77    MIN/SEC                 M/S
                ECADR   YDOT                    # 77    VEL3                    DP3
                OCT     0                       # 77    SPARE COMPONENT
                ECADR   RSTACK                  # 78    RR RANGE                DP1
                ECADR   RSTACK +2               # 78    RR RANGE RATE           DP1
                OCT     00000                   # 78    SPARE COMPONENT
                ECADR   CURSOR                  # 79    CDU DEG                 CDU
                ECADR   SPIRAL                  # 79    CDU DEG                 CDU
                ECADR   POSCODE                 # 79    WHOLE                   ARTH
                ECADR   DATAGOOD                # 80    WHOLE                   ARTH
                ECADR   OMEGAD                  # 80    DPDEG(360)              DP4
                OCT     0                       # 80    SPARE COMPONENT
                ECADR   DELVLVC                 # 81    VEL3                    DP3
                ECADR   DELVLVC +2              # 81    VEL3                    DP3
                ECADR   DELVLVC +4              # 81    VEL3                    DP3
                ECADR   DELVLVC                 # 82    VEL3                    DP3
                ECADR   DELVLVC +2              # 82    VEL3                    DP3
                ECADR   DELVLVC +4              # 82    VEL3                    DP3
                ECADR   DELVIMU                 # 83    VEL3                    DP3
                ECADR   DELVIMU +2              # 83    VEL3                    DP3
                ECADR   DELVIMU +4              # 83    VEL3                    DP3
                ECADR   DELVOV                  # 84    VEL3                    DP3
                ECADR   DELVOV +2               # 84    VEL3                    DP3
                ECADR   DELVOV +4               # 84    VEL3                    DP3
                ECADR   VGBODY                  # 85    VEL3                    DP3
                ECADR   VGBODY +2               # 85    VEL3                    DP3
                ECADR   VGBODY +4               # 85    VEL3                    DP3
                ECADR   DELVLVC                 # 86    VEL3                    DP3
                ECADR   DELVLVC +2              # 86    VEL3                    DP3
                ECADR   DELVLVC +4              # 86    VEL3                    DP3
                ECADR   AZ                      # 87    CDU DEG                 CDU
                ECADR   EL                      # 87    CDU DEG                 CDU
```

```
                OCT     0                       # 87    SPARE COMPONENT
                ECADR   STARAD                  # 88    FRAC                    FRAC
                ECADR   STARAD +2               # 88    FRAC                    FRAC
                ECADR   STARAD +4               # 88    FRAC                    FRAC
                ECADR   LANDLAT                 # 89    DPDEG(90)               DP3
                ECADR   LANDLONG                # 89    DPDEG(90)               DP3
                ECADR   LANDALT                 # 89    POS5                    DP1
                ECADR   RANGE                   # 90    POS5                    DP1
                ECADR   RRATE                   # 90    VEL3                    DP3
                ECADR   RTHETA                  # 90    DPDEG(360)              DP4
                ECADR   P21ALT                  # 91    POS4                    DP3
                ECADR   P21VEL                  # 91    VEL2                    DP4
                ECADR   P21GAM                  # 91    DPDEG(360)              DP4
                OCT     00000                   # 92    SPARE
                OCT     00000                   # 92
                OCT     00000                   # 92
                ECADR   OGC                     # 93    DPDEG(90)               DP3
                ECADR   OGC +2                  # 93    DPDEG(90)               DP3
                ECADR   OGC +4                  # 93    DPDEG(90)               DP3
                OCT     00000                   # 94    SPARE
                OCT     00000                   # 94
                OCT     00000                   # 94
                OCT     0                       # 95    SPARE
                OCT     0                       # 95    SPARE
                OCT     0                       # 95    SPARE
                OCT     0                       # 96    SPARE
                OCT     0                       # 96    SPARE
                OCT     0                       # 96    SPARE
                ECADR   DSPTEM1                 # 97    WHOLE                   ARTH
                ECADR   DSPTEM1 +1              # 97    WHOLE                   ARTH
                ECADR   DSPTEM1 +2              # 97    WHOLE                   ARTH
                ECADR   DSPTEM2                 # 98    WHOLE                   ARTH
                ECADR   DSPTEM2 +1              # 98    FRAC                    FRAC
                ECADR   DSPTEM2 +2              # 98    WHOLE                   ARTH
                ECADR   WWPOS                   # 99    POS9                    DP3
                ECADR   WWVEL                   # 99    VEL4                    DP2
                ECADR   WWBIAS                  # 99    RADIANS                 DP4

# END OF IDADDTAB

                                                # NN    SF ROUTINES
RUTMXTAB        OCT     16351                   # 40    M/S, DP3, DP3
                OCT     00142                   # 41    CDU, ARTH
                OCT     16347                   # 42    DP3, DP3, DP3
                OCT     16512                   # 43    DP4, DP4, DP3
                OCT     22347                   # 44    DP3, DP3, M/S
                OCT     24443                   # 45    ARTH, M/S, DP4
                OCT     00000                   # 46    OCT
                OCT     00553                   # 47    ARITH1, ARITH1
```

```
                OCT     00143                   # 48    ARTH, ARTH
                OCT     06347                   # 49    DP3, DP3, ARTH
                OCT     0                       # 50    SPARE
                OCT     00512                   # 51    DP4, DP4
                OCT     00012                   # 52    DP4
                OCT     00000                   # 53    SPARE
                OCT     24344                   # 54    DP1, DP3, DP4
                OCT     24503                   # 55    ARTH, DP4, DP4
                OCT     00512                   # 56    DP4, DP4
                OCT     00007                   # 57    DP3
                OCT     16347                   # 58    DP3, DP3, DP3
                OCT     16347                   # 59    DP3, DP3, DP3
                OCT     10347                   # 60    DP3, DP3, DP1
                OCT     24451                   # 61    M/S, M/S, DP4
                OCT     16447                   # 62    DP3, M/S, DP3
                OCT     10347                   # 63    DP3, DP3, DP1
                OCT     10354                   # 64    2INT, DP3, DP1
                OCT     20410                   # 65    HMS, HMS, HMS
                OCT     00304                   # 66    DP1, LRPOS
                OCT     10204                   # 67    DP1, DP1, DP1
                OCT     10452                   # 68    DP4, M/S, DP1
                OCT     00000                   # 69    SPARE
                OCT     0                       # 70    OCT, OCT, OCT
                OCT     0                       # 71    OCT, OCT, OCT
                OCT     00115                   # 72    360-CDU, CDU
                OCT     00115                   # 73    360-CDU, CDU
                OCT     24511                   # 74    M/S, DP4, DP4
                OCT     22447                   # 75    DP3, M/S, M/S
                OCT     16347                   # 76    DP3, DP3, DP3
                OCT     00351                   # 77    M/S, DP3
                OCT     00204                   # 78    DP1, DP1
                OCT     06102                   # 79    CDU, CDU, ARTH
                OCT     00503                   # 80    ARTH, DP4
                OCT     16347                   # 81    DP3, DP3, DP3
                OCT     16347                   # 82    DP3, DP3, DP3
                OCT     16347                   # 83    DP3, DP3, DP3
                OCT     16347                   # 84    DP3, DP3, DP3
                OCT     16347                   # 85    DP3, DP3, DP3
                OCT     16347                   # 86    DP3, DP3, DP3
                OCT     00102                   # 87    CDU, CDU
                OCT     02041                   # 88    FRAC FOR EACH
                OCT     10347                   # 89    DP3, DP3, DP1
                OCT     24344                   # 90    DP1, DP3, DP4
                OCT     24507                   # 91    DP3, DP4, DP4
                OCT     00000                   # 92    SPARE
                OCT     16347                   # 93    DP3, DP3, DP3
                OCT     00000                   # 94    SPARE
                OCT     0                       # 95    SPARE
                OCT     0                       # 96    SPARE
                OCT     06143                   # 97    ARTH, ARTH, ARTH
```

```
                OCT     06043                   # 98    ARTH, FRAC, ARTH
                OCT     24247                   # 99    DP3, DP2, DP4

# END OF RUTMXTAB

                SBANK=  LOWSUPER
```

```
                BANK    23
                SETLOC  LEMGEOM
                BANK

                SBANK=  LOWSUPER
                EBANK=  XSM

# THESE TWO ROUTINES COMPUTE THE ACTUAL STATE VECTOR FOR LM,CSM BY ADDING
# THE CONIC R,V AND THE DEVIATIONSR,V. THE STATE VECTORS ARE CONVERTED TO
# METERS B-29 AND METERS/CSEC B-7 AND STORED APPROPRIATELY IN RN,VN OR
# R-OTHER , V-OTHER FOR DOWNLINK. THE ROUTINES NAMES ARE SWITCHED IN THE
# OTHER VEHICLES COMPUTER.
#
# INPUT
#   STATE VECTOR IN TEMPORARY STORAGE AREA
#   IF STATE VECTOR IS SCALED POS B27 AND VEL B5
#      SET X2 TO +2
#   IF STATE VECTOR IS SCALED POS B29 AND VEL B7
#      SET X2 TO 0
#
# OUTPUT
#   R(T) IN RN, V(T) IN VN, T IN PIPTIME
# OR
#   R(T) IN R-OTHER, V(T) IN V-OTHER   (T IS DEFINED BY T-OTHER)

                COUNT*  $$/GEOM
SVDWN2          BOF     RVQ             # SW=1=AVETOMID DOING W-MATRIX INTEG.
                        AVEMIDSW
                        +1
                VLOAD   VSL*
                        TDELTAV
                        0 -7,2
                VAD     VSL*
                        RCV
                        0,2
                STOVL   RN
                        TNUV
                VSL*    VAD
                        0 -4,2
                        VCV
                VSL*
                        0,2
                STODL   VN
                        TET
                STORE   PIPTIME
                RVQ
```

```
SVDWN1          VLOAD   VSL*
                        TDELTAV
                        0 -7,2
                VAD     VSL*
                        RCV
                        0,2
                STOVL   R-OTHER
                        TNUV
                VSL*    VAD
                        0 -4,2
                        VCV
                VSL*
                        0,2
                STORE   V-OTHER
                RVQ
```

```
#          THE FOLLOWING ROUTINE TAKES A HALF UNIT TARGET VECTOR REFERRED TO NAV BASE COORDINATES AND FINDS BOTH
# GIMBAL ORIENTATIONS AT WHICH THE RR MIGHT SIGHT THE TARGET. THE GIMBAL ANGLES CORRESPONDING TO THE PRESENT MODE
# ARE LEFT IN MODEA AND THOSE WHICH WOULD BE USED AFTER A REMODE IN MODEB. THIS ROUTINE ASSUMES MODE 1 IS TRUNNION
# ANGLE LESS THAN 90 DEGS IN ABS VALUE WITH ARBITRARY SHAFT, WITH A CORRESPONDING DEFINITION FOR MODE 2. MODE
# SELECTION AND LIMIT CHECKING ARE DONE ELSEWHERE.
#
#          THE MODE 1 CONFIGURATION IS CALCULATED FROM THE VECTOR AND THEN MODE 2 IS FOUND USING THE RELATIONS
#
#          S(2) = 180 + S(1)
#          T(2) = 180 - T(1)
#
#     THE VECTOR ARRIVES IN MPAC WHERE TRG*SMNB OR *SMNB* WILL HAVE LEFT IT.

RRANGLES        STORE   32D
                DLOAD   DCOMP           # SINCE WE WILL FIND THE MODE 1 SHAFT
                        34D             # ANGLE LATER, WE CAN FIND THE MODE 1
                SETPD   ASIN            # TRUNNION BY SIMPLY TAKING THE ARCSIN OF
                        0               # THE Y COMPONENT, THE ASIN GIVIN AN
                PUSH    BDSU            # ANSWER WHOSE ABS VAL IS LESS THAN 90 DEG
                        LODPHALF
                STODL   4               # MODE 2 TRUNNION TO 4.

                        LO6ZEROS
                STOVL   34D             # UNIT THE PROJECTION OF THE VECTOR
                        32D             #   IN THE X-Z PLANE
                UNIT    BOVB            # IF OVERFLOW,TARGET VECTOR IS ALONG Y
                        LUNDESCH        # CALL FOR MANEUVER UNLESS ON LUNAR SURF
                STODL   32D             # PROJECTION VECTOR.
                        32D
                SR1     STQ
                        S2
                STODL   SINTH           # USE ARCTRIG SINCE SHAFT COULD BE ARB.
                        36D
                SR1
                STCALL  COSTH
                        ARCTRIG
```

```
                PUSH    DAD             # MODE 1 SHAFT TO 2.
                        LODPHALF
                STOVL   6
                        4
                RTB                     # FIND MODE 2 CDU ANGLES.
                        2V1STO2S
                STOVL   MODEB
                        0
                RTB                     # MODE 1 ANGLES TO MODE A.
                        2V1STO2S
                STORE   MODEA
                EXIT

                CS      RADMODES        # SWAP MODEA AND MODEB IF RR IN MODE 2.
                MASK    ANTENBIT
                CCS     A
                TCF     +4

                DXCH    MODEA
                DXCH    MODEB
                DXCH    MODEA

                TC      INTPRET
                GOTO
                        S2
```

```
# GIVEN RR TRUNNION AND SHAFT (T,S) IN TANGNB,+1,FIND THE ASSOCIATED
# LINE OF SIGHT IN NAV BASE AXES.  THE HALF UNIT VECTOR, .5(SIN(S)COS(T),
# -SIN(T),COS(S)COS(T)) IS LEFT IN MPAC AND 32D.

                SETLOC  INFLIGHT
                BANK

                COUNT*  $$/GEOM

RRNB            SLOAD   RTB
                        TANGNB
                        CDULOGIC
                SETPD   PUSH            # TRUNNION ANGLE TO 0
                        0
                SIN     DCOMP
                STODL   34D             # Y COMPONENT

                COS     PUSH            # .5 COS(T) TO 0
                SLOAD   RTB
                        TANGNB +1
                        CDULOGIC
RRNB1           PUSH    COS             # SHAFT ANGLE TO 2
                DMP     SL1
                        0
                STODL   36D             # Z COMPONENT

                SIN     DMP
                SL1
                STOVL   32D
                        32D
                RVQ

# THIS ENTRY TO RRNB REQUIRES THE TRUNNION AND SHAFT ANGLES IN MPAC AND MPAC +1 RESPECTIVELY

RRNBMPAC        STODL   20D             # SAVE SHAFT CDU IN 21.
                        MPAC            # SET MODE TO DP.  (THE PRECEEDING STORE
                                        # MAY BE DP. TP OR VECTOR.)
                RTB     SETPD
                        CDULOGIC
                        0
                PUSH    SIN             # TRUNNION ANGLE TO 0
                DCOMP
                STODL   34D             # Y COMPONENT
                COS     PUSH            # .5COS(T) TO 0
                SLOAD   RTB             # PICK UP CDU'S.
                        21D
                        CDULOGIC
                GOTO
                        RRNB1
```

```
                BANK    7
                SETLOC  IMUCOMP
                BANK

                EBANK=  NBDX

                COUNT*  $$/ICOMP
1/PIPA          CAF     LGCOMP          # SAVE EBANK OF CALLING PROGRAM
                XCH     EBANK
                TS      MODE

                CCS     GCOMPSW         # BYPASS IF GCOMPSW NEGATIVE
                TCF     +3
                TCF     +2
                TCF     IRIG1           # RETURN

1/PIPA1         CAF     FOUR            # PIPAZ, PIPAY, PIPAX
                TS      BUF +2

                INDEX   BUF +2
                CA      PIPASCF         # (P.P.M.) X 2(-9)
                EXTEND
                INDEX   BUF +2
                MP      DELVX           # (PP) X 2(+14) NOW (PIPA PULSES) X 2(+5)
                TS      Q               # SAVE MAJOR PART

                CA      L               # MINOR PART
                EXTEND
                MP      BIT6            # SCALE 2(+9)   SHIFT RIGHT 9
                INDEX   BUF +2
                TS      DELVX +1        # FRACTIONAL PIPA PULSES SCALED 2(+14)

                CA      Q               # MAJOR PART
                EXTEND
                MP      BIT6            # SCALE 2(+9)   SHIFT RIGHT 9
                INDEX   BUF  +2
                DAS     DELVX           # (PIPAI) + (PIPAI)(SFE)

                INDEX   BUF +2
                CS      PIPABIAS        # (PIPA PULSES)/(CS) X 2(-5)             *
                EXTEND
                MP      1/PIPADT        # (CS) X 2(+8)  NOW (PIPA PULSES) X 2(+3)*
                EXTEND
                MP      BIT4            # SCALE 2(+11)  SHIFT RIGHT 11           *
                INDEX   BUF +2
                DAS     DELVX           # (PIPAI) + (PIPAI)(SFE) - (BIAS)(DELTAT)

                CCS     BUF     +2      # PIPAZ, PIPAY, PIPAX
                AD      NEG1
                TCF     1/PIPA1 +1
```

```
        NOOP                    # LESS THAN ZERO IMPOSSIBLE
```

```
IRIGCOMP        TS      GCOMPSW         # INDICATE COMMANDS 2 PULSES OR LESS.
                TS      BUF             # INDEX COUNTER .  IRIGX, IRIGY, IRIGZ.

                TC      IRIGX           # COMPENSATE ACCELERATION TERMS

                CS      NBDX            # (GYRO PULSES)/(CS) X 2(-5)
                TC      DRIFTSUB        # -(NBOX)(DELTAT)   (GYRO PULSES) X 2(+14)

                TC      IRIGY           # COMPENSATE ACCELERATION TERMS

                CS      NBDY            # (GYRO PULSES)/(CS) X 2(-5)
                TC      DRIFTSUB        # -(NBDY)(DELTAT)   (GYRO PULSES) X 2(+14)

                TC      IRIGZ           # COMPENSATE ACCELERATION TERMS

                CA      NBDZ            # (GYRO PULSES)/(CS) X 2(-5)
                TC      DRIFTSUB        # +(NBDZ)(DELTAT)   (GYRO PULSES) X 2(+14)

                CCS     GCOMPSW         # ARE GYRO COMMANDS GREATER THAN 2 PULSES
                TCF     +2              # YES  SEND OUT GYRO TORQUING COMMANDS.
                TCF     IRIG1           # NO  RETURN

                CA      PRIO21          # PRIO GREATER THAN SERVICER
                TC      NOVAC           # SEND OUT GYRO TORQUING COMMANDS.
                EBANK=  NBDX
                2CADR   1/GYRO

                RELINT
IRIG1           CA      MODE            # RESTORE CALLERS EBANK
                TS      EBANK
                TCF     SWRETURN
```

```
IRIGX           EXTEND
                QXCH    MPAC +2         # SAVE Q
                EXTEND
                DCS     DELVX           # (PIPA PULSES) X 2(+14)
                DXCH    MPAC
                CA      ADIAX           # (GYRO PULSES)/(PIPA PULSE) X 2(-6)     *
                TC      GCOMPSUB        # -(ADIAX)(PIPAX)   (GYRO PULSES) X 2(+14)

                EXTEND                  #
                DCS     DELVY           # (PIPA PULSES) X 2(+14)
                DXCH    MPAC            #
                CS      ADSRAX          # (GYRO PULSES)/(PIPA PULSE) X 2(-6)     *
                TC      GCOMPSUB        # +(ADSRAX)(PIPAY)  (GYRO PULSES) X 2(+14)

#               EXTEND          # ***
#               DCS     DELVZ   # ***    (PIPA PULSES) X 2(+14)
#               DXCH    MPAC    # ***
#               CA      ADOAX   # ***    (GYRO PULSES)/(PIPA PULSE) X 2(-6)     *
#               TC      GCOMPSUB # ***   -(ADOAX)(PIPAZ)   (GYRO PULSES) X 2(+14)

                TC      MPAC  +2

IRIGY           EXTEND
                QXCH    MPAC  +2        # SAVE Q
                EXTEND
                DCS     DELVY           # (PIPA PULSES) X 2(+14)
                DXCH    MPAC
                CA      ADIAY           # (GYRO PULSES)/(PIPA PULSE) X 2(-6)     *
                TC      GCOMPSUB        # -(ADIAY)(PIPAY)   (GYRO PULSES) X 2(+14)

                EXTEND
                DCS     DELVZ           # (PIPA PULSES) X 2(+14)
                DXCH    MPAC
                CS      ADSRAY          # (GYRO PULSES)/(PIPA PULSE) X 2(-6)     *
                TC      GCOMPSUB        # +(ADSRAY)(PIPAZ)  (GYRO PULSES) X 2(+14)

#               EXTEND          # ***
#               DCS     DELVX   # ***    (PIPA PULSES) X 2(+14)
#               DXCH    MPAC    # ***
#               CA      ADOAY   # ***    (GYRO PULSES)/(PIPA PULSE) X 2(-6)     *
#               TC      GCOMPSUB # ***   -(ADOAY)(PIPAX)   (GYRO PULSES) X 2(+14)

                TC      MPAC +2

IRIGZ           EXTEND
                QXCH    MPAC +2         # SAVE Q
                EXTEND
                DCS     DELVY           # (PIPA PULSES) X 2(+14)
                DXCH    MPAC
                CA      ADSRAZ          # (GYRO PULSES)/(PIPA PULSE) X 2(-6)     *
```

```
                TC      GCOMPSUB        # -(ADSRAZ)(PIPAY)  (GYRO PULSES) X 2(+14)

                EXTEND
                DCS     DELVZ           # (PIPA PULSES) X 2(+14)
                DXCH    MPAC
                CA      ADIAZ           # (GYRO PULSES)/(PIPA PULSE) X 2(-6)     *
                TC      GCOMPSUB        # -(ADIAZ)(PIPAZ)   (GYRO PULSES) X 2(+14)

#               EXTEND          # ***
#               DCS     DELVX   # ***    (PIPA PULSE) X 2(+14)
#               DXCH    MPAC    # ***
#               CS      ADOAZ   # ***    (GYRO PULSES)/(PIPA PULSE) X 2(-6)     *
#               TC      GCOMPSUB # ***   +(ADOAZ)(PIPAX)   (GYRO PULSES) X 2(+14)

                TC      MPAC +2
```

```
GCOMPSUB        XCH     MPAC            # ADIA OR ADSRA COEFFICIENT ARRIVES IN A
                EXTEND                  # C(MPAC) = (PIPA PULSES) X 2(+14)
                MP      MPAC            # (GYRO PULSES)/(PIPA PULSE) X 2(-6)     *
                DXCH    VBUF            # NOW = (GYRO PULSES) X 2(+8)            *

                CA      MPAC +1         # MINOR PART OF PIPA PULSES
                EXTEND
                MP      MPAC            # ADIA OR ADSRA
                TS      L
                CAF     ZERO
                DAS     VBUF            # NOW = (GYRO PULSES) X 2(+8)            *

                CA      VBUF            # PARTIAL RESULT - MAJOR
                EXTEND
                MP      BIT9            # SCALE 2(+6)      SHIFT RIGHT           *
                INDEX   BUF             # RESULT = (GYRO PULSES) X 2(+14)
                DAS     GCOMP           # HI(ADIA)(PIPAI)  OR  HI(ADSRA)(PIPAI)

                CA      VBUF +1         # PARTIAL RESULT - MINOR
                EXTEND
                MP      BIT9            # SCALE 2(+6)   SHIFT RIGHT 6            *
                TS      L
                CAF     ZERO
                INDEX   BUF             # RESULT = (GYRO PULSES) X 2(+14)
                DAS     GCOMP           # (ADIA)(PIPAI) OR (ADSRA)(PIPAI)

                TC      Q
```

```
DRIFTSUB        EXTEND
                QXCH    BUF +1

                EXTEND                  # C(A) = NBD    (GYRO PULSES)/(CS) X 2(-5)
                MP      1/PIPADT        # (CS) X 2(+8)    NOW (GYRO PULSES) X 2(+3)
                LXCH    MPAC +1         # SAVE FOR FRACTIONAL COMPENSATION
                EXTEND
                MP      BIT4            # SCALE 2(+11)     SHIFT RIGHT 11
                INDEX   BUF
                DAS     GCOMP           # HI(NBD)(DELTAT)   (GYRO PULSES) X 2(+14)

                CA      MPAC +1         # NOW MINOR PART
                EXTEND
                MP      BIT4            # SCALE 2(+11)     SHIFT RIGHT 11
                TS      L
                CAF     ZERO
                INDEX   BUF             # ADD IN FRACTIONAL COMPENSATION
                DAS     GCOMP           # (NBD)(DELTAT)     (GYRO PULSES) X 2(+14)

DRFTSUB2        CAF     TWO             # PIPAX, PIPAY, PIPAZ
                AD      BUF
                XCH     BUF
                INDEX   A
                CCS     GCOMP           # ARE GYRO COMMANDS 1 PULSE OR GREATER
                TCF     +2              # YES
                TC      BUF +1          # NO

                MASK    COMPCHK         # DEC -1
                CCS     A               # ARE GYRO COMMANDS GREATER THAN 2 PULSES
                TS      GCOMPSW         # YES - SET GCOMPSW POSITIVE
                TC      BUF +1          # NO
```

```
1/GYRO          CAF     FOUR            # PIPAZ, PIPAY, PIPAX
                TS      BUF

                INDEX   BUF             # SCALE GYRO COMMANDS FOR IMUPULSE
                CA      GCOMP   +1      # FRACTIONAL PULSES
                EXTEND
                MP      BIT8            # SHIFT RIGHT 7
                INDEX   BUF
                TS      GCOMP   +1      # FRACTIONAL PULSES SCALED

                CAF     ZERO            # SET GCOMP = 0 FOR DAS INSTRUCTION
                INDEX   BUF
                XCH     GCOMP           # GYRO PULSES
                EXTEND
                MP      BIT8            # SHIFT RIGHT 7
                INDEX   BUF
                DAS     GCOMP           # ADD THESE TO FRACTIONAL PULSES ABOVE

                CCS     BUF             # PIPAZ, PIPAY, PIPAX
                AD      NEG1
                TCF     1/GYRO +1
LGCOMP          ECADR   GCOMP           # LESS THAN ZERO IMPOSSIBLE

                CAF     LGCOMP
                TC      BANKCALL
                CADR    IMUPULSE        # CALL GYRO TORQUING ROUTINE
                TC      BANKCALL
                CADR    IMUSTALL        # WAIT FOR PULSES TO GET OUT
                TCF     ENDOFJOB        # TEMPORARY

GCOMP1          CAF     FOUR            # PIPAZ, PIPAY, PIPAX
                TS      BUF

                INDEX   BUF             # RESCALE
                CA      GCOMP +1
                EXTEND
                MP      BIT8            # SHIFT MINOR PART LEFT 7 - MAJOR PART = 0
                INDEX   BUF
                LXCH    GCOMP +1        # BITS 8-14 OF MINOR PART WERE = 0

                CCS     BUF             # PIPAZ, PIPAY, PIPAX
                AD      NEG1
                TCF     GCOMP1 +1
COMPCHK         DEC     -1              # LESS THAN ZERO IMPOSSIBLE
                TCF     ENDOFJOB
```

```
NBDONLY         CCS     GCOMPSW         # BYPASS IF GCOMPSW NEGATIVE
                TCF     +3
                TCF     +2
                TCF     ENDOFJOB

                INHINT
                CCS     FLAGWRD2        # PREREAD T3RUPT MAY COINCIDE
                TCF     ENDOFJOB
                TCF     ENDOFJOB
                TCF     +1

                CA      FLAGWRD8        # IF SURFACE FLAG IS SET, SET TEM1
                MASK    BIT8            # POSITIVE SO THAT THE ACCELERATION TERMS
                TS      TEM1            # WILL BE COMPENSATED.
                EXTEND
                BZF     +3              # ARE WE ON THE SURFACE

                TC      IBNKCALL        # ON THE SURFACE
                CADR    PIPASR +3       # READ PIPAS, BUT DO NOT SCALE THEM

                CA      TIME1           # (CS) X 2(+14)
                XCH     1/PIPADT        # PREVIOUS TIME
                RELINT
                COM
                AD      1/PIPADT        # PRESENT TIME - PREVIOUS TIME
NBD2            AD      HALF            # CORRECT FOR POSSIBLE TIME1 TICK
                AD      HALF
                XCH     L               # IF TIME1 DID NOT TICK, REMOVE RESULTING
                XCH     L               # OVERFLOW.

NBD3            EXTEND                  # C(A) = DELTAT    (CS) X 2(+14)
                MP      BIT10           # SHIFT RIGHT 5
                DXCH    VBUF +2

                CA      ZERO
                TS      GCOMPSW         # INDICATE COMMANDS 2 PULSES OR LESS.
                TS      BUF             # INDEX  X, Y, Z.

                CCS     TEM1            # IF SURFACE FLAG IS SET,
                TC      IRIGX           # COMPENSATE ACCELERATION TERMS.

                EXTEND
                DCA     VBUF +2
                DXCH    MPAC            # DELTAT NOW SCALED (CS) X 2(+19)

                CS      NBDX            # (GYRO PULSES)/(CS) X 2(-5)
                TC      FBIASSUB        # -(NBDX)(DELTAT)   (GYRO PULSES) X 2(+14)

                CCS     TEM1            # IF SURFACE FLAG IS SET,
                TC      IRIGY           # COMPENSATE ACCELERATION TERMS.
```

```
                EXTEND
                DCS     VBUF +2
                DXCH    MPAC            # DELTAT SCALED (CS) X 2(+19)
                CA      NBDY            # (GYRO PULSES)/(CS) X 2(-5)
                TC      FBIASSUB        # -(NBDY)(DELTAT)   (GYRO PULSES) X 2(+14)

                CCS     TEM1            # IF SURFACE FLAG IS SET.
                TC      IRIGZ           # COMPENSATE ACCELERATION TERMS

                EXTEND
                DCS     VBUF +2
                DXCH    MPAC            # DELTAT SCALED (CS) X 2(+19)
                CS      NBDZ            # (GYRO PULSES)/(CS) X 2(-5)
                TC      FBIASSUB        # +(NBDZ)(DELTAT)   (GYRO PULSES) X 2(+14)

                CCS     GCOMPSW         # ARE GYRO COMMANDS GREATER THAN 2 PULSES
                TCF     1/GYRO          # YES
                TCF     ENDOFJOB        # NO
```

```
FBIASSUB        XCH     Q
                TS      BUF +1

                CA      Q                # NBD SCALED (GYRO PULSES)/(CS) X 2(-5)
                EXTEND
                MP      MPAC             # DELTAT SCALED (CS) X 2(+19)
                INDEX   BUF
                DAS     GCOMP            # HI(NBD)(DELTAT)   (GYRO PULSES) X 2(+14)

                CA      Q                # NOW FRACTIONAL PART
                EXTEND
                MP      MPAC +1
                TS      L
                CAF     ZERO
                INDEX   BUF
                DAS     GCOMP            # (NBD)(DELTAT)     (GYRO PULSES) X 2(+14)

                TCF     DRFTSUB2         # CHECK MAGNITUDE OF COMPENSATION

LASTBIAS        TC      BANKCALL
                CADR    PIPUSE1

                CCS     GCOMPSW
                TCF     +3
                TCF     +2
                TCF     ENDOFJOB

                CA      FLAGWRD8         # IF SURFACE FLAG IS SET, SET TEM1
                MASK    SURFFBIT         # POSITIVE SO THAT THE ACCELERATION TERMS
                TS      TEM1             # WILL BE COMPENSATED.

                CAF     PRIO31           # 2 SECONDS SCALED (CS) X 2(+8)
                XCH     1/PIPADT
                COM
                AD      PIPTIME +1
                TCF     NBD2

GCOMPZER        CAF     LGCOMP           # ROUTINE TO ZERO GCOMP BEFORE FIRST
                XCH     EBANK            # CALL TO 1/PIPA
                TS      MODE

                CAF     ZERO
                TS      GCOMPSW
                TS      GCOMP
                TS      GCOMP +1
                TS      GCOMP +2
                TS      GCOMP +3
                TS      GCOMP +4
```

```
                TS      GCOMP +5
                TCF     IRIG1           # RESTORE EBANK AND RETURN
```

```
# SUBROUTINE NAME:      V89CALL
# MOD NO:       0                       DATE:           9 JAN 1968
# MOD BY:       DIGITAL DEVEL GROUP     LOG SECTION:    R63
#
# FUNCTIONAL DESCRIPTION:
#
# CALLED BY VERB 89 ENTER DURING P00.  PRIO 10 USED.  CALCULATES AND
# DISPLAYS FINAL FDAI BALL ANGLES TO POINT LM +X OR +Z AXIS AT CSM.
#
# 1. KEY IN V 89 E ONLY IF IN PROG 00.  IF NOT IN P00, OPERATOR ERROR AND
# EXIT R63, OTHERWISE CONTINUE.
#
# 2. IF IN P00, DO IMU STATUS CHECK ROUTINE (R02BOTH).  IF IMU ON AND ITS
# ORIENTATION KNOWN TO LGC, CONTINUE.
#
# 3. FLASH DISPLAY V 04 N 06.  R2 INDICATES WHICH SPACECRAFT AXIS IS TO
# BE POINTED AT CSM.  INITIAL CHOICE IS PREFERRED (+Z) AXIS (R2=1).
# ASTRONAUT CAN CHANGE TO (+X) AXIS (R2 NOT = 1) BY V 22 E 2 E.  CONTINUE
# AFTER KEYING IN PROCEED.
#
# 4. BOTH VEHICLE STATE VECTORS UPDATED BY CONIC EQS.
#
# 5. HALF MAGNITUDE UNIT LOS VECTOR (IN STABLE MEMBER COORDINATES) AND
# HALF MAGNITUDE UNIT SPACECRAFT AXIS VECTOR (IN BODY COORDINATES)
# PREPARED FOR VECPOINT.
#
# 6. GIMBAL ANGLES FROM VECPOINT TRANSFORMED INTO FDAI BALL ANGLES BY
# BALLANGS.  FLASH DISPLAY V 06 N 18 AND AWAIT RESPONSE.
#
# 7. RECYCLE - RETURN TO STEP 4.
#    TERMINATE - EXIT R63.
#    PROCEED - RESET 3AXISFLG AND CALL R60LEM FOR ATTITUDE MANEUVER.
#
# CALLING SEQUENCE:     V 89 E.
#
# SUBROUTINES CALLED:   CHKPOOH, R02BOTH, GOXDSPF, CSMCONIC, LEMCONIC,
#                       VECPOINT, BALLANGS, R60LEM.
#
# NORMAL EXIT MODES:    TC ENDEXT
#
# ALARMS:       1. OPERATOR ERROR IF NOT IN P00.
#               2. PROGRAM ALARM IF IMU IS OFF.
#               3. PROGRAM ALARM IF IMU ORIENTATION IS UNKNOWN.
#
# OUTPUT:       NONE
#
# ERASABLE INITIALIZATION REQUIRED:  NONE
#
# DEBRIS:       OPTION1, +1, TDEC1, PDINTVSM, SCAXIS, CPHI, CTHETA, CPSI,
```

```
#               3AXISFLG.

                EBANK=  RONE
                BANK    32
                SETLOC  BAWLANGS
                BANK

                COUNT*  $$/R63
V89CALL         TC      BANKCALL        # IMU STATUS CHECK.  RETURNS IF ORIENTATION
                CADR    R02BOTH         # KNOWN.  ALARMS IF NOT.
                CAF     THREE           # ALLOW ASTRONAUT TO SELECT DESIRED
                TS      OPTIONX         # TRACKING ATTITUDE AXIS.
                CAF     ONE
                TS      OPTIONX  +1
                CAF     VB04N12         # V 04 N 12
                TC      BANKCALL
                CADR    GOFLASH
                TC      ENDEXT          # TERMINATE
                TC      +2              # PROCEED
                TC      -5              # DATA IN.  OPTION1+1 = 1 FOR Z AXIS
V89RECL         TC      INTPRET         #                       = 2 FOR X AXIS
                RTB     DAD
                        LOADTIME        # READ PRESENT TIME
                        DP1MIN
                STORE   TSTART82        # SAVE TIME FOR LEMCONIC CALL
                STCALL  TDEC1           # STORE TIME FOR CSMCONIC CALL
                        CSMCONIC        # CSM STATE VECTOR UPDATE
                VLOAD                   # CSMCONIC LEFT R VECTOR IN RATT
                        RATT
                STODL   RONE            # SAVE FOR LINE OF SIGHT (LOS) COMPUTATION
                        TSTART82
                STCALL  TDEC1           # STORE TIME FOR LEMCONIC CALL
                        LEMCONIC        # LEM STATE VECTOR UPDATE
                VLOAD   VSU             # CSM POSITION - LEM POSITION = LOS
                        RONE            # LOS VECTOR LEFT IN MPAC
                        RATT
                MXV     RTB             # (REFSMAT X LOS).  TRANSFORMS LOS FROM
                        REFSMMAT        # REFERENCE COORD TO STAB MEMB COORD.
                        NORMUNIT
                STORE   POINTVSM        # STORE LOS FOR VECPOINT CALCULATION
                EXIT
                CS      OPTIONX +1      # 1 FOR Z AXIS.  2 FOR X AXIS.
                AD      ONE
                EXTEND
                BZF     ALINEZ
ALINEX          TC      INTPRET         # X AXIS ALIGNMENT
                VLOAD
                        UNITX           # READ (.5, 0, 0)
```

```
V89CALL1        STCALL  SCAXIS          # STORE SELECTED ALIGNMENT AXIS
                        VECPOINT        # PUTS DESIRED GIM ANG (OG,IG,MG) IN TMPAC
                STORE   CPHI            # STORE GIMBAL ANGLES FOR BALLANGS CALL.
                EXIT
                TC      BANKCALL
                CADR    BALLANGS        # PUTS DESIRED BALL ANGLE IN FDAIX,Y,Z
                CAF     VB06N18         # V 06 N 18
                TC      BANKCALL        # NOUN 18 REFERS TO FDAIX,Y,Z
                CADR    GOFLASH
                TC      ENDEXT          # TERMINATE
                TC      +2              # PROCEED
                TC      V89RECL         # RECYCLE
                TC      DOWNFLAG        # RESET 3 AXIS FLAG
                ADRES   3AXISFLG        # RESET BIT6 FLAG WORD 5
                TC      BANKCALL        # PERFORMS LEM MANEUVER TO ALIGN SELECTED
                CADR    R60LEM          # SPACECRAFT AXIS TO CSM.
                TCF     ENDEXT          # TERMINATE R63

ALINEZ          TC      INTPRET         # Z AXIS ALIGNMENT
                VLOAD   GOTO
                        UNITZ           # READ (0, 0, .5)
                        V89CALL1

VB04N12         VN      412
VB06N18         VN      0618
```

```
DP1MIN          2DEC    6000
```

```
# BLOCK 2 LGC ATTITUDE MANEUVER ROUTINE -- KALCMANU
#
# MOD 2         DATE 5/1/67     BY DON KEENE
#
# PROGRAM DESCRIPTION
#
# KALCMANU IS A ROUTINE WHICH GENERATES COMMANDS FOR THE LM DAP TO CHANGE THE ATTITUDE OF THE SPACECRAFT
# DURING FREE FALL.  IT IS DESIGNED TO MANEUVER THE SPACECRAFT FROM ITS INITIAL ORIENTATION TO SOME DESIRED
# ORIENTATION SPECIFIED BY THE PROGRAM WHICH CALLS KALCMANU, AVOIDING GIMBAL LOCK IN THE PROCESS.  IN THE
# MOD 2 VERSION, THIS DESIRED ATTITUDE IS SPECIFIED BY A SET OF OF THREE COMMANDED CDU ANGLES STORES AS 2'S COMPLEMENT
# SINGLE PRECISION ANGLES IN THE THREE CONSECUTIVE LOCATIONS, CPHI, CTHETA, CPSI, WHERE
#
#       CPHI = COMMANDED OUTER GIMBAL ANGLE
#       CTHETA = COMMANDED INNER GIMBAL ANGLE
#       CPSI = COMMANDED MIDDLE GIMBAL ANGLE
#
# WHEN POINTING A SPACECRAFT AXIS (I.E., X, Y, Z, THE AOT, THRUST AXIS, ETC.) THE SUBROUTINE VECPOINT MAY BE
# USED TO GENERATE THIS SET OF DESIRED CDU ANGLES (SEE DESCRIPTION IN R60).
#
# WITH THIS INFORMATION KALCMANU DETERMINES THE DIRECTION OF THE SINGLE EQUIVALENT ROTATION (COF ALSO U) AND THE
# MAGNITUDE OF THE ROTATION (AM) TO BRING THE S/C FROM ITS INITIAL ORIENTATION TO ITS FINAL ORIENTATION.
# THIS DIRECTION REMAINS FIXED BOTH IN INERTIAL COORDINATES AND IN COMMANDED S/C AXES THROUGHOUT THE
#                  _
# MANEUVER.  ONCE COF AND AM HAVE BEEN DETERMINED, KALCMANU THEN EXAMINES THE MANEUVER TO SEE IF IT WILL BRING
#                                       _
# THE S/C THROUGH GIMBAL LOCK.  IF SO, COF AND AM ARE READJUSTED SO THAT THE S/C WILL JUST SKIM THE GIMBAL
# LOCK ZONE AND ALIGN THE X-AXIS.  IN GENERAL A FINAL YAW ABOUT X WILL BE NECESSARY TO COMPLETE THE MANEUVER.
# NEEDLESS TO SAY, NEITHER THE INITIAL NOR THE FINAL ORIENTATION CAN BE IN GIMBAL LOCK.
#
# FOR PROPER ATTITUDE CONTROL THE DIGITAL AUTOPILOT MUST BE GIVEN AN ATTITUDE REFERENCE WHICH IT CAN TRACK.
# KALCMANU DOES THIS BY GENERATING A REFERENCE OF DESIRED GIMBAL ANGLES (CDUXD, CDUYD, CDUZD) WHICH ARE UPDATED
# EVERY ONE SECOND DURING THE MANEUVER.  TO ACHIEVE A SMOOTHER SEQUENCE OF COMMANDS BETWEEN SUCCESSIVE UPDATES,
# THE PROGRAM ALSO GENERATES A SET OF INCREMENTAL CDU ANGLES (DELDCDU) TO BE ADDED TO CDU DESIRED BY THE DIGITAL
# AUTOPILOT.  KALCMANU ALSO CALCULATES THE COMPONENT MANEUVER RATES (OMEGAPD, OMEGAQD, OMEGARD), WHICH CAN
#                                      _
# BE DETERMINED SIMPLY BY MULTIPLYING COF BY SOME SCALAR (ARATE) CORRESPONDING TO THE DESIRED ROTATIONAL RATE.
#
# AUTOMATIC MANEUVERS ARE TIMED WTH THE HELP OF WAITLIST SO THAT AFTER A SPECIFIED INTERVAL THE Y AND Z
# DESIRED RATES ARE SET TO ZERO AND THE DESIRED CDU ANGLES (CDUYD, CDUZD) ARE SET EQUAL TO THE FINAL DESIRED CDU
# ANGLES (CTHETA, CPSI).  IF ANY YAW REMAINS DUE TO GIMBAL LOCK AVOIDANCE, THE FINAL YAW MANEUVER IS
# CALCULATED AND THE DESIRED YAW RATE SET TO SOME FIXED VALUE (ROLLRATE = + OR - 2 DEGREES PER SEC).
# IN THIS CASE ONLY AN INCREMENTAL CDUX ANGLE (DELFROLL) IS SUPPLIED TO THE DAP.  AT THE END OF THE YAW
# MANEUVER OR IN THE EVENT THAT THERE WAS NO FINAL YAW, CDUXD IS SET EQUAL TO CPHI AND THE X-AXIS DESIRED
# RATE SET TO ZERO.  THUS, UPON COMPLETION OF THE MANEUVER THE S/C WILL FINISH UP IN A LIMIT CYCLE ABOUT THE
# DESIRED GIMBAL ANGLES.
#
# PROGRAM LOGIC FLOW
#
# KALCMANU IS CALLED AS A HIGH PRIORITY JOB WITH ENTRY POINTS AT KALCMAN3 AND VECPOINT.  IT FIRST PICKS
# UP THE CURRENT CDU ANGLES TO BE USED AS THE BASIS FOR ALL COMPUTATIONS INVOLVING THE INITIAL S/C ORIENTATION.
```

```
# IT THEN DETERMINES THE DIRECTION COSINE MATRICES RELATING BOTH THE INITIAL AND FINAL S/C ORIENTATION TO STABLE
#               *   *                                                                               *
# MEMBER AXES (MIS,MFS).  IT ALSO COMPUTES THE MATRIX RELATING FINAL S/C AXES TO INITIAL S/C AXES (MFI).  THE
# ANGLE OF ROTATION (AM) IS THEN EXTRACTED FROM THIS MATRIX, AND TEST ARE MADE TO DETERMINE IF
#
#       A)      AM LESS THAN .25 DEGREES (MINANG)
#       B)      AM GREATER THAN 170 DEGREES (MAXANG)
#
# IF AM IS LESS THAN .25 DEGREES, NO COMPLICATED AUTOMATIC MANEUVERING IS NECESSARY.  THEREFORE, WE CAN SIMPLY
# SET CDU DESIRED EQUAL TO THE FINAL CDU DESIRED ANGLES AND TERMINATE THE JOB.
#
# IF AM IS GREATER THAN .25 DEGREES BUT LESS THAN 170 DEGREES THE AXES OF THE SINGLE EQUIVALENT ROTATION
#   _                                                        *
# (COF) IS EXTRACTED FROM THE SKEW SYMMETRIC COMPONENTS OF MFI.
#                                                                                     *     *
# IF AM GREATER THAN 170 DEGREES AN ALTERNATE METHOD EMPLOYING THE SYMMETRIC PART OF MFI (MFISYM) IS USED
#               _
# TO DETERMINE COF.
#
# THE PROGRAM THEN CHECKS TO SEE IF THE MANEUVER AS COMPUTED WILL BRING THE S/C THROUGH GIMBAL LOCK.  IF
# SO, A NEW MANEUVER IS CALCULATED WHICH WILL JUST SKIM THE GIMBAL LOCK ZONE AND ALIGN THE S/C X-AXIS.  THIS
# METHOD ASSURES THAT THE ADDITIONAL MANEUVERING TO AVOID GIMBAL LOCK WILL BE KEPT TO A MINIMUM.  SINCE A FINAL
# P AXIS YAW WILL BE NECESSARY, A SWITCH IS RESET (STATE SWITCH 31) TO ALLOW FOR THE COMPUTATION OF THIS FINAL
# YAW.
#
# AS STATED PREVIOUSLY, KALCMANU GENERATES A SEQUENCE OF DESIRED GIMBAL ANGLES WHICH ARE UPDATED EVERY
#                                                                                               _
# SECOND.  THIS IS ACCOMPLISHED BY A SMALL ROTATION OF THE DESIRED S/C FRAME ABOUT THE VECTOR COF.  THE NEW
# DESIRED REFERENCE MATRIX IS THEN,
#        *               *       *
#       MIS     =       MIS     DEL
#          N+1             N
#        *
# WHERE DEL IS THE MATRIX CORRESPONDING TO THIS SMALL ROTATION.  THE NEW CDU ANGLES CAN THEN BE EXTRACTED
#       *
# FROM MIS.
#
# AT THE BEGINNING OF THE MANEUVER THE AUTOPILOT DESIRED RATES (OMEGAPD, OMEGAQD, OMEGARD) AND THE
# MANEUVER TIMINGS ARE ESTABLISHED.  ON THE FIRST PASS AND ON ALL SUBSEQUENT UPDATES THE CDU DESIRED
# ANGLES ARE LOADED WITH THE APPROPRIATE VALUES AND THE INCREMENTAL CDU ANGLES ARE COMPUTED.  THE AGC CLOCKS
# (TIME1 AND TIME2) ARE THEN CHECKED TO SEE IF THE MANEUVER WILL TERMINATE BEFORE THE NEXT UPDATE.  IF
# NOT, KALCMANU CALLS FOR ANOTHER UPDATE (RUN AS A JOB WITH PRIORITY TBD) IN ONE SECOND.  ANY DELAYS IN THIS
# CALLING SEQUENCE ARE AUTOMATICALLY COMPENSATED IN CALLING FOR THE NEXT UPDATE.
#
# IF IT IS FOUND THAT THE MANEUVER IS TO TERMINATE BEFORE THE NEXT UPDATE A ROUTINE IS CALLED (AS A WAIT-
# LIST TASK) TO STOP THE MANEUVER AT THE APPROPRIATE TIME AS EXPLAINED ABOVE.
```

```
# CALLING SEQUENCE
#
# IN ORDER TO PERFORM A KALCMANU SUPERVISED MANEUVER, THE COMMANDED GIMBAL ANGLES MUST BE PRECOMPUTED AND
# STORED IN LOCATIONS CPHI, CTHETA, CPSI.  THE USER'S PROGRAM MUST THEN CLEAR STATE SWITCH NO 33 TO ALLOW THE
# ATTITUDE MANEUVER ROUTINE TO PERFORM ANY FINAL P-AXIS YAW INCURRED BY AVOIDING GIMBAL LOCK.  THE MANEUVER IS
# THEN INITIATED BY ESTABLISHING THE FOLLOWING EXECUTIVE JOB
#                               *
#       CAF     PRIO XX
#                             --
#       INHINT
#       TC      FINDVAC
#       2CADR   KALCMAN3
#       RELINT
#
# THE USER'S PROGRAM MAY EITHER CONTINUE OR WAIT FOR THE TERMINATION OF THE MANEUVER.  IF THE USER WISHES TO
# WAIT, HE MAY PUT HIS JOB TO SLEEP WITH THE FOLLOWING INSTRUCTIONS:
#
#       L       TC      BANKCALL
#       L+1     CADR    ATTSTALL
#       L+2     (BAD RETURN)
#       L+3     (GOOD RETURN)
#
# UPON COMPLETION OF THE MANEUVER, THE PROGRAM WILL BE AWAKENED AT L+3 IF THE MANEUVER WAS COMPLETED
# SUCCESSFULLY, OR AT L+2 IF THE MANEUVER WAS ABORTED.  THIS ABORT WOULD OCCUR IF THE INITIAL OR FINAL ATTITUDE
# WAS IN GIMBAL LOCK.
#
# *** NOTA BENE ***  IF IT IS ASSUMED THAT THE DESIRED MANEUVERING RATE (0.5, 2, 5, 10 DEG/SEC) HAS BEEN SELECTED BY
# KEYBOARD ENTRY PRIOR TO THE EXECUTION OF KALCMANU.
#
# IT IS ALSO ASSUMED THAT THE AUTOPILOT IS IN THE AUTO MODE.  IF THE MODE SWITCH IS CHANGED DURING THE
# MANEUVER, KALCMANU WILL TERMINATE VIA GOODEND WITHIN 1 SECOND SO THAT R60 MAY REQUEST A TRIM OF THE S/C ATTITUDE
# SUBROUTINES.
#
# KALCMANU USES A NUMBER OF INTERPRETIVE SUBROUTINES WHICH MAY BE OF GENERAL INTEREST.  SINCE THESE ROUTINES
# WERE PROGRAMMED EXCLUSIVELY FOR KALCMANU, THEY ARE NOT, AS YET, GENERALLY AVAILABLE FOR USE BY OTHER PROGRAMS.
#
# MXM3
# ----
#
# THIS SUBROUTINE MULTIPLIES TWO 3X3 MATRICES AND LEAVES THE RESULT IN THE FIRST 18 LOCATIONS OF THE PUSH
# DOWN LIST, I.E.,
#                       [ M     M     M  ]
#                       [  0     1     2 ]
#       *               [                ]          *               *
#       M       =       [ M     M     M  ]    =     M1      X       M2
#                       [  3     4     5 ]
#                       [                ]
#                       [ M     M     M  ]
#                       [  6     7     8 ]
```

```
#                                                                                                     *
# INDEX REGISTER X1 MUST BE LOADED WITH THE COMPLEMENT OF THE STARTING ADDRESS FOR M1, AND X2 MUST BE
#                                                                     *
# LOADED WITH THE COMPLEMENT OF THE STARTING ADDRESS FOR M2.  THE ROUTINE USES THE FIRST 20 LOCATIONS OF THE PUSH
# DOWN LIST.  THE FIRST ELEMENT OF THE MATRIX APPEARS IN PDO.  PUSH UP FOR M .
#                                                                              8
# TRANSPOS
# --------
#
# THIS ROUTINE TRANSPOSES A 3X3 MATRIX AND LEAVES THE RESULT IN THE PUSH DOWN LIST, I.E.,
#
#       *               * T
#       M       =       M1
#
# INDEX REGISTER X1 MUST CONTAIN THE COMPLEMENT OF THE STARTING ADDRESS FOR M1.  PUSH UP FOR THE FIRST AND SUB-
#                        *
# SEQUENT COMPONENTS OF M.  THIS SUBROUTINE ALSO USES THE FIRST 20 LOCATIONS OF THE PUSH DOWN LIST.
#
# CDU TO DCM
# ----------
#
# THIS SUBROUTINE CONVERTS THREE CDU ANGLES IN T(MPAC) TO A DIRECTION COSINE MATRIX (SCALED BY 2) RELATING
# THE CORRESPONDING S/C ORIENTATIONS TO THE STABLE MEMBER FRAME.  THE FORMULAS FOR THIS CONVERSION ARE
#
#       M       =       COSY COSZ
#        0
#
#       M       =       -COSY SINZ COSX + SINY SINX
#        1
#
#       M       =       COSY SINZ SINX + SINY COSX
#        2
#
#       M       =       SINZ
#        3
#
#       M       =       COSZ COSX
#        4
#
#       M       =       -COSZ SINX
#        5
#
#       M       =       -SINY COSZ
#        6
#
#       M       =       SINY SINZ COSX + COSY SINX
#        7
```

```
#       M       =       -SINY SINZ SINX + COSY COSX
#        8
#
# WHERE         X       =       OUTER GIMBAL ANGLE
#               Y       =       INNER GIMBAL ANGLE
#               Z       =       MIDDLE GIMBAL ANGLE
#
# THE INTERPRETATION OF THIS MATRIX IS AS FOLLOWS:
#
# IF A , A , A  REPRESENT THE COMPONENTS OF A VECTOR IN S/C AXES THEN THE COMPONENTS OF THE SAME VECTOR IN
#     X   Y   Z
# STABLE MEMBER AXES (B , B , B ) ARE
#                      X   Y   Z
#
#       [ B  ]                  [ A  ]
#       [  X ]                  [  X ]
#       [    ]                  [    ]
#       [ B  ]            *     [ A  ]
#       [  Y ]     =      M     [  Y ]
#       [    ]                  [    ]
#       [ B  ]                  [ B  ]
#       [  Z ]                  [  Z ]
#
# THE SUBROUTINE WILL STORE THIS MATRIX IN SEQUENTIAL LOCATIONS OF ERASABLE MEMORY AS SPECIFIED BY THE CALLING
#                                                                                                           *
# PROGRAM.  TO DO THIS THE CALLING PROGRAM MUST FIRST LOAD X2 WITH THE COMPLEMENT OF THE STARTING ADDRESS FOR M.
#
# INTERNALLY, THE ROUTINE USES THE FIRST 16 LOCATIONS OF THE PUSH DOWN LIST, ALSO STEP REGISTER S1 AND INDEX
# REGISTER X2.
#
# DCM TO CDU
# ----------
#                                                                    *
# THIS ROUTINE EXTRACTS THE CDU ANGLES FROM A DIRECTION COSINE MATRIX (M SCALED BY 2) RELATING S/C AXIS TO
#                                                                               *
# STABLE MEMBER AXES.  X1 MUST CONTAIN THE COMPLEMENT OF THE STARTING ADDRESS FOR M.  THE SUBROUTINE LEAVES THE
# CORRESPONDING GIMBAL ANGLES IN V(MPAC) AS DOUBLE PRECISION 1'S COMPLEMENT ANGLES SCALED BY 2PI.  THE FORMULAS
# FOR THIS CONVERSION ARE
#
#       Z       =       ARCSIN (M  )
#                                3
#
#       Y       =       ARCSIN (-M /COSZ)
#                                 6
#
# IF M  IS NEGATIVE, Y IS REPLACED BY PI SGN Y - Y.
#     0
```

```
#       X       =       ARCSIN (-M /COSZ)
#                                 5
#
# IF M  IS NEGATIVE, X IS REPLACED BY PI SGN X - X.
#     4
#
# THIS ROUTINE DOES NOT SET THE PUSH DOWN POINTER, BUT USES THE NEXT 8 LOCATIONS OF THE PUSH DOWN LIST AND
# RETURNS THE POINTER TO ITS ORIGINAL SETTING.  THIS PROCEDURE ALLOWS THE CALLER TO STORE THE MATRIX AT THE TOP OF
# THE PUSH DOWN LIST.
#
# DELCOMP
# -------
#                                                      *
# THIS ROUTINE COMPUTES THE DIRECTION COSINE MATRIX (DEL) RELATING ON
#                                                                          _
# IS ROTATED WITH RESPECT TO THE FIRST BY AN ANGLE, A, ABOUT A UNIT VECTOR U.  THE FORMULA FOR THIS MATRIX IS
#
#        *              *        _ _T              *
#       DEL     =       I COSA + U U  (1 - COSA) + V  SINA
#                                                   X
#
# WHERE          *               [ 1    0    0 ]
#               I       =       [ 0    1    0 ]
#                               [ 0    0    1 ]
#
#                               [    2                             ]
#                               [  U           U  U          U  U  ]
#                               [   X           X  Y          X  Z ]
#                               [                                  ]
#               _ _T            [                 2                ]
#               U U     =       [ U  U          U            U  U  ]
#                               [  Y  X          Y            Y  Z ]
#                               [                                  ]
#                               [                              2   ]
#                               [ U  U         U  U           U    ]
#                               [  Z  X         Z  Y           Z   ]
#
#
#                               [   0           -U              U  ]
#                               [                 Z              Y ]
#               *               [                                  ]
#               V       =       [  U            0              -U  ]
#                X              [   Z                             X ]
#                               [                                  ]
#                               [ -U            U               0  ]
#                               [   Y             X                ]
#
```

```
#
#       _
#       U       =       UNIT ROTATION VECTOR RESOLVED INTO S/C AXES.
#       A       =       ROTATION ANGLE
#
#                        *
# THE INTERPRETATION OF DEL IS AS FOLLOWS:
#
# IF A , A , A  REPRESENT THE COMPONENTS OF A VECTOR IN THE ROTATED FRAME, THEN THE COMPONENTS OF THE SAME
#     X   Y   Z
# VECTOR IN THE ORIGINAL S/C AXES (B , B , B ) ARE
#                                    X   Y   Z
#
#       [ B  ]                  [ A  ]
#       [  X ]                  [  X ]
#       [    ]                  [    ]
#       [ B  ]            *     [ A  ]
#       [  Y ]     =     DEL    [  Y ]
#       [    ]                  [    ]
#       [ B  ]                  [ B  ]
#       [  Z ]                  [  Z ]
#
# THE ROUTINE WILL STORE THIS MATRIX (SCALED UNITY) IN SEQUENTIAL LOCATIONS OF ERASABLE MEMORY BEGINNING WITH
#                                                                                             _
# THE LOCATION CALLED DEL.  IN ORDER TO USE THE ROUTINE, THE CALLING PROGRAM MUST FIRST STORE U (A HALF UNIT
# DOUBLE PRECISION VECTOR) IN THE SET OF ERASABLE LOCATIONS BEGINNING WITH THE ADDRESS CALLED COF.  THE ANGLE, A,
# MUST THEN BE LOADED INTO D(MPAC).
#
# INTERNALLY, THE PROGRAM ALSO USES THE FIRST 10 LOCATIONS OF THE PUSH DOWN LIST.
#
# READCDUK
# --------
#
# THIS BASIC LANGUAGE SUBROUTINE LOADS T(MPAC) WITH THE THREE CDU ANGLES.
#
# SIGNMPAC
# --------
#
# THIS IS A BASIC LANGUAGE SUBROUTINE WHICH LIMITS THE MAGNITUDE OF D(MPAC) TO + OR - DPOSMAX ON OVERFLOW.
#
# PROGRAM STORAGE ALLOCATION
#
#       1)      FIXED MEMORY            1059 WORDS
#       2)      ERASABLE MEMORY           98
#       3)      STATE SWITCHES             3
```

```
#       4)      FLAGS                       1
#
# JOB PRIORITIES
#
#       1)      KALCMANU                 TBD
#       2)      ONE SECOND UPDATE        TBD
#
# SUMMARY OF STATE SWITCHES AND FLAGWORDS USED BY KALCMANU.
#
#       STATE           FLAGWRD 2       SETTING         MEANING
#       SWITCH NO.      BIT NO.
#
#         *
#       31              14              0               MANEUVER WENT THROUGH GIMBAL LOCK
#                                       1               MANEUVER DID NOT GO THROUGH GIMBAL LOCK
#         *
#       32              13              0               CONTINUE UPDATE PROCESS
#                                       1               START UPDATE PROCESS
#
#       33              12              0               PERFORM FINAL P AXIS YAW IF REQUIRED
#                                       1               IGNORE ANY FINAL P-AXIS YAW
#
#       34              11              0               SIGNAL END OF KALCMANU
#                                       1               KALCMANU IN PROCESS.    USER MUST SET SWITCH BEFORE INITIATING
#
#       * INTERNAL TO KALCMANU
#
# SUGGESTIONS FOR PROGRAM INTEGRATION
#
# THE FOLLOWING VARIABLES SHOULD BE ASSIGNED TO UNSWITCH ERASABLE:
#
#       CPHI
#       CTHETA
#       CPSI
#       POINTVSM +5
#       SCAXIS   +5
#       DELDCDU
#       DELDCDU1
#       DELDCDU2
#       RATEINDX
#
# THE FOLLOWING SUBROUTINES MAY BE PUT IN A DIFFERENT BANK
#
#       MXM3
```

```
#       TRANSPGS
#       SIGNMPAC
#       READCDUK
#       CDUTODCM
```

```
                BANK    15
                SETLOC  KALCMON1
                BANK

                EBANK=  BCDU

# THE THREE DESIRED CDU ANGLES MUST BE STORED AS SINGLE PRECISION TWO'S COMPLEMENT ANGLES IN THE THREE SUCCESSIVE
# LOCATIONS, CPHI, CTHETA, CPSI.

                COUNT*  $$/KALC
KALCMAN3        TC      INTPRET         # PICK UP THE CURRENT CDU ANGLES AND
                RTB                     #       COMPUTE THE MATRIX FROM INITIAL S/C
                        READCDUK        #       AXES TO FINAL S/C AXES.
                STORE   BCDU            # STORE INITIAL S/C ANGLES
                SLOAD   ABS             # CHECK THE MAGNITUDE OF THE DESIRED
                        CPSI            # MIDDLE GIMBAL ANGLE
                DSU     BPL
                        LOCKANGL        # IF GREATER THAN 70 DEG ABORT MANEUVER
                        TOOBADF
                AXC,2   TLOAD
                        MIS
                        BCDU
                CALL                    # COMPUTE THE TRANSFORMATION FROM INITIAL
                        CDUTODCM        # S/C AXES TO STABLE MEMBER AXES
                AXC,2   TLOAD
                        MFS             # PREPARE TO CALCULATE ARRAY MFS
                        CPHI
                CALL
                        CDUTODCM
SECAD           AXC,1   CALL            # MIS AND MFS ARRAYS CALCULATED         $2
                        MIS
                        TRANSPOS
                VLOAD   STADR
                STOVL   TMIS +12D
                STADR
                STOVL   TMIS +6
                STADR
                STORE   TMIS            # TMIS = TRANSPOSE(MIS) SCALED BY 2
                AXC,1   AXC,2
                        TMIS
                        MFS
                CALL
                        MXM3
                VLOAD   STADR
                STOVL   MFI +12D
                STADR
                STOVL   MFI +6
                STADR
                STORE   MFI             # MFI = TMIS MFS (SCALED BY 4)
                SETPD   CALL            # TRANSPOSE MFI IN PD LIST
```

```
                        18D
                        TRNSPSPD
                VLOAD   STADR
                STOVL   TMFI    +12D
                STADR
                STOVL   TMFI    +6
                STADR
                STORE   TMFI            # TMFI = TRANSPOSE (MFI) SCALED BY 4

# CALCULATE COFSKEW AND MFISYM

                DLOAD   DSU
                        TMFI    +2
                        MFI     +2
                PDDL    DSU             # CALCULATE COF SCALED BY 2/SIN(AM)
                        MFI     +4
                        TMFI    +4
                PDDL    DSU
                        TMFI    +10D
                        MFI     +10D
                VDEF
                STORE   COFSKEW         # EQUALS MFISKEW

# CALCULATE AM AND PROCEED ACCORDING TO ITS MAGNITUDE

                DLOAD   DAD
                        MFI
                        MFI     +16D
                DSU     DAD
                        DP1/4TH
                        MFI     +8D
                STORE   CAM             # CAM = (MFI0+MFI4+MFI8-1)/2 HALF SCALE
                ARCCOS
                STORE   AM              # AM=ARCCOS(CAM)        (AM SCALED BY 2)
                DSU     BPL
                        MINANG
                        CHECKMAX
                TLOAD                   # MANEUVER LESS THAN .25 DEGREES
                        CPHI            # GO DIRECTLY INTO ATTITUDE HOLD
                STCALL  CDUXD           # ABOUT COMMANDED ANGLES
                        TOOBADI         # STOP RATE AND EXIT

CHECKMAX        DLOAD   DSU
                        AM
                        MAXANG
                BPL     VLOAD
                        ALTCALC         # UNIT
                        COFSKEW         # COFSKEW
                UNIT
                STORE   COF             # COF IS THE MANEUVER AXIS
```

```
                GOTO                            # SEE IF MANEUVER GOES THRU GIMBAL LOCK
                        LOCSKIRT
ALTCALC         VLOAD   VAD                     # IF AM GREATER THAN 170 DEGREES
                        MFI
                        TMFI
                VSR1
                STOVL   MFISYM
                        MFI     +6
                VAD     VSR1
                        TMFI    +6
                STOVL   MFISYM  +6
                        MFI     +12D
                VAD     VSR1
                        TMFI    +12D
                STORE   MFISYM  +12D            # MFISYM=(MFI+TMFI)/2   SCALED BY 4

# CALCULATE COF

                DLOAD   SR1
                        CAM
                PDDL    DSU                     # PDO CAM                                $4
                        DPHALF
                        CAM
                BOVB    PDDL                    # PS2 1 - CAM                            $2
                        SIGNMPAC
                        MFISYM  +16D
                DSU     DDV
                        0
                        2
                SQRT    PDDL                    # COFZ = SQRT(MFISYM8-CAM)/(1-CAM)
                        MFISYM  +8D             #                               $ ROOT 2
                DSU     DDV
                        0
                        2
                SQRT    PDDL                    # COFY = SQRT(MFISYM4-CAM)/(1-CAM)  $ROOT2
                        MFISYM
                DSU     DDV
                        0
                        2
                SQRT    VDEF                    # COFX = SQRT(MFISYM-CAM)/(1-CAM)  $ROOT 2
                UNIT
                STORE   COF

# DETERMINE LARGEST COF AND ADJUST ACCORDINGLY

COFMAXGO        DLOAD   DSU
                        COF
                        COF     +2
                BMN     DLOAD                   # COFY G COFX
```

```
                            COMP12
                            COF
                DSU         BMN
                            COF     +4
                            METHOD3         # COFZ G COFX OR COFY
                GOTO
                            METHOD1         # COFX G COFY OR COFZ
COMP12          DLOAD       DSU
                            COF     +2
                            COF     +4
                BMN
                            METHOD3         # COFZ G COFY OR COFX

METHOD2         DLOAD       BPL             # COFY MAX
                            COFSKEW +2      # UY
                            U2POS
                VLOAD       VCOMP
                            COF
                STORE       COF
U2POS           DLOAD       BPL
                            MFISYM  +2      # UX UY
                            OKU21
                DLOAD       DCOMP           # SIGN OF UX OPPOSITE GARBLED
                            COF
                STORE       COF
OKU21           DLOAD       BPL
                            MFISYM +10D     # UY UZ
                            LOCSKIRT
                DLOAD       DCOMP           # SIGN OF UZ OPPOSITE TO UY
                            COF     +4
                STORE       COF     +4
                GOTO
                            LOCSKIRT
METHOD1         DLOAD       BPL             # COFX MAX
                            COFSKEW         # UX
                            U1POS
                VLOAD       VCOMP
                            COF
                STORE       COF
U1POS           DLOAD       BPL
                            MFISYM  +2      # UX UY
                            OKU12
                DLOAD       DCOMP
                            COF     +2      # SIGN OF UY OPPOSITE TO UX
                STORE       COF     +2
OKU12           DLOAD       BPL
                            MFISYM  +4      # UX UZ
                            LOCSKIRT
                DLOAD       DCOMP           # SIGN OF UZ OPPOSITE TO UY
                            COF     +4
```

```
                STORE   COF     +4
                GOTO
                        LOCSKIRT
METHOD3         DLOAD   BPL             # COFZ MAX
                        COFSKEW +4      # UZ
                        U3POS
                VLOAD   VCOMP
                        COF
                STORE   COF
U3POS           DLOAD   BPL
                        MFISYM  +4      # UX UZ
                        OKU31
                DLOAD   DCOMP
                        COF             # SIGN OF UX OPPOSITE TO UZ
                STORE   COF
OKU31           DLOAD   BPL
                        MFISYM  +10D    # UY UZ
                        LOCSKIRT
                DLOAD   DCOMP
                        COF     +2      # SIGN OF UY OPPOSITE TO UZ
                STORE   COF     +2
                GOTO
                        LOCSKIRT
```

```
# MATRIX OPERATIONS

                BANK    13
                SETLOC  KALCMON2
                BANK

                EBANK=  BCDU

MXM3            SETPD   VLOAD*          # MXM3 MULTIPLIES 2 3X3 MATRICES
                        0               # AND LEAVES RESULT IN PD LIST
                        0,1             # AND MPAC
                VXM*    PDVL*
                        0,2
                        6,1
                VXM*    PDVL*
                        0,2
                        12D,1
                VXM*    PUSH
                        0,2
                RVQ

# RETURN WITH MIXM2 IN PD LIST

TRANSPOS        SETPD   VLOAD*          # TRANSPOS TRANSPOSES A 3X3 MATRIX
                        0               #       AND LEAVES RESULT IN PD LIST
                        0,1             # MATRIX ADDRESS IN XR1
                PDVL*   PDVL*
                        6,1
                        12D,1
                PUSH                    # MATRIX IN PD
TRNSPSPD        EXIT                    # ENTER WITH MATRIX AT 0 IN PD LIST
                INDEX   FIXLOC
                DXCH    12
                INDEX   FIXLOC
                DXCH    16
                INDEX   FIXLOC
                DXCH    12
                INDEX   FIXLOC
                DXCH    14
                INDEX   FIXLOC
                DXCH    4
                INDEX   FIXLOC
                DXCH    14
                INDEX   FIXLOC
                DXCH    2
                INDEX   FIXLOC
                DXCH    6
                INDEX   FIXLOC
                DXCH    2
```

```
                TC      INTPRET
                RVQ

                BANK    15
                SETLOC  KALCMON1
                BANK

                EBANK=  BCDU

MINANG          2DEC    0.00069375

MAXANG          2DEC    0.472222222

# GIMBAL LOCK CONSTANTS

# D = MGA CORRESPONDING TO GIMBAL LOCK = 60 DEGREES
#       NGL = BUFFER ANGLE (TO AVOID DIVISIONS BY ZERO) = 2 DEGREES

SD              2DEC    .433015         # = SIN(D)                      $2

K3S1            2DEC    .86603          # = SIN(D)                      $1

K4              2DEC    -.25            # = -COS(D)                     $2

K4SQ            2DEC    .125            # = COS(D)COS(D)                $2

SNGLCD          2DEC    .008725         # = SIN(NGL)COS(D)              $2

CNGL            2DEC    .499695         # COS(NGL)                      $2

LOCKANGL        DEC     .388889         # = 70 DEGREES

# INTERPRETIVE SUBROUTINE TO READ THE CDU ANGLES

READCDUK        CA      CDUZ            # LOAD T(MPAC) WITH CDU ANGLES
                TS      MPAC    +2
                EXTEND
                DCA     CDUX            # AND CHANGE MODE TO TRIPLE PRECISION
                TCF     TLOAD   +6

CDUTODCM        AXT,1   SSP
                OCT     3
                        S1
                OCT     1               # SET XR1, S1, AND PD FOR LOOP
                STORE   7
                SETPD
                        0
LOOPSIN         SLOAD*  RTB
                        10D,1
                        CDULOGIC
```

```
                STORE   10D             # LOAD PD WITH  0 SIN(PHI)
                SIN     PDDL            #               2 COS(PHI)
                        10D             #               4 SIN(THETA)
                COS     PUSH            #               6 COS(THETA)
                TIX,1   DLOAD           #               8 SIN(PSI)
                        LOOPSIN         #               10 COS(PSI)
                        6
                DMP     SL1
                        10D
                STORE   0,2             # C0 = COS(THETA)COS(PSI)
                DLOAD   DMP
                        4
                        0
                PDDL    DMP             # (PD6 SIN(THETA)SIN(PHI))
                        6
                        8D
                DMP     SL1
                        2
                BDSU    SL1
                        12D
                STORE   2,2             # C1=-COS(THETA)SIN(PSI)COS(PHI)
                DLOAD   DMP
                        2
                        4
                PDDL    DMP             # (PD7 COS(PHI)SIN(THETA)) SCALED 4
                        6
                        8D
                DMP     SL1
                        0
                DAD     SL1
                        14D
                STORE   4,2             # C2=COS(THETA)SIN(PSI)SIN(PHI)
                DLOAD
                        8D
                STORE   6,2             # C3=SIN(PSI)
                DLOAD
                        10D
                DMP     SL1
                        2
                STORE   8D,2            # C4=COS(PSI)COS(PHI)
                DLOAD   DMP
                        10D
                        0
                DCOMP   SL1
                STORE   10D,2           # C5=-COS(PSI)SIN(PHI)
                DLOAD   DMP
                        4
                        10D
                DCOMP   SL1
                STORE   12D,2           # C6=-SIN(THETA)COS(PSI)
```

```
                DLOAD
                DMP     SL1             # (PUSH UP 7)
                        8D
                PDDL    DMP             # (PD7 COS(PHI)SIN(THETA)SIN(PSI)) SCALE 4
                        6
                        0
                DAD     SL1             # (PUSH UP 7)
                STADR                   # C7=COS(PHI)SIN(THETA)SIN(PSI)
                STORE   14D,2           #       +COS(THETA)SIN(PHI)
                DLOAD
                DMP     SL1             # (PUSH UP 6)
                        8D
                PDDL    DMP             # (PD6 SIN(THETA)SIN(PHI)SIN(PSI)) SCALE 4
                        6
                        2
                DSU     SL1             # (PUSH UP 6)
                STADR
                STORE   16D,2           # C8=-SIN(THETA)SIN(PHI)SIN(PSI)
                RVQ                     # +COS(THETA)COS(PHI)

# CALCULATION OF THE MATRIX DEL......
#
#        *         *          __T           *
#       DEL = (IDMATRIX)COS(A)+UU (1-COS(A))+UX SIN(A)          SCALED 1
#
#       WHERE Ū IS A UNIT VECTOR (DP SCALED 2) ALONG THE AXIS OF ROTATION.
#       A IS THE ANGLE OF ROTATION (DP SCALED 2)
#
#       UPON ENTRY, THE STARTING ADDRESS OF Ū IS COF, AND A IS IN MPAC

DELCOMP         SETPD   PUSH            # MPAC CONTAINS THE ANGLE A
                        0
                SIN     PDDL            # PD0 = SIN(A)
                COS     PUSH            # PD2 = COS(A)
                SR2     PDDL            # PD2 = COS(A)                          $8
                BDSU    BOVB
                        DPHALF
                        SIGNMPAC
                PDDL                    # PDA = 1-COS(A)

# COMPUTE THE DIAGONAL COMPONENTS OF DEL

                        COF
                DSQ     DMP
                        4
                DAD     SL3
                        2
                BOVB
                        SIGNMPAC
```

```
                STODL   KEL             # UX UX(1-COS(A)) +COS(A)               $1
                        COF     +2
                DSQ     DMP
                        4
                DAD     SL3
                        2
                BOVB
                        SIGNMPAC
                STODL   KEL     +8D     # UY UY(1-COS(A)) +COS(A)               $1
                        COF     +4
                DSQ     DMP
                        4
                DAD     SL3
                        2
                BOVB
                        SIGNMPAC
                STORE   KEL     +16D    # UZ UZ(1-COS(A)) +COS(A)               $1

# COMPUTE THE OFF DIAGONAL TERMS OF DEL

                DLOAD   DMP
                        COF
                        COF     +2
                DMP     SL1
                        4
                PDDL    DMP             # D6    UX UY (1-COS A)                 $4
                        COF     +4
                        0
                PUSH    DAD             # D8    UZ SIN A                        $4
                        6
                SL2     BOVB
                        SIGNMPAC
                STODL   KEL     +6
                BDSU    SL2
                BOVB
                        SIGNMPAC
                STODL   KEL     +2
                        COF
                DMP     DMP
                        COF     +4
                        4
                SL1     PDDL            # D6    UX UZ (1-COS A)                 $4
                        COF     +2
                DMP     PUSH            # D8    UY SIN(A)
                        0
                DAD     SL2
                        6
                BOVB
                        SIGNMPAC
                STODL   KEL     +4      # UX UZ (1-COS(A))+UY SIN(A)
```

```
                BDSU    SL2
                BOVB
                        SIGNMPAC
                STODL   KEL      +12D   # UX UZ (1-COS(A))-UY SIN(A)
                        COF     +2
                DMP     DMP
                        COF     +4
                        4
                SL1     PDDL            # D6    UY UZ (1-COS(A))                $ 4
                        COF
                DMP     PUSH            # D8    UX SIN(A)
                        0
                DAD     SL2
                        6
                BOVB
                        SIGNMPAC
                STODL   KEL     +14D    # UY UZ(1-COS(A)) +UX SIN(A)
                BDSU    SL2
                BOVB
                        SIGNMPAC
                STORE   KEL     +10D    # UY UZ (1-COS(A)) -UX SIN(A)
                RVQ

# DIRECTION COSINE MATRIX TO CDU ANGLE ROUTINE
# X1 CONTAINS THE COMPLEMENT OF THE STARTING ADDRESS FOR MATRIX (SCALED 2).
# LEAVE CDU ANGLES SCALED 2PI IN V(MPAC).
# COS(MGA) WILL BE LEFT IN S1 (SCALED 1).
#
# THE DIRECTION COSINE MATRIX RELATING S/C AXES TO STABLE MEMBER AXES CAN BE WRITTEN AS:
#
#       C  = COS(THETA) COS(PSI
#        0
#
#       C  = -COS(THETA) SIN(PSI) COS(PHI) + SIN(THETA) SIN(PHI)
#        1
#
#       C  = COS(THETA) SIN(PSI) SIN(PHI) + SIN(THETA) COS(PHI)
#        2
#
#       C  = SIN(PSI)
#        3
#
#       C  = COS(PSI) COS(PHI)
#        4
#
#       C  = -COS(PSI) SIN(PHI)
#        5
#
#       C  = -SIN(THETA) COS(PSI)
#        6
#
#       C  = SIN(THETA) SIN(PSI) COS(PHI) + COS (THETA) SIN(PHI)
#        7
#
#       C  = -SIN(THETA) SIN(PSI) SIN(PHI) + COS(THETA)COS(PHI)
#        8
```

```
#
# WHERE PHI = OGA
#       THETA = IGA
#       PSI = MGA

DCMTOCDU        DLOAD*  ARCSIN
                        6,1
                PUSH    COS             # PD +0         PSI
                SL1     BOVB
                        SIGNMPAC
                STORE   S1
                DLOAD*  DCOMP
                        12D,1
                DDV     ARCSIN
                        S1
                PDDL*   BPL             # PD +2         THETA
                        0,1             # MUST CHECK THE SIGN OF COS(THETA)
                        OKTHETA         # TO DETERMINE THE PROPER QUADRANT.
                DLOAD   DCOMP
                BPL     DAD
                        SUHALFA
                        DPHALF
                GOTO
                        CALCPHI
SUHALFA         DSU
                        DPHALF
CALCPHI         PUSH
OKTHETA         DLOAD*  DCOMP
                        10D,1
                DDV     ARCSIN
                        S1
                PDDL*   BPL             # PUSH DOWN PHI
                        8D,1
                        OKPHI
                DLOAD   DCOMP           # PUSH UP PHI
                BPL     DAD
                        SUHALFAP
                        DPHALF
                GOTO
                        VECOFANG
SUHALFAP        DSU     GOTO
                        DPHALF
                        VECOFANG
OKPHI           DLOAD                   # PUSH UP PHI
VECOFANG        VDEF    RVQ
```

```
# ROUTINES FOR TERMINATING THE AUTOMATIC MANEUVER AND RETURNING TO USER.

TOOBADF         EXIT
                TC      ALARM
                OCT     00401

                TCF     NOGO            # DO NOT ZERO ATTITUDE ERRORS

                TC      BANKCALL
                CADR    ZATTEROR        # ZERO ATTITUDE ERRORS

NOGO            TC      BANKCALL
                CADR    STOPRATE        # STOP RATES

                CAF     TWO
                INHINT                  # ALL RETURNS ARE NOW MADE VIA GOODEND
                TC      WAITLIST
                EBANK=  BCDU
                2CADR   GOODMANU

                TCF     ENDOFJOB

TOOBADI         EXIT
                TCF     NOGO
```

```
                BANK    15

                SETLOC  KALCMON1
                BANK

# DETECTING GIMBAL LOCK
LOCSKIRT        EQUALS  NOGIMLOC

NOGIMLOC        SET
                        CALCMAN3
WCALC           LXC,1   DLOAD*
                        RATEINDX        # CHOOSE THE DESIRED MANEUVER RATE
                        ARATE,1         # FROM A LIST OF FOUR
                SR4     CALL            # COMPUTE THE INCREMENTAL ROTATION MATRIX
                        DELCOMP         # DEL CORRESPONDING TO A 1 SEC ROTATION
                                        # ABOUT COF
                DLOAD*  VXSC
                        ARATE,1
                        COF
                STODL   BRATE           # COMPONENT MANEUVER RATES 45 DEG/SEC
                        AM
                DMP     DDV*
                        ANGLTIME
                        ARATE,1
                SR
                        5
                STORE   TM              # MANEUVER EXECUTION TIME SCALED AS T2
                SETGO
                        CALCMAN2        # 0(OFF) = CONTINUE MANEUVER
                        NEWANGL +1      # 1(ON) = START MANEUVER

# THE FOUR SELECTABLE FREE FALL MANEUVER RATES SELECTED BY
# LOADING RATEINDX WITH 0,2,4,6, RESPECTIVELY

ARATE           2DEC    .0088888888     # = 0.2 DEG/SEC         $ 22.5 DEG/SEC

                2DEC    .0222222222     # = 0.5 DEG/SEC         $ 22.5 DEG/SEC

                2DEC    .0888888888     # = 2.0 DEG/SEC         $ 22.5 DEG/SEC

                2DEC    .4444444444     # = 10.0 DEG/SEC        $ 22.5 DEG/SEC

ANGLTIME        2DEC    .0001907349     # = 1008-19 FUDGE FACTOR TO CONVERT
                                        # MANEUVER ANGLE TO MANEUVER TIME
```

```
# GENERATION OF STEERING COMMANDS FOR DIGITAL AUTOPILOT FREE FALL MANEUVERS
#
# NEW COMMANDS WILL BE GENERATED EVERY ONE SECOND DURING THE MANEUVER

                EBANK=  TTEMP

NEWDELHI        TC      BANKCALL        # CHECK FOR AUTO STABILIZATION
                CADR    ISITAUTO        # ONLY
                CCS     A
                TCF     NOGO -2
NEWANGL         TC      INTPRET
                AXC,1   AXC,2
                        MIS             # COMPUTE THE NEW MATRIX FROM S/C TO
                        KEL             # STABLE MEMBER AXES
                CALL
                        MXM3
                VLOAD   STADR
                STOVL   MIS +12D        # CALCULATE NEW DESIRED CDU ANGLES
                STADR
                STOVL   MIS +6D
                STADR
                STORE   MIS
                AXC,1   CALL
                        MIS
                        DCMTOCDU        # PICK UP THE NEW CDU ANGLES FROM MATRIX
                RTB
                        V1STO2S
                STORE   NCDU            # NEW CDU ANGLES
                BONCLR  EXIT
                        CALCMAN2
                        MANUSTAT        # TO START MANEUVER
                CAF     TWO             #           +0 OTHERWISE
INCRDCDU        TS      SPNDX
                INDEX   SPNDX
                CA      BCDU            # INITIAL CDU ANGLES
                EXTEND                  # OR PREVIOUS DESIRED CDU ANGLES
                INDEX   SPNDX
                MSU     NCDU
                EXTEND
                SETLOC  KALCMON1
                BANK
                MP      DT/TAU
                CCS     A               # CONVERT TO 2S COMPLEMENT
                AD      ONE
                TCF     +2
                COM
                INDEX   SPNDX
                TS      DELDCDU         # ANGLE INCREMENTS TO BE ADDED TO
                INDEX   SPNDX           # CDUXD, CDUYD, CDUZD EVERY TENTH SECOND
```

```
                CA      NCDU            # BY LEM DAP
                INDEX   SPNDX
                XCH     BCDU
                INDEX   SPNDX
                TS      CDUXD
                CCS     SPNDX
                TCF     INCRDCDU        # LOOP FOR THREE AXES

                RELINT

# COMPARE PRESENT TIME WITH TIME TO TERMINATE MANEUVER

TMANUCHK        TC      TIMECHK
                TCF     CONTMANU
                CAF     ONE
MANUSTAL        INHINT                  # END MAJOR PART OF MANEUVER WITHIN 1 SEC
                TC      WAITLIST        # UNDER WAITLIST CALL TO MANUSTOP
                EBANK=  TTEMP
                2CADR   MANUSTOP

                RELINT
                TCF     ENDOFJOB

TIMECHK         EXTEND
                DCS     TIME2
                DXCH    TTEMP
                EXTEND
                DCA     TM
                DAS     TTEMP
                CCS     TTEMP
                TC      Q
                TCF     +2
                TCF     2NDRETRN
                CCS     TTEMP +1
                TC      Q
                TCF     MANUOFF
                COM
MANUOFF         AD      ONESEK +1
                EXTEND
                BZMF    2NDRETRN
                INCR    Q
2NDRETRN        INCR    Q
                TC      Q

DT/TAU          DEC     .1

MANUSTAT        EXIT                    # INITIALIZATION ROUTINE
                EXTEND                  # FOR AUTOMATIC MANEUVERS
                DCA     TIME2
```

```
                DAS     TM              # TM+TO    MANEUVER COMPLETION TIME
                EXTEND
                DCS     ONESEK
                DAS     TM              # (TM+TO)-1
                INHINT
                CAF     TWO
RATEBIAS        TS      KSPNDX
                DOUBLE
                TS      KDPNDX
                INDEX   A
                CA      BRATE
                INDEX   KSPNDX          # STORE MANEUVER RATE IN
                TS      OMEGAPD         # OMEGAPD, OMEGAQD, OMEGARD
                EXTEND
                BZMF    +2              # COMPUTE ATTITUDE ERROR
                COM                     # OFFSET = (WX)ABS(WX)/2AJX
                EXTEND                  # WHERE AJX= 2-JET ACCELERATION
                MP      BIASCALE        # = -1/16
                EXTEND
                INDEX   KDPNDX
                MP      BRATE
                EXTEND
                INDEX   KSPNDX
                DV      1JACC           # =AJX  $ 90 DEG/SEC-SEC
                INDEX   KSPNDX
                TS      DELPEROR        #     $ 180 DEG
                CCS     KSPNDX
                TCF     RATEBIAS

                CA      TIME1
                AD      ONESEK +1
                XCH     NEXTIME
                TCF     INCRDCDU -1

ONESEK          DEC     0
                DEC     100

BIASCALE        OCT     75777           # = -1/16

CONTMANU        CS      TIME1           # RESET FOR NEXT DCDU UPDATE
                AD      NEXTIME
                CCS     A
                AD      ONE
                TCF     MANUCALL
                AD      NEGMAX
                COM
MANUCALL        INHINT                  # CALL FOR NEXT UPDATE VIA WAITLIST
                TC      WAITLIST
                EBANK=  TTEMP
                2CADR   UPDTCALL
```

```
                CAF     ONESEK +1       # INCREMENT TIME FOR NEXT UPDATE
                ADS     NEXTIME
                TCF     ENDOFJOB

UPDTCALL        CAF     PRIO26          # SATELLITE PROGRAM TO CALL FOR UPDATE
                TC      FINDVAC         # OF STEERING COMMANDS
                EBANK=  TTEMP
                2CADR   NEWDELHI

                TC      TASKOVER
```

```
# ROUTINE FOR TERMINATING AUTOMATIC MANEUVERS

MANUSTOP        CAF     ZERO            # ZERO MANEUVER RATES
                TS      DELDCDU2
                TS      OMEGARD
                TS      DELRERØR
                TS      DELDCDU1
                TS      OMEGAQD
                TS      DELQERØR
                CA      CPSI            # SET DESIRED GIMBAL ANGLES TO
                TS      CDUZD           # DESIRED FINAL GIMBAL ANGLES
                CA      CTHETA
                TS      CDUYD
ENDROLL         CA      CPHI            # NO FINAL YAW
                TS      CDUXD
                CAF     ZERO
                TS      OMEGAPD         # I.E. MANEUVER DID NOT GO THRU
                TS      DELDCDU         # GIMBAL LOCK ORIGINALLY
                TS      DELPERØR
GOODMANU        CA      ATTPRIO         # RESTORE USERS PRIO
                TS      NEWPRIO

                CA      ZERO            # ZERO ATTCADR
                DXCH    ATTCADR

                TC      SPVAC           # RETURN TO USER

                TC      TASKOVER
```

```
                EBANK=  XSM

                BANK    33
                SETLOC  E/PROG
                BANK

                COUNT*  $$/P07

# SPECIAL PROGRAMS TO EASE THE PANGS OF ERASABLE MEMORY PROGRAMS.
#
# E/BKCALL      FOR DOING BANKCALLS FROM AND RETURNING TO ERASABLE.
#
# THIS ROUTINE IS CALLABLE FROM ERASABLE OR FIXED.  LIKE BANKCALL, HOWEVER, SWITCHING BETWEEN S3 AND S4
# IS NOT POSSIBLE.
#
# THE CALLING SEQUENCE IS:
#
#       TC      BANKCALL
#       CADR    E/BKCALL
#       CADR    ROUTINE         # WHERE TO WANT TO GO IN FIXED.
#       RETURN HERE FROM DISPLAY TERMINATE, BAD STALL OR TC Q.
#       RETURN HERE FROM DISPLAY PROCEED OR GOOD RETURN FROM STALL.
#       RETURN HERE FROM DISPLAY ENTER OR RECYCLE.
#
# THIS ROUTINE REQUIRES TWO ERASABLES (EBUF2, +1) IN UNSWITCHED WHICH ARE UNSHARED BY INTERRUPTS AND
# OTHER EMEMORY PROGRAMS.
#
# A + L ARE PRESERVED THROUGH BANKCALL AND E/BKCALL.

E/BKCALL        DXCH    BUF2            # SAVE A,L AND GET DP RETURN.
                DXCH    EBUF2           # SAVE DP RETURN.
                INCR    EBUF2           # RETURN +1 BECAUSE DOUBLE CADR.
                CA      BBANK
                MASK    LOW10           # GET CURRENT EBANK.  (SBANK SOMEDAY)
                ADS     EBUF2   +1      # FORM BBCON.  (WAS FBANK)
                NDX     EBUF2
                CA      0       -1      # GET CADR OF ROUTINE.
                TC      SWCALL          # GO TO ROUTINE, SETTING Q TO SWRETURN
                                        # AND RESTORING A + L.
                TC      +4              # TX Q, V34, OR BAD STALL RETURN.
                TC      +2              # PROCEED OR GOOD STALL RETURN.
                INCR    EBUF2           # ENTER OR RECYCLE RETURN.
                INCR    EBUF2
E/SWITCH        DXCH    EBUF2
                DTCB
```

```
# E/CALL          FOR CALLING A FIXED MEMORY INTERPRETIVE SUBROUTINE FROM ERASABLE AND RETURNING TO ERASABLE.
#
# THE CALLING SEQUENCE IS...
#
#       RTB
#               E/CALL
#       CADR    ROUTINE                 # THE INTERPRETIVE SUBROUTINE YOU WANT.
#                                       # RETURNS HERE IN INTERPRETIVE.

E/CALL          LXCH    LOC             # ADRES -1 OF CADR.
                INDEX   L
                CA      L               # CADR IN A.
                INCR    L
                INCR    L               # RETURN ADRES IN L.
                DXCH    EBUF2           # STORE CADR AND RETURN
                TC      INTPRET
                CALL
                        EBUF2           # INDIRECTLY EXECUTE ROUTINE.  IT MUST
                EXIT                    # LEAVE VIA RVQ OR EQUIVALENT.
                LXCH    EBUF2   +1      # PICK UP RETURN.
                TCF     INTPRET +2      # SET LOC AND RETURN TO CALLER.
```

```
# E/JOBWAK      FOR WAKING UP ERASABLE MEMORY JOBS.
#
# THIS ROUTINE MUST BE CALLED IN INTERRUPT OR WITH INTERRUPTS INHIBITED.
#
# THE CALLING SEQUENCE IS:
#
#       INHINT
#       ...
#       CA      WAKEADR         # ADDRESS OF SLEEPING JOB
#       TC      IBNKCALL
#       CADR    E/JOBWAK
#       ...                     # RETURNS HERE
#       RELINT                  # IF YOU DID AND INHINT.

                BANK    33
                SETLOC  E/PROG
                BANK

                COUNT*   $$/P07

E/JOBWAK        TC      JOBWAKE         # ARRIVE IWTH ADRES IN A.
                CS      BIT11
                NDX     LOCCTR
                ADS     LOC             # KNOCK FIXED MEMORY BIT OUT OF ADRES.
                TC      RUPTREG3        # RETURN
```

```
# NAME --        IMU PERFORMANCE TESTS 2
#
# DATE --        MARCH 20, 1967
#
# BY --          SYSTEM TEST GROUP 864-6900 EXT. 1274
#
# MODNO. --      ZERO
#
# FUNCTIONAL DESCRIPTION
#
# POSITIONING ROUTINES FOR THE IMU PERFORMANCE TESTS AS WELL AS SOME OF
# THE TESTS THEMSELVES.  FOR A DESCRIPTION OF THESE SUBROUTINES AND THE
# OPERATING PROCEDURES (TYPICALLY) SEE STG MEMO 685.  THEORETICAL REF. E-1973

                BANK    33
                SETLOC  IMU2
                BANK
                EBANK=  POSITON
                COUNT*  $$/P07

REDO            TC      NEWMODEX
                MM      07

GEOIMUTT        TC      IMUZERR
IMUBACK         CA      ZERO
                TS      NDXCTR
                TS      TORQNDX
                TS      TORQNDX +1
                TS      OVFLOWCK
NBPOSPL         CA      DEC17
                TS      ZERONDX
                CA      XNBADR
                TC      ZEROING
                CA      HALF
                TS      XNB
GUESS           TC      INTPRET
LATAZCHK        DLOAD   SL2
                        LATITUDE
                STODL   DSPTEM1 +1
                        AZIMUTH
                RTB     EXIT
                        1STO2S
                XCH     MPAC
                TS      DSPTEM1
                CAF     VN0641
                TC      BANKCALL
                CADR    GOFLASH
                TC      ENDTEST1
                TC      +2
                TC      -5
```

```

                TC      INTPRET
                SLOAD   RTB
                        DSPTEM1
                        CDULOGIC
                STORE   AZIMUTH
                SLOAD   SR2
                        DSPTEM1 +1
                STORE   LATITUDE
                COS     DCOMP
                SL1
                STODL   WANGI
                        LATITUDE
                SIN     SL1
                STODL   WANGO
                        AZIMUTH
                PUSH    SIN
                STORE   YNB     +2
                STODL   ZNB     +4
                COS
                STORE   YNB     +4
                DCOMP
POSGMBL         STCALL  ZNB     +2
                        CALCGA
                EXIT
                TC      BANKCALL
                CADR    IMUCOARS
                CAF     BIT14           # IF BIT14 SET, GIMBAL LOCK
                MASK    FLAGWRD3
                EXTEND
                BZF     +2
                INCR    NDXCTR          # +1 IF IN GIMBAL LOCK, OTHERWISE 0
                TC      DOWNFLAG
                ADRES   GLOKFAIL        # RESET GIMBAL LOCK FLAG
                TC      IMUSLLLG
                CCS     NDXCTR          # IF ONE GO AND DO A PIPA TEST ONLY
                TC      PIPACHK         # ALIGN AND MEASURE VERTICAL PIPA RATE
                TC      FINIMUDD
                EXTEND
                DCA     PERFDLAY
                TC      LONGCALL        # DELAY WHILE SUSPENSION STABILIZES
                EBANK=  POSITON
                2CADR   GOESTIMS

                CA      ESTICADR
                TC      JOBSLEEP
GOESTIMS        CA      ESTICADR
                TC      JOBWAKE
                TC      TASKOVER
ESTICADR        CADR    ESTIMS
TORQUE          CA      ZERO
```

```
                TS      DSPTEM2
                CA      DRIFTI
                TS      DSPTEM2 +1
                INDEX   POSITON
                TS      SOUTHDR -1
                TC      SHOW

PIPACHK         INDEX   NDXCTR          # PIPA TEST
                TC      +1
                TC      EARTHR*
                CA      DEC17           # ALLOW PIP COUNTER TO OVERFLOW 17 TIMES
                TS      DATAPL  +4      # IN THE ALLOTTED TIME INTERVAL
                CA      DEC58
                TS      LENGTHOT
                CA      ONE
                TS      RESULTCT
                CA      ZERO
                INDEX   PIPINDEX
                TS      PIPAX
                TS      DATAPL
                TC      CHECKG
                INHINT
                CAF     TWO
                TC      TWIDDLE
                EBANK=  XSM
                ADRES   PIPATASK
                TC      ENDOFJOB

PIPATASK        EXTEND
                DIM     LENGTHOT
                CA      LENGTHOT
                EXTEND
                BZMF    STARTPIP
                CAF     BIT10
                TC      TWIDDLE
                EBANK=  XSM
                ADRES   PIPATASK
STARTPIP        CAF     PRIO20
                TC      FINDVAC
                EBANK=  XSM
                2CADR   PIPJOBB

                TC      TASKOVER

PIPJOBB         INDEX   NDXCTR
                TC      +1
                TC      EARTHR*
                CA      LENGTHOT
```

```
                EXTEND
                BZMF     +2
                TC       ENDOFJOB

                CA       FIVE
                TS       RESULTCT
                TC       CHECKG
                CCS      DATAPL  +1
                TC       +4
                TC       CCSHOLE
                CS       DATAPL  +4
                TS       DATAPL  +4
                EXTEND
                DCS      DATAPL
                DAS      DATAPL  +4

                TC       INTPRET
                DLOAD    DSU
                         DATAPL  +6
                         DATAPL  +2
                BPL      CALL
                         AINGOTN
                         OVERFFIX
AINGOTN         PDDL     DDV
                         DATAPL  +4
                DMPR     RTB
                         DEC585          # DEC585 HAS BEEN REDEFINED FOR LEM
                         SGNAGREE
                STORE    DSPTEM2
                EXIT
                CCS      NDXCTR
                TC       COAALIGN        # TAKE PLATFORM OUT OF GIMBAL LOCK
                TC       SHOW
VERTDRFT        CA       3990DEC         # ABOUT 1 HOUR VERTICAL DRIFT TEST
                TS       LENGTHOT
                INDEX    POSITON
                CS       SOUTHDR -2
                TS       DRIFTT
                CCS      PIPINDEX        # OFFSET PLATFORM TO MISS PIP DEAD-ZONES
                TCF      PON4            # Z-UP IN POS 4
PON2            CS       BIT5            # X-UP
                ADS      ERCOMP  +2
                CA       BIT5
                ADS      ERCOMP  +4
                TCF      PON
PON4            CS       BIT5
                ADS      ERCOMP  +2
                CA       BIT5
                ADS      ERCOMP
PON             TC       EARTHR*
```

```
                CA      ZERO            # ALLOW ONLY SOUTH GYRO EARTH RATE COMPENS
                TS      ERVECTOR
                TS      ERVECTOR +1
GUESS1          CAF     POSMAX
                TS      TORQNDX
                TS      TORQNDX +1
                CA      CDUX
                TS      LOSVEC
                TC      ESTIMS
VALMIS          CA      DRIFTO
                TS      DSPTEM2 +1
                CA      ZERO
                TS      DSPTEM2
                TC      SHOW

ENDTEST1        TC      DOWNFLAG
                ADRES   IMUSE
                CS      ZERO
                TC      NEWMODEA
                TC      ENDEXT
```

```
OVERFFIX        DAD     DAD
                        DPPOSMAX
                        ONEDPP
                RVQ

COAALIGN        EXTEND                  # COARSE ALIGN SUBROUTINE
                QXCH    ZERONDX
                CA      ZERO
                TS      THETAD
                TS      THETAD +1
                TS      THETAD +2
                TC      BANKCALL
                CADR    IMUCOARS
ALIGNCOA        TC      BANKCALL
                CADR    IMUSTALL
                TC      SOMERR2
                TC      ZERONDX

IMUSLLLG        EXTEND
                QXCH    ZERONDX
                TC      ALIGNCOA

FINIMUDD        EXTEND
                QXCH    ZERONDX
                TC      BANKCALL
                CADR    IMUFINE
                TC      ALIGNCOA

IMUZERR         EXTEND
                QXCH    ZERONDX
                TC      BANKCALL
                CADR    IMUZERO
                TC      ALIGNCOA

CHECKG          EXTEND                  # PIP PULSE CATCHING ROUTINE
                QXCH    QPLACE
                TC      +6
CHECKG1         RELINT
                CA      NEWJOB
                EXTEND
                BZMF    +6
                TC      CHANG1
                INHINT
                INDEX   PIPINDEX
                CS      PIPAX
                TS      ZERONDX
                INHINT
```

```
                INDEX   PIPINDEX
                CA      PIPAX
                AD      ZERONDX
                EXTEND
                BZF     CHECKG1
                INDEX   PIPINDEX
                CA      PIPAX
                INDEX   RESULTCT
                TS      DATAPL
                TC      FINETIME
                INDEX   RESULTCT
                TS      DATAPL +1
                INDEX   RESULTCT
                LXCH    DATAPL +2
                RELINT
ENDCHKG         TC      QPLACE

ZEROING         TS      L
                TCF     +2
ZEROING1        TS      ZERONDX
                CAF     ZERO
                INDEX   L
                TS      0
                INCR    L
                CCS     ZERONDX
                TCF     ZEROING1
                TC      Q
```

```
ERTHRVSE        DLOAD   PDDL
                        SCHZEROS        # PD24 = (SIN             -COS      0)(OMEG/MS)
                        LATITUDE
                COS     DCOMP
                PDDL    SIN
                        LATITUDE
                VDEF    VXSC
                        OMEG/MS
                STORE   ERVECTOR
                RTB
                        LOADTIME
                STOVL   TMARK
                        SCHZEROS
                STORE   ERCOMP
                RVQ

EARTHR          ITA     RTB
                        S2
                        LOADTIME
                STORE   TEMPTIME
                DSU     BPL
                        TMARK
                        ERTHR
                CALL
                        OVERFFIX
ERTHR           SL      VXSC
                        9D
                        ERVECTOR
                MXV     VAD
                        XSM
                        ERCOMP
                STODL   ERCOMP
                        TEMPTIME
                STORE   TMARK
                AXT,1   RTB
                ECADR   ERCOMP
                        PULSEIMU
                GOTO
                        S2

EARTHR*         EXTEND
                QXCH    QPLACES
                TC      INTPRET
                CALL
                        EARTHR
                EXIT
                TC      IMUSLLLG
                TC      QPLACES

SHOW            EXTEND
```

```
                QXCH    QPLACE
SHOW1           CA      POSITON
                TS      DSPTEM2 +2
                CA      VB06N98
                TC      BANKCALL
                CADR    GOFLASH
                TC      ENDTEST1        # V34
                TC      QPLACE          # V33
                TCF     SHOW1

3990DEC         DEC     3990
VB06N98         VN      0698
VN0641          VN      0641
DEC17           =       ND1
DEC58           DEC     58
OGCPL           ECADR   OGC
1SECX           =       1SEC
XNBADR          GENADR  XNB
XSMADR          GENADR  XSM
                BLOCK   2
                COUNT*  $$/P07
FINETIME        INHINT                  # RETURNS WITH INTERRUPT INHIBITED
                EXTEND
                READ    LOSCALAR
                TS      L
                EXTEND
                RXOR    LOSCALAR
                EXTEND
                BZF     +4
                EXTEND
                READ    LOSCALAR
                TS      L
 +4             CS      POSMAX
                AD      L
                EXTEND
                BZF     FINETIME +1
                EXTEND
                READ    HISCALAR
                TC Q
```

```
# PROGRAM --    IMU PERFORMANCE TESTS 4
# DATE --       NOV 15, 1966
# BY --         GEORGE SCHMIDT IL7-146 EXT 1126
# MOD NO-ZERO
#
# FUNCTIONAL DESCRIPTION
#
# THIS SECTION CONSISTS OF THE FILTER FOR THE GYRO DRIFT TESTS.  NO COMPASS
# IS DONE IN LEM.  FOR A DESCRIPTION OF THE FILTER SEE E-1973.  THIS
# SECTION IS ENTERED FROM IMU 2.  IT RETURNS THERE AT END OF TEST.
#
# EARTHR,OGC ZERO,ERTHRVSE
#
# NORMAL EXIT
#
# LENGTHOT GOES TO ZERO -- RETURN TO IMU PERF TESTS 2 CONTROL
#
# ALARMS
#
# 1600  OVERFLOW IN DRIFT TEST
# 1601  BAD IMU MODING IN ANY ROUTINE THAT USES IMUSTALL
#       OUTPUT
#
# FLASHING DISPLAY OF RESULTS -- CONTROLLED IN IMU PERF TESTS 2
#
# DEBRIS
#
# ALL CENTRALS -- ALL OF EBANK XSM
```

```
                BANK    33
                SETLOC  IMU4
                BANK
                COUNT*  $$/P07

                EBANK=  XSM

ESTIMS          INHINT
                CAE     1SECXT
                TC      TWIDDLE
                EBANK=  XSM
                ADRES   ALLOOP
                CAF     ZERO            # ZERO THE PIPAS
                TS      PIPAX
                TS      PIPAY
                TS      PIPAZ
                RELINT
                CA      77DECML
                TS      ZERONDX
                CA      ALXXXZ
                TC      ZEROING
                TC      INTPRET
                SLOAD
                        SCHZEROS
                STOVL   GCOMPSW -1
                        INTVAL  +2
                STOVL   ALX1S
                        SCHZEROS
                STORE   DELVX
                STORE   GCOMP
                SLOAD
                        TORQNDX
                DCOMP   BMN
                        VERTSKIP
                CALL
                        ERTHRVSE
VERTSKIP        EXIT
                TC      SLEEPIE +1
```

```
ALLOOP          CA      OVFLOWCK
                EXTEND
                BZF     +2
                TC      TASKOVER
                CCS     ALTIM
                CA      A               # SHOULD NEVER HIT THIS LOCATION
                TS      ALTIMS
                CS      A
                TS      ALTIM
                CS      ONE
                AD      GEOCOMPS
                EXTEND
                BZF     +4
                CA      LENGTHOT
                EXTEND
                BZMF    +5
                CAE     1SECXT
                TC      TWIDDLE
                EBANK=  XSM
                ADRES   ALLOOP
                CAF     ZERO
                XCH     PIPAX
                TS      DELVX
                CAF     ZERO
                XCH     PIPAY
                TS      DELVY
                CAF     ZERO
                XCH     PIPAZ
                TS      DELVZ
SPECSTS         CAF     PRIO20
                TC      FINDVAC
                EBANK=  XSM
                2CADR   ALFLT           # START THE JOB

                TC      TASKOVER
```

```
ALFLT           CCS     GEOCOMPS
                TC      +2
                TC      NORMLOP
                TC      BANKCALL
                CADR    1/PIPA
NORMLOP         TC      INTPRET
                DLOAD
                        INTVAL
                STOVL   S1
                        DELVX
                VXM     VSL1
                        XSM
                DLOAD   DCOMP
                        MPAC +3
                STODL   DPIPAY
                        MPAC +5
                STORE   DPIPAZ

                SETPD   AXT,1
                        0
                        8D
                SLOAD   DCOMP
                        GEOCOMPS
                BMN
                        PERFERAS
ALCGKK          SLOAD   BMN
                        ALTIMS
                        ALFLT3
ALKCG           AXT,2   LXA,1           # LOADS SLOPES AND TIME CONSTANTS AT RQST
                        12D
                        ALX1S
ALKCG2          DLOAD*  INCR,1
                        ALFDK   +144D,1
                DEC     -2
                STORE   ALDK    +10D,2
                TIX,2   SXA,1
                        ALKCG2
                        ALX1S

ALFLT3          AXT,1
                        8D
DELMLP          DLOAD*  DMP
                        DPIPAY  +8D,1
                        PIPASC
                SLR     BDSU*
                        9D
                        INTY    +8D,1
                STORE   INTY    +8D,1
                PDDL    DMP*
                        VELSC
```

```
                          VLAUN   +8D,1
                  SL2R
                  DSU     STADR
                  STORE   DELM    +8D,1
                  STORE   DELM    +10D,1
                  TIX,1   AXT,2
                          DELMLP
                          4
ALILP             DLOAD*  DMPR*
                          ALK     +4,2
                          ALDK    +4,2
                  STORE   ALK     +4,2
                  TIX,2   AXT,2
                          ALILP
                          8D
ALKLP             LXC,1   SXA,1
                          CMPX1
                          CMPX1
                  DLOAD*  DMPR*
                          ALK     +1,1
                          DELM    +8D,2
                  DAD*
                          INTY    +8D,2
                  STORE   INTY    +8D,2
                  DLOAD*  DAD*
                          ALK     +12D,2
                          ALDK    +12D,2
                  STORE   ALK     +12D,2
                  DMPR*   DAD*
                          DELM    +8D,2
                          INTY    +16D,2
                  STORE   INTY    +16D,2
                  DLOAD*  DMP*
                          ALSK    +1,1
                          DELM    +8D,2
                  SL1R    DAD*
                          VLAUN   +8D,2
                  STORE   VLAUN   +8D,2
                  TIX,2   AXT,1
                          ALKLP
                          8D

LOOSE             DLOAD*  PDDL*
                          ACCWD   +8D,1
                          VLAUN   +8D,1
                  PDDL*   VDEF
                          POSNV   +8D,1
                  MXV     VSL1
                          TRANSM1
```

```
                DLOAD
                        MPAC
                STORE   POSNV    +8D,1
                DLOAD
                        MPAC     +3
                STORE   VLAUN   +8D,1
                DLOAD
                        MPAC    +5
                STORE   ACCWD   +8D,1
                TIX,1
                        LOOSE

                AXT,2   AXT,1            # EVALUATE SINES AND COSINES
                        6
                        2
BOOP            DLOAD*  DMPR
                        ANGX    +2,1
                        GEORGEJ
                SR2R
                PUSH    SIN
                SL3R    XAD,1
                        X1
                STORE   16D,2
                DLOAD
                COS
                STORE   22D,2            # COSINES
                TIX,2
                        BOOP

PERFERAS        EXIT
                CA      EBANK7
                TS      EBANK
                EBANK=  ATIGINC
                TC      ATIGINC          # GOTO ERASABLE TO CALCULATE ONLY TO RETN

#                            CAUTION
#
# THE ERASABLE PROGRAM THAT DOES THE CALCULATIONS MUST BE LOADED
# BEFORE ANY ATTEMPT IS MAKE TO RUN THE IMU PERFORMANCE TEST

                EBANK=  AZIMUTH
                CCS     LENGTHOT
                TC      SLEEPIE
                CCS     TORQNDX
                TCF     +2
                TC      SETUPER1
                CA      CDUX
                TS      LOSVEC   +1      # FOR TROUBLESHOOTING VD POSNS 2$4
```

```
SETUPER1        TC      INTPRET
                DLOAD   PDDL            # ANGLES FROM DRIFT TEST ONLY
                        ANGZ
                        ANGY
                PDDL    VDEF
                        ANGX
                VCOMP   VXSC
                        GEORGEJ
                MXV     VSR1
                        XSM
                STORE   OGC
                EXIT

                CA      OGCPL
                TC      BANKCALL
                CADR    IMUPULSE
                TC      IMUSLLLG
GEOSTRT4        CCS     TORQNDX         # ONLY POSITIVE IF IN VERTICAL DRIFT TEST
                TC      VALMIS
                TC      INTPRET
                CALL
                        ERTHRVSE
                EXIT
                TC      TORQUE

SLEEPIE         TS      LENGTHOT        # TEST NOT OVER-DECREMENT LENGTHOT
                CCS     TORQNDX         # ARE WE DOING VERTDRIFT
                TC      EARTHR*
                TC      ENDOFJOB

SOMEERRR        CA      EBANK5
                TS      EBANK
                CA      ONE
                TS      OVFLOWCK        # STOP ALLOOP FROM CALLING ITSELF
                TC      ALARM
                OCT     1600
                TC      ENDTEST1
SOMERR2         CAF     OCT1601
                TC      VARALARM
                TC      DOWNFLAG
                ADRES   IMUSE
                TC      ENDOFJOB

OCT1601         OCT     01601
DEC585          OCT     06200           # 3200 B+14 ORDER IS IMPORTANT
SCHZEROS        2DEC    .00000000
```

```
                2DEC    .00000000

                OCT     00000
ONEDPP          OCT     00000           # ORDER IS IMPORTANT
                OCT     00001

INTVAL          OCT     4
                OCT     2
                DEC     144
                DEC     -1
SOUPLY          2DEC    .93505870       # INITIAL GAINS FOR PIP OUTPUTS

                2DEC    .26266423       # INITIAL GAINS/4 FOR ERECTION ANGLES

77DECML         DEC     77
ALXXXZ          GENADR  ALX1S   -1
PIPASC          2DEC    .13055869

VELSC           2DEC    -.52223476      # 512/980.402

ALSK            2DEC    .17329931       # SSWAY VEL GAIN X 980.402/4096

                2DEC    -.00835370      # SSWAY ACCEL GAIN X 980.402/4096

GEORGEJ         2DEC    .63661977

GEORGEK         2DEC    .59737013
```

```
# PROGRAM NAME -- KEYBOARD AND DISPLAY PROGRAM
# MOD NO -- 4            DATE -- 27 APRIL 1967            ASSEMBLY -- PINDANCE REV 18
# MOD BY -- FILENE
# LOG SECTION -- PINBALL GAME BUTTONS AND LIGHTS
#
# FUNCTIONAL DESCRIPTION
#
# THE KEYBOARD AND DISPLAY SYSTEM PROGRAM OPERATES UNDER EXECUTIVE
# CONTROL AND PROCESSES INFORMATION EXCHANGED BETWEEN THE AGC AND THE
# COMPUTER OPERATOR.  THE INPUTS TO THE PROGRAM ARE FROM THE KEYBOARD,
# FROM INTERNAL PROGRAM, AND FROM THE UPLINK.
#
# THE LANGUAGE OF COMMUNICATION WITH THE PROGRAM IS A PAIR OF WORDS
# KNOWN AS VERB AND NOUN.  EACH OF THESE IS REPRESENTED BY A 2 CHARACTER
# DECIMAL NUMBER.  THE VERB CODE INDICATES WHAT ACTION IS TO BE TAKEN, THE
# NOUN CODE INDICATES TO WHAT THIS ACTION IS APPLIED.  NOUNS USUALLY
# REFER TO A GROUP OF ERASABLE REGISTERS.
#
# VERBS ARE GROUPED INTO DISPLAYS, LOADS, MONITORS (DISPLAYS THAT ARE
# UPDATED ONCE PER SECOND), SPECIAL FUNCTIONS, AND EXTENDED VERBS (THESE
# ARE OUTSIDE OF THE DOMAIN OF PINBALL AND CAN BE FOUND UNDER LOG SECTION
# 'EXTENDED VERBS').
#
# A LIST OF VERBS AND NOUNS IS GIVEN IN LOG SECTION 'ASSEMBLY AND
# OPERATION INFORMATION'.
#
#
# CALLING SEQUENCES --
#
# KEYBOARD:
# EACH DEPRESSION OF A KEYBOARD BUTTON ACTIVATES AN INTERRUPT KEYRUPT1
# AND PLACES THE 5 BIT KEY CODE INTO CHANNEL 15.  KEYRUPT1 PLACES THE KEY
# CODE INTO MPAC, ENTERS AN EXECUTIVE REQUEST FOR THE KEYBOARD AND DISPLAY
# PROGRAM (AT 'CHARIN'), AND EXECUTES A RESUME.
#
# UPLINK:
# EACH WORD RECEIVED BY THE UPLINK ACTIVATES INTERRUPT UPRUPT, WHICH
# PLACES THE 5 BIT KEY CODE INTO MPAC, ENTERS AN EXECUTIVE REQUEST FOR THE
# KEYBOARD AND DISPLAY PROGRAM (AT 'CHARIN') AND EXECUTES A RESUME.
#
# INTERNAL PROGRAMS:
# INTERNAL PROGRAMS CALL PINBALL AT `NVSUB' WITH THE DESIRED VERB/NOUN
# CODE IN A (LOW 7 BITS FOR NOUN, NEXT 7 BITS FOR VERB).  DETAILS
# DESCRIBED ON REMARKS CARDS JUST BEFORE 'NVSUB' AND 'NVSBWAIT' (SEE
# SYMBOL TABLE FOR PAGE NUMBERS).
#
# NORMAL EXIT MODES --
#
#       IF PINBALL WAS CALLED BY EXTERNAL ACTION, THERE ARE FOUR EXITS:
#               1)      ALL BUT (2), (3), AND (4) EXIT DIRECTLY TO ENDOFJOB.
#               2)      EXTENDED VERBS GO TO THE EXTENDED VERB FAN AS PART OF THE
```

```
#                       PINBALL EXECUTIVE JOB WITH PRIORITY 30000.  IT IS THE
#                       RESPONSIBILITY OF THE EXTEDED VERB CALLED TO EVENTUALLY
#                       CHANGE PRIORITY (IF NECESSARY) AND DO AN ENDOFJOB.
#                       ALSO PINBALL IS A NOVAC JOB.  EBANK SET FOR COMMON.
#               3)      VERB 37.  CHANGE OF PROGRAM (MAJOR MODE) CALLS 'V37' IN THE
#                       SERVICE ROUTINES AS PART OF THE PINBALL EXEC JOB WITH PRIO
#                       30000.  THE NEW PROGRAM CODE (MAJOR MODE) IS LEFT IN A.
#               4)      KEY RELEASE BUTTON CALLS 'PINBRNCH' IN THE DISPLAY INTERFACE
#                       ROUTINES AS PART OF THE PINBALL EXEC JOB WITH PRIO 30000 IF
#                       THE KEY RELEASE LIGHT IS OFF AND 'CADRSTOR' IS NOT +0.
#
# IF PINBALL WAS CALLED BY INTERNAL PROGRAMS, EXIT FROM PINBALL IS BACK
# TO CALLING ROUTINE.  DETAILS DESCRIBED IN REMARKS CARDS JUST BEFORE
# 'NVSUB' AND 'NVSBWAIT' (SEE SYMBOL TABLE FOR PAGE NUMBERS).
#
# ALARM OR ABORT EXIT MODES --
#
#       EXTERNAL INITIATION:
#       IF SOME IMPROPER SEQUENCE OF KEY CODES IS DETECTED, THE OPERATOR
#       ERROR LIGHT IS TURNED ON AND EXIT IS TO 'ENDOFJOB'.
#
#       INTERNAL PROGRAM INITIATION:
#       IF AN ILLEGAL V/N COMBINATION IS ATTEMPTED, AN ABORT IS CAUSED
#       (WITH OCTAL 01501).
#       IF A SECOND ATTEMPT IS MADE TO GO TO SLEEP IN PINBALL, AN ABORT IS
#       CAUSED (WITH OCTAL 01206). THERE ARE TWO WAYS TO GO TO SLEEP IN PINBALL:
#               1) ENDIDLE OR DATAWAIT.
#               2) NVSBWAIT, PRENVBSY, OR NVSUBUSY.
#
# CONDITIONS LEADING TO THE ABOVE ARE DESCRIBED IN FORTHCOMING MIT/IL
# E-REPORT DESCRIBING KEYBOARD AND DISPLAY OPERATION FOR 278.
#
# OUTPUT --
#
# INFORMATION TO BE SENT TO THE DISPLAY PANEL IS LEFT IN THE 'DSPTAB'
# BUFFERS REGISTERS (UNDER EXEC CONTROL). 'DSPOUT' (A PART OF T4RUPT)
# HANDLES THE PLACING OF THE 'DSPTAB' INFORMATION INTO OUTPUT CHANNEL 10
# IN INTERRUPT.
#
# ERASABLE INITIALIZATION --
#
# FRESH START AND RESTART INITIALIZE THE NECESSARY E REGISTERS FOR
# PINBALL IN 'STARTSUB'.  REGISTERS ARE:  DSPTAB BUFFER, CADRSTOR,
# REQRET, CLPASS, DSPLOCK, MONSAVE, MONSAVE1, VERBREG, NOUNREG, DSPLIST,
# DSPCOUNT, NOUT.
#
# A COMPLETE LIST OF ALL THE ERASABLES (BOTH RESERVED AND TEMPORARIES) FOR
```

```
# PINBALL IS GIVEN BELOW.
#
# THE FOLLOWING ARE OF GENERAL INTEREST --
#
# REMARKS CARDS PRECEDE THE REFERENCED SYMBOL DEFINITION.  SEE SYMBOL
# TABLE TO FIND APPROPRIATE PACE NUMBERS.
#
#       NVSUB           CALLING POINT FOR INTERNAL USE OF PINBALL.
#                       OF RELATED INTEREST     NVSBWAIT
#                                               NVSUBUSY
#                                               PRENVBSY
#
#       ENDIDLE         ROUTINE FOR INTERNAL PROGRAMS WISHING TO TO SLEEP WHILE
#                       AWAITING OPERATORS RESPONSE.
#
#       DSPMM           ROUTINE BY WHICH AN INTERNAL PROGRAM MAY DISPLAY A DECIMAL
#                       PROGRAM CODE (MAJOR MODE) IN THE PROGRAM (MAJOR MODE) LIGHTS.
#                       (DSPMM DOES NOT DISPLAY DIRECTLY BUT ENTERS EXEC REQUEST
#                       FOR DSPMMJB WITH PRIO 30000 AND RETURNS TO CALLER.)
#
#       BLANKSUB        ROUTINE BY WHICH AN INTERNAL PROGRAM MAY BLANK ANY
#                       COMBINATION OF THE DISPLAY REGISTERS R1, R2, R3.
#
#       JAMTERM         ROUTINE BY WHICH AN INTERNAL PROGRAM MAY PERFORM THE
#       JAMPROC         TERMINATE (V 34) OR PROCEED (V 33) FUNCTION.
#
#       MONITOR         VERBS FOR PERIODIC (1 PER SEC) DISPLAY.
#
#       PLEASE PERFORM, PLEASE MARK SITUATIONS
#               REMARKS DESCRIBING HOW AN INTERNAL ROUTINE SHOULD HANDLE
#               THESE SITUATIONS CAN BE FOUND JUST BEFORE 'NVSUB' (SEE
#               SYMBOL TABLE FOR PAGE NUMBER).
#
#       THE NOUN TABLE FORMAT IS DESCRIBED ON A PAGE OF REMARKS CARDS JUST
#       BEFORE 'DSPABC' (SEE SYMBOL TABLE FOR PAGE NUMBER).
#
#       THE NOUN TABLES THEMSELVES ARE FOUND IN LOG SECTION 'PINBALL NOUN
#       TABLES'.
#
# FOR FURTHER DETAILS ABOUT OPERATION OF THE KEYBOARD AND DISPLAY SYSTEM
# PROGRAM, SEE THE MISSION PLAN AND/OR MIT/IL E-2129
# DESCRIBING KEYBOARD AND DISPLAY OPERATION FOR 278.


# THE FOLLOWING QUOTATION IS PROVIDED THROUGH THE COURTESY OF THE AUTHORS.
#
# "IT WILL BE PROVED TO THY FACE THAT THOU HAST MEN ABOUT THEE THAT
# USUALLY TALK OF A NOUN AND A VERB, AND SUCH ABOMINABLE WORDS AS NO
```

```
# CHRISTIAN EAR CAN ENDURE TO HEAR."
#                                       HENRY 6, ACT 2, SCENE 4


# THE FOLLOWING ASSIGNMENTS FOR PINBALL ARE MADE ELSEWHERE
#
# DSPCOUNT      ERASE                   # DISPLAY POSITION INDICATOR
# DECBRNCH      ERASE                   # +DEC, -DEC, OCT INDICATOR
# VERBREG       ERASE                   # VERB CODE
# NOUNREG       ERASE                   # NOUN CODE
# XREG          ERASE                   # R1 INPUT BUFFER
# YREG          ERASE                   # R2 INPUT BUFFER
# ZREG          ERASE                   # R3 INPUT BUFFER
# XREGLP        ERASE                   # LO PART OF XREG (FOR DEC CONV ONLY)
# YREGLP        ERASE                   # LO PART OF YREG (FOR DEC CONV ONLY)
# HITEMOUT      =       YREGLP          # TEMP FOR DISPLAY OF HRS, MIN, SEC
#                                       #       MUST = LOTEMOUT-1.
# ZREGLP        ERASE                   # LO PART OF ZREG (FOR DEC CONV ONLY)
# LOTEMOUT      =       ZREGLP          # TEMP FOR DISPLAY OF HRS, MIN, SEC
#                                       #       MUST = HITEMOUT+1.
# MODREG        ERASE                   # MODE CODE
# DSPLOCK       ERASE                   # KEYBOARD/SUBROUTINE CALL INTERLOCK
# REQRET        ERASE                   # RETURN REGISTER FOR LOAD
# LOADSTAT      ERASE                   # STATUS INDICATOR FOR LOADTST
# CLPASS        ERASE                   # PASS INDICATOR FOR CLEAR
# NOUT          ERASE                   # ACTIVITY COUNTER FOR DSPTAB
# NOUNCADR      ERASE                   # MACHINE CADR FOR NOUN
# MONSAVE       ERASE                   # N/V CODE FOR MONITOR. (= MONSAVE1-1)
# MONSAVE1      ERASE                   # NOUNCADR FOR MONITOR (MATBS) = MONSAVE+1
# MONSAVE2      ERASE                   # NVMONOPT OPTIONS
# DSPTAB        ERASE           +13D    # 0-10, DISPLAY PANEL BUFFER, 11-13, C RELAYS
# CADRSTOR      ERASE                   # ENDIDLE STORAGE
# NVQTEM        ERASE                   # NVSUB STORAGE FOR CALLING ADDRESS
#                                       # MUST = NVBNKTEM-1.
# NVBNKTEM      ERASE                   # NVSUB STORAGE FOR CALLING BANK
#                                       # MUST = NVQTEM+1
# VERBSAVE      ERASE                   # NEEDED FOR RECYCLE
# DSPLIST       ERASE                   # WAITING REG FOR DSP SYST INTERNAL USE
# EXTVBACT      ERASE                   # EXTENDED VERB ACTIVITY INTERLOCK
# DSPTEM1       ERASE                   # BUFFER STORAGE AREA 1 (MOSTLY FOR TIME)
# DSPTEM2       ERASE                   # BUFFER STORAGE AREA 2 (MOSTLY FOR DEG)
#
# END OF ERASABLES RESERVED FOR PINBALL EXECUTIVE ACTION
#
# TEMPORARIES FOR PINBALL EXECUTIVE ACTION
```

```
# DSEXIT         =       INTB15+          # RETURN FOR DSPIN
# EXITEM         =       INTB15+          # RETURN FOR SCALE FACTOR ROUTINE SELECT
# BLANKRET       =       INTB15+          # RETURN FOR 2BLANK

# WRDRET         =       INTBIT15         # RETURN FOR 5BLANK.
# WDRET          =       INTBIT15         # RETURN FOR DSPWD
# DECRET         =       INTBIT15         # RETURN FOR PUTCOM(DEC LOAD)
# 21/22REG       =       INTBIT15         # TEMP FOR CHARIN

# UPDATRET       =       POLISH           # RETURN FOR UPDATNN, UPDATVB
# CHAR           =       POLISH           # TEMP FOR CHARIN
# ERCNT          =       POLISH           # COUNTER FOR ERROR LIGHT RESET
# DECOUNT        =       POLISH           # COUNTER FOR SCALING AND DISPLAY (DEC)

# SGNON          =       VBUF             # TEMP FOR +,- ON
# NOUNTEM        =       VBUF             # COUNTER FOR MIXNOUN FETCH
# DISTEM         =       VBUF             # COUNTER FOR OCTAL DISPLAY VERB
# DECTEM         =       VBUF             # COUNTER FOR FETCH (DEC DISPLAY VERBS)

# SGNOFF         =       VBUF    +1       # TEMP FOR +,- ON
# NVTEMP         =       VBUF    +1       # TEMP FOR NVSUB
# SFTEMP1        =       VBUF    +1       # STORAGE FOR SF CONST HI PART (=SFTEMP2-1)
# HITEMIN        =       VBUF    +1       # TEMP FOR LOAD OF HRS, MIN, SEC
#                                         #       MUST = LOTEMIN-1.
# CODE           =       VBUF    +2       # FOR DSPIN
# SFTEMP2        =       VBUF    +2       # STORAGE FOR SF CONST LO PART (=SFTEMP1+1)
# LOTEMIN        =       VBUF    +2       # TEMP FOR LOAD OF HRS, MIN, SEC
#                                         #       MUST = HITEMIN+1
# MIXTEMP        =       VBUF    +3       # FOR MIXNOUN DATA
# SIGNRET        =       VBUF    +3       # RETURN FOR +,- ON
# ALSO MIXTEMP+1 = VBUF+4, MIXTEMP+2 = VBUF+5.

# ENTRET         =       DOTINC           # EXIT FROM ENTER

# WDCNT          =       DOTRET           # CHAR COUNTER FOR DSPWD
# INREL          =       DOTRET           # INPUT BUFFER SELECTOR (X, Y, Z, REG)

# DSPMMTEM       =       MATINC           # DSPCOUNT SAVE FOR DSPMM
# MIXBR          =       MATINC           # INDICATOR FOR MIXED OR NORMAL NOUN

# TEM1           ERASE                    # EXEC TEMP
# DSREL          =       TEM1             # REL ADDRESS FOR DSPIN

# TEM2           ERASE                    # EXEC TEMP
# DSMAG          =       TEM2             # MAGNITUDE STORE FOR DSPIN
# IDADDTEM       =       TEM2             # MIXNOUN INDIRECT ADDRESS STORAGE
# TEM3           ERASE                    # EXEC TEMP
# COUNT          =       TEM3             # FOR DSPIN
```

```
# TEM4           ERASE                       # EXEC TEMP
# LSTPTR         =       TEM4                # LIST POINTER FOR GRABUSY
# RELRET         =       TEM4                # RETURN FOR RELDSP
# FREERET        =       TEM4                # RETURN FOR FREEDSP
# DSPWDRET       =       TEM4                # RETURN FOR DSPSIGN
# SEPSCRET       =       TEM4                # RETURN FOR SEPSEC
# SEPMNRET       =       TEM4                # RETURN FOR SEPMIN

# TEM5           ERASE                       # EXEC TEMP
# NOUNADD        =       TEM5                # TEMP STORAGE FOR NOUN ADDRESS

# NNADTEM        ERASE                       # TEMP FOR NOUN ADDRESS TABLE ENTRY
# NNTYPTEM       ERASE                       # TEMP FOR NOUN TYPE TABLE ENTRY
# IDAD1TEM       ERASE                       # TEMP FOR INDIR ADDRESS TABLE ENTRY (MIXNN)
#                                            # MUST = IDAD2TEM-1, = IDAD3TEM-2.
# IDAD2TEM       ERASE                       # TEMP FOR INDIR ADDRESS TABLE ENTRY (MIXNN)
#                                            # MUST = IDAD1TEM+1, IDAD3TEM-1.
# IDAD3TEM       ERASE                       # TEMP FOR INDIR ADDRESS TABLE ENTRY (MIXNN)
#                                            # MUST = IDAD1TEM+2, IDAD2TEM+1.
# RUTMXTEM       ERASE                       # TEMP FOR SF ROUT TABLE ENTRY (MIXNN ONLY)
#
# END OF TEMPORARIES FOR PINBALL EXECUTIVE ACTION.
#
# ADDITIONAL TEMPORARIES FOR PINBALL EXECUTIVE ACTION
#
#       MPAC, THRU MPAC +6
#       BUF, +1, +2
#       BUF2, +1, +2
#       MPTEMP
#       ADDRWD
#
# END OF ADDITIONAL TEMPS FOR PINBALL EXEC ACTION
#
# RESERVED FOR PINBALL INTERRUPT ACTION
#
# DSPCNT         ERASE                       # COUNTER FOR DSPOUT
# UPLOCK         ERASE                       # BIT1 = UPLINK INTERLOCK (ACTIVATED BY
#                                            # RECEPTION OF A BAD MESSAGE IN UPLINK)
#
# END OF ERASABLES RESERVED FOR PINBALL INTERRUPT ACTION
#
# TEMPORARIES FOR PINBALL INTERRUPT ACTION
#
# KEYTEMP1       =       WAITEXIT            # TEMP FOR KEYRUPT, UPRUPT
# DSRUPTEM       =       WAITEXIT            # TEMP FOR DSPOUT
# KEYTEMP2       =       RUPTAGN             # TEMP FOR KEYRUPT, UPRUPT
#
# END OF TEMPORARIES FOR PINBALL INTERRUPT ACTION
```

```
# THE INPUT CODES ASSUMED FOR THE KEYBOARD ARE,
# 0              10000
# 1              00001
# 9              01001
# VERB           10001
# ERROR RES      10010
# KEY RLSE       11001
# +              11010
# -              11011
# ENTER          11100
# CLEAR          11110
# NOUN           11111
#
# (2003 RSB -- THE PROCEED KEY HAS NO KEYCODE; IT IS READ BY AN ALTERNATE MECHANISM.)
#
# OUTPUT FORMAT FOR DISPLAY PANEL.  SET OUT0 TO AAAABCCCCCDDDDD.
# A'S   SELECTS A RELAYWORD. THIS DETERMINES WHICH PAIR OF CHARACTERS ARE
#       ENERGIZED.
# B     FOR SPECIAL RELAYS SUCH AS SIGNS ETC.
# C'S   5 BIT RELAY CODE FOR LEFT CHAR OF PAIR SELECTED BY RELAYWORD.
# D'S   5 BIT RELAY CODE FOR RIGHT CHAR OF PAIR SELECTED BY RELAYWORD.
#
# THE PANEL APPEARS AS FOLLOWS,
# MD1   MD2                                 (MAJOR MODE)
# VD1   VD2  (VERB)       ND1      ND2      (NOUN)
# R1D1  R1D2     R1D3     R1D4     R1D5     (R1)
# R2D1  R2D2     R2D3     R2D4     R2D5     (R2)
# R3D1  R3D2     R3D3     R3D4     R3D5     (R3)
#
# EACH OF THESE IS GIVEN A DSPCOUNT NUMBER FOR USE WITHIN COMPUTATION ONLY
#
# MD1   25                R2D1     11               ALL ARE OCTAL
# MD2   24                R2D2     10
# VD1   23                R2D3      7
# VD2   22                R2D4      6
# ND1   21                R2D5      5
# ND2   20                R3D1      4
# R1D1  16                R3D2      3
# R1D2  15                R3D3      2
# R1D3  14                R3D4      1
# R1D4  13                R3D5      0
# R1D5  12
#
# THERE IS AN 11-REGISTER TABLE (DSPTAB) FOR THE DISPLAY PANEL.
#
# DSPTAB                  RELAYWD         BIT11           BITS 10-6       BITS 5-1
# RELADD
# 10             1011                             MD1 (25)        MD2 (24)
# 9              1010                             VD1 (23)        VD2 (22)
# 8              1001                             ND1 (21)        ND2 (20)
# 7              1000                                             R1D1 (16)
```

```
# 6             0111            +R1             R1D2 (15)       R1D3 (14)
# 5             0110            -R1             R1D4 (13)       R1D5 (12)
# 4             0101            +R2             R2D1 (11)       R2D2 (10)
# 3             0100            -R2             R2D3 (7)        R2D4 (6)
# 2             0011                            R2D5 (5)        R3D1 (4)
# 1             0010            +R3             R3D2 (3)        R3D3 (2)
# 0             0001            -R3             R3D4 (1)        R3D5 (0)
#               0000        NO RELAYWORD
#
# THE 5-BIT OUTPUT RELAY CODES ARE:
#
# BLANK 00000
# 0     10101
# 1     00011
# 2     11001
# 3     11011
# 4     01111
# 5     11110
# 6     11100
# 7     10011
# 8     11101
# 9     11111
#
# OUTPUT BITS USED BY PINBALL:
#
#       KEY RELEASE LIGHT       -       BIT 5 OF CHANNEL 11
#       VERB/NOUN FLASH         -       BIT 6 OF CHANNEL 11
#       OPERATOR ERROR LIGHT    -       BIT 7 OF CHANNEL 11
```

```
# START OF EXECUTIVE SECTION OF PINBALL

                BANK    40
                SETLOC  PINBALL1
                BANK

                COUNT*  $$/PIN
CHARIN          CAF     ONE             # BLOCK DISPLAY SYST
                XCH     DSPLOCK         # MAKE DSP SYST BUSY, BUT SAVE OLD
                TS      21/22REG        # C(DSPLOCK) FOR ERROR LIGHT RESET.
                CCS     CADRSTOR        # ALL KEYS EXCEPT ER TURN ON KR LITE IF
                TC      +2              # CADRSTOR IS FULL.  THIS REMINDS OPERATOR
                TC      CHARIN2         # TO RE-ESTABLISH A FLASHING DISPLAY
                CS      ELRCODE1        # WHICH HE HAS OBSCURED WITH DISPLAYS OF
                AD      MPAC            # HIS OWN (SEE REMARKS PRECEDING ROUTINE
                EXTEND                  # VBRELDSP).
                BZF     CHARIN2
                TC      RELDSPON
CHARIN2         XCH     MPAC
                TS      CHAR
                INDEX   A
                TC      +1              # INPUT CODE    FUNCTION
                TC      CHARALRM        # 0
                TC      NUM             # 1
                TC      NUM             # 2
                TC      NUM             # 3
                TC      NUM             # 4
                TC      NUM             # 5
                TC      NUM             # 6
                TC      NUM             # 7
                TC      89TEST          # 10            8
                TC      89TEST          # 11            9
                TC      CHARALRM        # 12
                TC      CHARALRM        # 13
                TC      CHARALRM        # 14
                TC      CHARALRM        # 15
                TC      CHARALRM        # 16
                TC      CHARALRM        # 17
                TC      NUM     -2      # 20            0
                TC      VERB            # 21            VERB
                TC      ERROR           # 22            ERROR LIGHT RESET
                TC      CHARALRM        # 23
                TC      CHARALRM        # 24
                TC      CHARALRM        # 25
                TC      CHARALRM        # 26
                TC      CHARALRM        # 27
                TC      CHARALRM        # 30
                TC      VBRELDSP        # 31            KEY RELEASE
                TC      POSGN           # 32            +
```

```
                TC      NEGSGN          # 33            -
                TC      ENTERJMP        # 34            ENTER
                TC      CHARALRM        # 35
                TC      CLEAR           # 36            CLEAR
                TC      NOUN            # 37            NOUN

ELRCODE1        OCT     22
ENTERJMP        TC      POSTJUMP
                CADR    ENTER

89TEST          CCS     DSPCOUNT
                TC      +4              # +
                TC      +3              # +0
                TC      ENDOFJOB        # - BLOCK DATA IN IF DSPCOUNT IS - OR -0
                TC      ENDOFJOB        # -0
                CAF     THREE
                MASK    DECBRNCH
                CCS     A
                TC      NUM             # IF DECBRNCH IS +, 8 OR 9 OK
                TC      CHARALRM        # IF DECBRNCH IS +0, REJECT 8 OR 9

# NUM ASSEMBLES OCTAL 3 BITS AT A TIME.  FOR DECIMAL IT CONVERTS INCOMING
# WORD AS A FRACTION, KEEPING RESULTS TO DP.
# OCTAL RESULTS ARE LEFT IN XREG, YREG, OR ZREG.  HI PART OF DEC IN XREG,
# YREG, ZREG.  THE LOW PARTS IN XREGLP, YREGLP, OR ZREGLP.
# DECBRNCH IS LEFT AT +0 FOR OCT, +1 FOR + DEC, +2 FOR - DEC.
# IF DSPCOUNT WAS LEFT -, NO MORE DATA IS ACCEPTED.

                CAF     ZERO
                TS      CHAR
NUM             CCS     DSPCOUNT
                TC      +4              # +
                TC      +3              # +0
                TC      +1              # -BLOCK DATA IN IF DSPCOUNT IS -
                TC      ENDOFJOB        # -0
                TC      GETINREL
                CCS     CLPASS          # IF CLPASS IS + OR +0, MAKE IT +0.
                CAF     ZERO
                TS      CLPASS
                TC      +1
                INDEX   CHAR
                CAF     RELTAB
                MASK    LOW5
                TS      CODE
                CA      DSPCOUNT
                TS      COUNT
                TC      DSPIN
                CAF     THREE
```

```
                MASK    DECBRNCH
                CCS     A               # +0, OCTAL.  +1, + DEC.  +2, - DEC.
                TC      DECTOBIN        # +
                INDEX   INREL           # +0 OCTAL
                XCH     VERBREG
                TS      CYL
                CS      CYL
                CS      CYL
                XCH     CYL
                AD      CHAR
                TC      ENDNMTST
DECTOBIN        INDEX   INREL
                XCH     VERBREG
                TS      MPAC            # SUM X 2EXP-14 IN MPAC
                CAF     ZERO
                TS      MPAC +1
                CAF     TEN             # 10 X 2EXP-14
                TC      SHORTMP         # 10SUM X 2EXP-28 IN MPAC, MPAC+1
                XCH     MPAC +1
                AD      CHAR
                TS      MPAC +1
                TC      ENDNMTST        # NO OF
                ADS     MPAC            # OF MUST BE 5TH CHAR
                TC      DECEND
ENDNMTST        INDEX   INREL
                TS      VERBREG
                CS      DSPCOUNT
                INDEX   INREL
                AD      CRITCON
                EXTEND
                BZF     ENDNUM          # -0, DSPCOUNT = CRITCON
                TC      MORNUM          # - , DSPCOUNT G/ CRITCON
ENDNUM          CAF     THREE
                MASK    DECBRNCH
                CCS     A
                TC      DECEND
ENDALL          CS      DSPCOUNT        # BLOCK NUMIN BY PLACING DSPCOUNT
                TC      MORNUM +1       # NEGATIVELY
DECEND          CS      ONE
                AD      INREL
                EXTEND
                BZMF    ENDALL          # IF INREL=0,1 (VBREG,NNREG) LEAVE WHOLE
                TC      DMP             # IF INREL=2,3,4 (R1,R2,R3), CONVERT TO FRAC
                                        # MULT SUM X 2EXP-28 IN MPAC, MPAC+1 BY
                ADRES   DECON           # 2EXP14/10EXP5, GIVES (SUM/10EXP5)X2EXP-14
                CAF     THREE           # IN MPAC, +1, +2.
                MASK    DECBRNCH
                INDEX   A
                TC      +0
                TC      +DECSGN
```

```
                EXTEND                  # - CASE
                DCS     MPAC +1
                DXCH    MPAC +1
+DECSGN         XCH     MPAC +2
                INDEX   INREL
                TS      XREGLP -2
                XCH     MPAC +1
                INDEX   INREL
                TS      VERBREG
                TC      ENDALL
MORNUM          CCS     DSPCOUNT        # DECREMENT DSPCOUNT
                TS      DSPCOUNT
                TC      ENDOFJOB

CRITCON         OCT     22              # (DEC 18)
                OCT     20              # (DEC 16)
                OCT     12              # (DEC 10)
                OCT     5
                OCT     0

DECON           2DEC    1 E-5 B14       # 2EXP14/10EXP5 = .16384 DEC

# GETINREL GETS PROPER DATA REG REL ADDRESS FOR CURRENT C(DSPCOUNT) AND
# PUTS IN INTO INREL. +0 VERBREG, 1 NOUNREG, 2 XREG, 3 YREG, 4 ZREG.

GETINREL        INDEX   DSPCOUNT
                CAF     INRELTAB
                TS      INREL           # (A TEMP. REG)
                TC      Q

INRELTAB        OCT     4               # R3D5 (DSPCOUNT = 0)
                OCT     4               # R3D4           =(1)
                OCT     4               # R3D3           =(2)
                OCT     4               # R3D2           =(3)
                OCT     4               # R3D1           =(4)
                OCT     3               # R2D5           =(5)
                OCT     3               # R2D4           =(6)
                OCT     3               # R2D3           =(7)
                OCT     3               # R2D2           =(8D)
                OCT     3               # R2D1           =(9D)
                OCT     2               # R1D5           =(10D)
                OCT     2               # R1D4           =(11D)
                OCT     2               # R1D3           =(12D)
                OCT     2               # R1D2           =(13D)
                OCT     2               # R1D1           =(14D)
                TC      CCSHOLE         # NO DSPCOUNT NUMBER = 15D
                OCT     1               # ND2            =(16D)
                OCT     1               # ND1            =(17D)
```

```
                OCT     0               # VD2           =(18D)
                OCT     0               # VD1           =(19D)

VERB            CAF     ZERO
                TS      VERBREG
                CAF     VD1
NVCOM           TS      DSPCOUNT
                TC      2BLANK
                CAF     ONE
                TS      DECBRNCH        # SET FOR DEC V/N CODE
                CAF     ZERO
                TS      REQRET          # SET FOR ENTPASO
                CAF     ENDINST         # IF DSPALARM OCCURS BEFORE FIRST ENTPASO
                TS      ENTRET          # OR NVSUB, ENTRET MUST ALREADY BE SET
                                        # TO TC ENDOFJOB
                TC      ENDOFJOB
NOUN            CAF     ZERO
                TS      NOUNREG
                CAF     ND1             # ND1, OCT 21 (DEC 17)
                TC      NVCOM

NEGSGN          TC      SIGNTEST
                TC      -ON
                CAF     TWO
BOTHSGN         INDEX   INREL           # SET DEC COMP BIT TO 1 (IN DECBRNCH)
                AD      BIT7            # BIT 5 FOR R1.  BIT 4 FOR R2.
                ADS     DECBRNCH        # BIT 3 FOR R3.
FIXCLPAS        CCS     CLPASS          # IF CLPASS IS + OR +0. MAKE IT +0.
                CAF     ZERO
                TS      CLPASS
                TC      +1
                TC      ENDOFJOB

POSGN           TC      SIGNTEST
                TC      +ON
                CAF     ONE
                TC      BOTHSGN

+ON             LXCH    Q
                TC      GETINREL
                INDEX   INREL
                CAF     SGNTAB -2
                TS      SGNOFF
                AD      ONE
                TS      SGNON
SGNCOM          CAF     ZERO
                TS      CODE
                XCH     SGNOFF
```

```
                TC      11DSPIN
                CAF     BIT11
                TS      CODE
                XCH     SGNON
                TC      11DSPIN
                TC      L
-ON             LXCH    Q
                TC      GETINREL
                INDEX   INREL
                CAF     SGNTAB -2
                TS      SGNON
                AD      ONE
                TS      SGNOFF
                TC      SGNCOM

SGNTAB          OCT     5               # -R1
                OCT     3               # -R2
                OCT     0               # -R3

SIGNTEST        LXCH    Q               # ALLOWS +,- ONLY WHEN DSPCOUNT=R1D1,
                CAF     THREE           # R2D1, OR R3D1.  ALLOWS ONLY FIRST OF
                MASK    DECBRNCH        # CONSECUTIVE +/- CHARACTERS.
                CCS     A               # IF LOW2 BITS OF DECBRNCH NOT= 0,  SIGN
                TC      ENDOFJOB        # FOR THIS WORD ALREADY IN. REJECT.
                CS      R1D1
                TC      SGNTST1
                CS      R2D1
                TC      SGNTST1
                CS      R3D1
                TC      SGNTST1
                TC      ENDOFJOB        # NO MATCH FOUND. SIGN ILLEGAL
SGNTST1         AD      DSPCOUNT
                EXTEND
                BZF     +2              # MATCH FOUND
                TC      Q
                TC      L               # SIGN LEGAL

# CLEAR BLANKS WHICH R1, R2, R3 IS CURRENT OR LAST TO BE DISPLAYED (PERTINENT
# XREG, YREG, ZREG IS CLEARED).  SUCCESSIVE CLEARS TAKE CARE OF EACH RX
# L/ RC UNTIL R1 IS DONE. THEN NO FURTHER ACTION.
#
# THE SINGLE COMPONENT LOAD VERBS ALLOW ONLY THE SINGLE RC THAT IS
# APPROPRIATE TO BE CLEARED.
#
# CLPASS        +0 PASS0, CAN BE BACKED UP
#               +NZ HIPASS, CAN BE BACKED UP
#               -NZ PASS0, CANNOT BE BACKED UP
```

```
CLEAR           CCS     DSPCOUNT
                AD      ONE
                TC      +2
                AD      ONE
                INDEX   A               # DO NOT CHANGE DSPCOUNT BECAUSE MAY LATER
                CAF     INRELTAB        # FAIL LEGALTST.
                TS      INREL           # MUST SET INREL, EVEN FOR HIPASS.
                CCS     CLPASS
                TC      CLPASHI         # +
                TC      +2              # +0    IF CLPASS IS +0 OR -, IT IS PASS0
                TC      +1              # -
                CA      INREL
                TC      LEGALTST
                TC      CLEAR1
CLPASHI         CCS     INREL
                TS      INREL
                TC      LEGALTST
                CAF     DOUBLK +2       # +3 TO - NUMBER, BACKS DATA REQUESTS.
                ADS     REQRET
                CA      INREL
                TS      MIXTEMP         # TEMP STORAGE FOR INREL
                EXTEND
                DIM     VERBREG         # DECREMENT VERB AND RE-DISPLAY
                TC      BANKCALL
                CADR    UPDATVB
                CA      MIXTEMP
                TS      INREL           # RESTORE INREL
CLEAR1          TC      CLR5
                INCR    CLPASS          # ONLY IF CLPASS IS + OR +0.
                TC      ENDOFJOB        # SET FOR HIGHER PASS.
CLR5            LXCH    Q               # USES 5BLANK BUT AVOIDS ITS TC GETINREL
                TC      5BLANK +2
LEGALTST        AD      NEG2
                CCS     A
                TC      Q               # LEGAL         INREL G/ 2
                TC      CCSHOLE
                TC      ENDOFJOB        # ILLEGAL       INREL=0,1
                TC      Q               # LEGAL         INREL=2

# 5BLANK BLANKS 5 CHAR DISPLAY WORD IN R1, R2, OR R3. IT ALSO ZEROES XREG,
# YREG, OR ZREG. PLACE ANY + DSPCOUNT NUMBER FOR PERTINENT RC INTO DSPCOUNT.
# DSPCOUNT IS LEFT SET TO LEFT MOST DSP NUMB FOR RC JUST BLANKED.

                TS      DSPCOUNT        # NEEDED FOR BLANKSUB
5BLANK          LXCH    Q
                TC      GETINREL
                CAF     ZERO
                INDEX   INREL
                TS      VERBREG         # ZERO X, Y, Z, REG.
```

```
                INDEX   INREL
                TS      XREGLP  -2
                TS      CODE
                INDEX   INREL           # ZERO PERTINENT DEC COMP BIT.
                CS      BIT7            # PROTECT OTHERS
                MASK    DECBRNCH
                MASK    BRNCHCON        # ZERO LOW 2 BITS.
                TS      DECBRNCH
                INDEX   INREL
                CAF     SINBLANK -2     # BLANK ISOLATED CHAR SEPARATELY
                TS      COUNT
                TC      DSPIN
5BLANK1         INDEX   INREL
                CAF     DOUBLK -2
                TS      DSPCOUNT
                TC      2BLANK
                CS      TWO
                ADS     DSPCOUNT
                TC      2BLANK
                INDEX   INREL
                CAF     R1D1 -2
                TS      DSPCOUNT        # SET DSPCOUNT TO LEFT MOST DSP NUMBER
                TC      L               # OF REG. JUST BLANKED

SINBLANK        OCT     16              # DEC 14
                OCT     5
                OCT     4
DOUBLK          OCT     15              # DEC 13
                OCT     11              # DEC 9
                OCT     3

BRNCHCON        OCT     77774

# 2BLANK BLANKS TWO CHAR. PLACE DSP NUMBER OF LEFT CHAR OF THE PAIR INTO
# DSPCOUNT. THIS NUMBER IS LEFT IN DSPCOUNT

2BLANK          CA      DSPCOUNT
                TS      SR
                CS      BLANKCON
                INHINT
                INDEX   SR
                XCH     DSPTAB
                EXTEND
                BZMF    +2              # IF OLD CONTENTS -, NOUT OK
                INCR    NOUT            # IF OLD CONTENTS +, +1 TO NOUT
                RELINT                  # IF -, NOUT OK
                TC      Q
BLANKCON        OCT     4000
```

```
# ENTER PASS 0 IS THE EXECUTE FUNCTION. HIGHER ORDER ENTERS ARE TO LOAD
# DATA. THE SIGN OF REQRET DETERMINES THE PASS, + FOR PASS 0, - FOR HIGHER
# PASSES.
#
# MACHINE CADR TO BE SPECIFIED (MCTBS) NOUNS DESIRE AN ECADR TO BE LOADED
# WHEN USED WITH LOAD VERBS, MONITOR VERBS, OR DISPLAY VERBS (EXCEPT
# VERB = FIXED MEMORY DISPLAY, WHICH REQUIRES AN FCADR).

                BANK    41
                SETLOC  PINBALL2
                BANK

                COUNT*  $$/PIN
NVSUBB          TC      NVSUB1          # STANDARD LEAD INS. DONT MOVE.
LOADLV1         TC      LOADLV

                                        # END OF STANDARD LEAD INS.

ENTER           CAF     ZERO
                TS      CLPASS
                CAF     ENDINST
                TS      ENTRET
                CCS     REQRET
                TC      ENTPAS0         # IF +, PASS 0
                TC      ENTPAS0         # IF +, PASS 0
                TC      +1              # IF -, NOT PASS 0
ENTPASHI        CAF     MMADREF
                AD      REQRET          # IF L/ 2 CHAR IN FOR MM CODE, ALARM
                EXTEND                  # AND RECYCLE (DECIDE AT MMCHANG+1).
                BZF     ACCEPTWD
                CAF     THREE           # IF DEC, ALARM IF L/ 5 CHAR IN FOR DATA,
                MASK    DECBRNCH        # BUT LEAVE REQRET - AND FLASH ON, SO
                CCS     A               # OPERATOR CAN SUPPLY MISSING NUMERICAL
                TC      +2              # CHARACTERS AND CONTINUE.
                TC      ACCEPTWD        # OCTAL. ANY NUMBER OF CHAR OK.
                CCS     DSPCOUNT
                TC      GODSPALM        # LESS THAN 5 CHAR DEC(DSPCOUNT IS +)
                TC      GODSPALM        # LESS THAN 5 CHAR DEC(DSPCOUNT IS +)
                TC      +1              # 5 CHAR IN (DSPCOUNT IS -)
ACCEPTWD        CS      REQRET          # 5 CHAR IN (DSPCOUNT IS -)
                TS      REQRET          # SET REQRET +.
                TC      FLASHOFF
                TC      REQRET

ENTEXIT         =       ENTRET

MMADREF         ADRES   MMCHANG +1      # ASSUMES TC REQMM AT MMCHANG.
```

```
LOWVERB         DEC     28              # LOWER VERB THAT AVOIDS NOUN TEST.

ENTPASO         CAF     ZERO            # NOUN VERB SUB ENTERS HERE
                TS      DECBRNCH
                CS      VD1             # BLOCK FURTHER NUM CHAR, SO THAT STRAY
                TS      DSPCOUNT        # CHAR DO NOT GET INTO VERB OR NOUN LTS.
TESTVB          CS      VERBREG         # IF VERB IS G/E LOWVB, SKIP NOUN TEST.
                TS      VERBSAVE        # SAVE VERB FOR POSSIBLE RECYCLE.
                AD      LOWVERB         # LOWVERB - VB
                EXTEND
                BZMF    VERBFAN         # VERB G/ E LOWVERB
TESTNN          EXTEND                  # VERB L/ LOWVERB
                DCA     LODNNLOC        # SWITCH BANKS TO NOUN TABLE READING
                DXCH    Z               # ROUTINE.
                INDEX   MIXBR
                TC      +0
                TC      +2              # NORMAL
                TC      MIXNOUN         # MIXED
                CCS     NNADTEM         # NORMAL
                TC      VERBFAN -2      #       NORMAL  IF +
                TC      GODSPALM        # NOT IN USE    IF +0
                TC      REQADD          # SPECIFY MACHINE CADR IF -
                INCR    NOUNCADR        # AUGMENT MACHINE CADR IF -0
                TC      SETNADD         # ECADR FROM NOUNCADR, SETS EB, NOUNADD.
                TC      INTMCTBS +2
REQADD          CAF     BIT15           # SET CLPASS FOR PASS 0 ONLY
                TS      CLPASS
                CS      ENDINST         # TEST IF REACHED HERE FROM INTERNAL OR
                AD      ENTEXIT         #       FROM EXTERNAL
                EXTEND
                BZF     +2              # EXTERNAL MACH CADR TO BE SPECIFIED
                TC      INTMCTBS
                TC      REQDATZ         # EXTERNAL MACH CADR TO BE SPECIFIED
                CCS     DECBRNCH        # ALARM AND RECYCLE IF DECIMAL USED
                TC      ALMCYCLE        # FOR MCTBS.
                CS      VD1             # OCTAL USED OK
                TS      DSPCOUNT        # BLOCK NUM CHAR IN
                CCS     CADRSTOR
                TC      +3              # EXTERNAL MCTBS DISPLAY WILL LEAVE FLASH
                TC      USEADD          # ON IF ENDIDLE NOT = +0.
                TC      +1
                TC      FLASHON
USEADD          XCH     ZREG
                TC      SETNCADR        # ECADR INTO NOUNCADR. SET EB, NOUNADD.
                EXTEND
                DCA     LODNNLOC        # SWITCH BANKS TO NOUN TABLE READING
                DXCH    Z               # ROUTINE.
                TC      VERBFAN

                EBANK=  DSPCOUNT
```

```
LODNNLOC        2CADR   LODNNTAB

NEG5            OCT     77772

INTMCTBS        CA      MPAC    +2      # INTERNAL MACH CADR TO BE SPECIFIED.
                TC      SETNCADR        # ECADR INTO NOUNCADR. SET EB. NOUNADD.
                CS      FIVE            # NVSUB CALL LEFT CADR IN MPAC+2 FOR MACH
                AD      VERBREG         # CADR TO BE SPECIFIED.
                EXTEND
                BZF     VERBFAN         # DONT DISPLAY CADR IF VB = 05.
                CAF     R3D1            # VB NOT = 05. DISPLAY CADR.
                TS      DSPCOUNT
                CA      NOUNCADR
                TC      DSPOCTWO
                TC      VERBFAN

                AD      ONE
                TC      SETNCADR        # ECADR INTO NOUNCADR. SETS EB, NOUNADD.
VERBFAN         CS      LST2CON
                AD      VERBREG         # VERB = LST2CON
                CCS     A
                AD      ONE             # VERB G/ LST2CON
                TC      +2
                TC      VBFANDIR        # VERB L/ LST2CON
                TS      MPAC
                TC      RELDSP          # RELEASE DISPLAY SYST
                TC      POSTJUMP        # GO TO GOEXTVB WITH VB=40 IN MPAC.
                CADR    GOEXTVB
LST2CON         DEC     40              # FIRST LIST2 VERB (EXTENDED VERB)

VBFANDIR        INDEX   VERBREG
                CAF     VERBTAB
                TC      BANKJUMP

VERBTAB         CADR    GODSPALM        # VB00 ILLEGAL
                CADR    DSPA            # VB01 DISPLAY OCT COMP 1 (R1)
                CADR    DSPB            # VB02 DISPLAY OCT COMP 2 (R1)
                CADR    DSPC            # VB03 DISPLAY OCT COMP 3 (R1)
                CADR    DSPAB           # VB04 DISPLAY OCT COMP 1,2 (R1,R2)
                CADR    DSPABC          # VB05 DISPLAY OCT COMP 1,2,3 (R1,R2,R3)
                CADR    DECDSP          # VB06 DECIMAL DISPLAY
                CADR    DSPDPDEC        # VB07 DP DECIMAL DISPLAY (R1,R2)
                CADR    GODSPALM        # VB08 SPARE
                CADR    GODSPALM        # VB09 SPARE
                CADR    DSPALARM        # VB10 SPARE
                CADR    MONITOR         # VB11 MONITOR OCT COMP 1 (R1)
                CADR    MONITOR         # VB12 MONITOR OCT COMP 2 (R1)
                CADR    MONITOR         # VB13 MONITOR OCT COMP 3 (R1)
                CADR    MONITOR         # VB14 MONITOR OCT COMP 1,2 (R1,R2)
```

```
                CADR    MONITOR         # VB15 MONITOR OCT COMP 1,2,3 (R1,R2,R3)
                CADR    MONITOR         # VB16 MONITOR DECIMAL
                CADR    MONITOR         # VB17 MONITOR DP DEC (R1,R2)
                CADR    GODSPALM        # VB18 SPARE
                CADR    GODSPALM        # VB19 SPARE
                CADR    GODSPALM        # VB20 SPARE
                CADR    ALOAD           # VB21 LOAD COMP 1 (R1)
                CADR    BLOAD           # VB22 LOAD COMP 2 (R2)
                CADR    CLOAD           # VB23 LOAD COMP 3 (R3)
                CADR    ABLOAD          # VB24 LOAD COMP 1,2 (R1,R2)
                CADR    ABCLOAD         # VB25 LOAD COMP 1,2,3 (R1,R2,R3)
                CADR    GODSPALM        # VB26 SPARE
                CADR    DSPFMEM         # VB27 FIXED MEMORY DISPLAY
                                        # THE FOLLOWING VERBS MAKE NO NOUN TEST
                CADR    GODSPALM        # VB28 SPARE
                CADR    GODSPALM        # VB29 SPARE
REQEXLOC        CADR    VBRQEXEC        # VB30 REQUEST EXECUTIVE
                CADR    VBRQWAIT        # VB31 REQUEST WAITLIST
                CADR    VBRESEQ         # VB32 RESEQUENCE
                CADR    VBPROC          # VB33 PROCEED WITHOUT DATA
                CADR    VBTERM          # VB34 TERMINATE CURRENT TEST OR LOAD REQ
                CADR    VBTSTLTS        # VB35 TEST LIGHTS
                CADR    SLAP1           # VB36 FRESH START
                CADR    MMCHANG         # VB37 CHANGE MAJOR MODE
                CADR    GODSPALM        # VB38 SPARE
                CADR    GODSPALM        # VB39 SPARE

# THE LIST2 VERBFAN IS LOCATED IN THE EXTENDED VERB BANK.
```

```
# NNADTAB CONTAINS A RELATIVE ADDRESS, IDADDREL (IN LOW 10 BITS), REFERRING
# TO WHERE 3 CONSECUTIVE ADDRESSES ARE STORED (IN IDADDTAB).
# MIXNOUN GETS DATA AND STORES IN MIXTEMP,+1,+2.  IT SETS NOUNADD FOR
# MIXTEMP.

MIXNOUN         CCS     NNADTEM
                TC      +4              # + IN USE
                TC      GODSPALM        # +0 NOT IN USE
                TC      +2              # - IN USE
                TC      +1              # -0 IN USE
                CS      SIX
                AD      VERBREG
                EXTEND
                BZMF    +2              # VERB L/E 6
                TC      VERBFAN         # AVOID MIXNOUN SWAP IF VB NOT = DISPLAY
                CAF     TWO
MIXNN1          TS      DECOUNT
                AD      MIXAD
                TS      NOUNADD         # SET NOUNADD TO MIXTEMP + K
                INDEX   DECOUNT         # GET IDADDTAB ENTRY FOR COMPONENT K
                CA      IDAD1TEM        # OF NOUN.
                TS      NOUNTEM
                                        # TEST FOR DP (FOR OCT DISPLAY).  IF SO, GET
                                        #       MINOR PART ONLY.
                TC      SFRUTMIX        # GET SF ROUT NUMBER IN A
                TC      DPTEST
                TC      MIXNN2          # NO DP
                INCR    NOUNTEM         # DP GET MINOR PART
MIXNN2          CA      NOUNTEM
                MASK    LOW11           # ESUBK (NO DP) OR (ESUBK)+1 (GARBLED) FOR DP
                TC      SETEBANK        # SET EBANK, LEAVE EADRES IN A.
                INDEX   A               # PICK UP C(ESUBK) NOT DP
                CA      0               # OR C((ESUBK)+1) FOR DP MINOR PART
                INDEX   NOUNADD
                XCH     0               # STORE IN MIXTEM + K
                CCS     DECOUNT
                TC      MIXNN1
                TC      VERBFAN

MIXAD           TC      MIXTEMP

# DPTEST        ENTER WITH SF ROUT NUMBER IN A.
#               RETURNS TO L+1 IF NO DP.
#               RETURNS TO L+2 IF DP.

DPTEST          INDEX   A
                TCF     +1
                TC      Q               # OCTAL ONLY NO DP
                TC      Q               # FRACT NO DP
```

```
                TC      Q               # DEG   NO DP
                TC      Q               # ARITH  NO DP
                TCF     DPTEST1         # DP1OUT
                TCF     DPTEST1         # DP2OUT
                TC      Q               # LRPOSOUT NO DP (DATA IN CHANNEL 33)
                TCF     DPTEST1         # DP3OUT
                TC      Q               # HMS    NO DP
                TC      Q               # M/S    NO DP
                TCF     DPTEST1         # DP4OUT
                TC      Q               # ARITH1   NO DP
                TC      Q               # 2INTOUT  NO DP TO GET HI PART IN MPAC
                TC      Q               # 360-CDU   NO DP
DPTEST1         INDEX   Q
                TC      1               # RETURN TO L+2

REQDATX         CAF     R1D1
                TCF     REQCOM
REQDATY         CAF     R2D1
                TCF     REQCOM
REQDATZ         CAF     R3D1
REQCOM          TS      DSPCOUNT
                CS      Q
                TS      REQRET
                TC      BANKCALL
                CADR    5BLANK
                TC      FLASHON
ENDRQDAT        TC      ENTEXIT

                TS      NOUNREG
UPDATNN         XCH     Q
                TS      UPDATRET
                EXTEND
                DCA     LODNNLOC        # SWITCH BANKS TO NOUN TABLE READING
                DXCH    Z               # ROUTINE.
                CCS     NNADTEM
                AD      ONE             # NORMAL
                TCF     PUTADD
                TCF     PUTADD +1       # MCTBS  DONT CHANGE NOUNADD
                TCF     PUTADD +1       # MCTBI  DONT CHANGE NOUNADD
PUTADD          TC      SETNCADR        # ECADR INTO NOUNCADR. SETS EB. NOUNADD.
                CAF     ND1
                TS      DSPCOUNT
                CA      NOUNREG
                TCF     UPDAT1

                TS      VERBREG
UPDATVB         XCH     Q
                TS      UPDATRET
                CAF     VD1
```

```
                TS      DSPCOUNT
                CA      VERBREG
UPDAT1          TC      POSTJUMP        # CANT USE SWCALL TO GO TO DSPDECVN, SINCE
                CADR    GOVNUPDT        # UPDATVB CAN ITSELF BE CALLED BY SWCALL.
                TC      UPDATRET

GOALMCYC        TC      ALMCYCLE        # NEEDED BECAUSE BANKJUMP CANT HANDLE F/F.

GODSPALM        TC      POSTJUMP
                CADR    DSPALARM
```

```
# NOUN TABLES
#
# NOUN CODE L/40, NORMAL NOUN CASE.  NOUN CODE G/E 40, MIXED NOUN CASE.
# FOR NORMAL CASE, NNADTAB CONTAINS ONE       ECADR      FOR EACH NOUN.
# +0 INDICATES NOUN NOT USED.   - ENTRY INDICATES MACHINE CADR (E OR F) TO
# BE SPECIFIED. -1 INDICATES CHANNEL TO BE SPECIFIED.  -0 INDICATES AUGMENT
# OF LAST MACHINE CADR SUPPLIED.
#
# FOR MIXED CASE, NNADTAB CONTAINS ONE INDIRECT ADDRESS (IDADDREL) IN LOW
# 10 BITS, AND THE COMPONENT CODE NUMBER IN THE HIGH 5 BITS.
#
# NNTYPTAB IS A PACKED TABLE OF THE FORM MMMMMNNNNNPPPPP.
#
# FOR THE NORMAL CASE,  M'S ARE THE COMPONENT CODE NUMBER.
#                       N'S ARE THE SF ROUTINE CODE NUMBER.
#                       P'S ARE THE SF CONSTANT CODE NUMBER.
#
# MIXED-CASE,           M'S ARE THE SF CONSTANT3 CODE NUMBER    3 COMPONENT CASE
#                       N'S ARE THE SF CONSTANT2 CODE NUMBER
#                       P'S ARE THE SF CONSTANT1 CODE NUMBER
#                       N'S ARE THE SF CONSTANT2 CODE NUMBER    2 COMPONENT CASE
#                       P'S ARE THE SF CONSTANT1 CODE NUMBER
#                       P'S ARE THE SF CONSTANT1 CODE NUMBER    1 COMPONENT CASE
#
# THERE IS ALSO AN INDIRECT ADDRESS TABLE (IDADDTAB) FOR MIXED CASE ONLY.
# EACH ENTRY CONTAINS ONE ECADR.    IDADDREL IS THE RELATIVE ADDRESS OF
# THE FIRST OF THESE ENTRIES.
#
# THERE IS ALSO A SCALE FACTOR ROUTINE NUMBER TABLE (RUTMXTAB) FOR MIXED
# CASE ONLY.  THERE IS ONE ENTRY PER MIXED NOUN.  THE FORM IS,
#
#       QQQQQRRRRRSSSSS
#
# Q'S ARE THE SF ROUTINE 3 CODE NUMBER          3 COMPONENT CASE
# R'S ARE THE SF ROUTINE 2 CODE NUMBER
# S'S ARE THE SF ROUTINE 1 CODE NUMBER
# R'S ARE THE SF ROUTINE 2 CODE NUMBER          2 COMPONENT CASE
# S'S ARE THE SF ROUTINE 1 CODE NUMBER
#
# IN OCTAL DISPLAY AND LOAD (OCT OR DEC) VERBS, EXCLUDE USE OF VERBS WHOSE
# COMPONENT NUMBER IS GREATER THAN THE NUMBER OF COMPONENTS IN NOUN.
# (ALL MACHINE ADDRESS TO BE SPECIFIED NOUNS ARE 3 COMPONENT.)
#
# IN MULTI-COMPONENT LOAD VERBS, NO MIXING OF OCTAL AND DECIMAL DATA
# COMPONENT WORDS IS ALLOWED. ALARM IF VIOLATION.
#
# IN DECIMAL LOADS OF DATA, 5 NUMERICAL CHARACTERS MUST BE KEYED IN
# BEFORE EACH ENTER. IF NOT, ALARM.
```

```
# DISPLAY VERBS

DSPABC          CS      TWO
                TC      COMPTEST
                INDEX   NOUNADD
                CS      2
                XCH     BUF     +2
DSPAB           CS      ONE
                TC      COMPTEST
                INDEX   NOUNADD
                CS      1
                XCH     BUF     +1
DSPA            TC      DECTEST
                TC      TSTFORDP
                INDEX   NOUNADD
                CS      0
DSPCOM1         XCH     BUF
                TC      DSPCOM2
DSPB            CS      ONE
                TC      DCOMPTST
                INDEX   NOUNADD
                CS      1
                TC      DSPCOM1
DSPC            CS      TWO
                TC      DCOMPTST
                INDEX   NOUNADD
                CS      2
                TC      DSPCOM1
DSPCOM2         CS      TWO             # A  B  C  AB ABC
                AD      VERBREG         # -1 -0 +1 +2 +3   IN A
                CCS     A               # +0 +0 +0 +1 +2   IN A AFTER CCS
                TC      DSPCOM3
                TC      ENTEXIT
                TC      +1
DSPCOM3         TS      DISTEM          # +0 +1 +2 INTO DISTEM
                INDEX   A
                CAF     R1D1
                TS      DSPCOUNT
                INDEX   DISTEM
                CS      BUF
                TC      DSPOCTWO
                XCH     DISTEM
                TC      DSPCOM2 +2

# COMPTEST ALARMS IF COMPONENT NUMBER OF VERB (LOAD OR OCT DISPLAY) IS
# GREATER THAN THE HIGHEST COMPONENT NUMBER OF NOUN.

COMPTEST        TS      SFTEMP1         # VERB COMP
                LXCH    Q
COMPTST1        TC      GETCOMP
                TC      LEFT5
                MASK    THREE           # NOUN COMP
```

```
                AD      SFTEMP1         # NOUN COMP = VERB COMP
                CCS     A
                TC      L               # NOUN COMP G/ VERB COMP
                TC      CCSHOLE
                TC      GODSPALM        # NOUN COMP L/ VERB COMP
NDCMPTST        TC      L               # NOUN COMP = VERB COMP

# DCOMPTST ALARMS IF DECIMAL ONLY BIT (BIT4 OF COMP CODE NUMBER) = 1.
# IF NOT, IT PERFORMS REGULAR COMPTEST.

DCOMPTST        TS      SFTEMP1         # - VERB COMP
                LXCH    Q
                TC      DECTEST
                TC      COMPTST1

DECTEST         EXTEND                  # ALARMS IF DEC ONLY BIT = 1 (BIT4 OF COMP
                QXCH    MPAC +2         # CODE NUMBER).  RETURNS IF NOT.
                TC      GETCOMP
                MASK    BIT14
                CCS     A
                TC      GODSPALM
                TC      MPAC +2

DCTSTCYC        LXCH    Q               # ALARMS AND RECYCLES IF DEC ONLY BIT = 1
                TC      GETCOMP         # (BIT4 OF COMP CODE NUMBER). RETURNS
                MASK    BIT14           # IF NOT.  USED BY LOAD VERBS.
                CCS     A
                TC      ALMCYCLE
                TC      L

# NOUNTEST ALARMS IF NO-LOAD BIT (BIT5 OF COMP CODE NUMBER) = 1.
# IF NOT, IT RETURNS.

NOUNTEST        LXCH    Q
                TC      GETCOMP
                CCS     A
                TC      L
                TC      L
                TC      GODSPALM

TSTFORDP        LXCH    Q               # TEST FOR DP. IF SO, GET MINOR PART ONLY.
                CA      NNADTEM
                AD      ONE             # IF NNADTEM = -1, CHANNEL TO BE SPECIFIED
                EXTEND
                BZF     CHANDSP
                INDEX   MIXBR
                TC      +0
                TC      +2              # NORMAL
```

```
                TC      L               # MIXED CASE ALREADY HANDLED IN MIXNOUN
                TC      SFRUTNOR
                TC      DPTEST
                TC      L               # NO DP
                INCR    NOUNADD         # DP    E+1 INTO NOUNADD FOR MINOR PART.
                TC      L

CHANDSP         CA      NOUNCADR
                MASK    LOW9
                EXTEND
                INDEX   A
                READ    0
                CS      A
                TCF     DSPCOM1

COMPICK         ADRES   NNTYPTEM
                ADRES   NNADTEM

GETCOMP         INDEX   MIXBR           # NORMAL                        MIXED
                CAF     COMPICK -1      # ADRES NNTYPTEM                ADRES NNADTEM
                INDEX   A
                CA      0               # C(NNTYPTEM)                   C(NNADTEM)
                MASK    HI5             # GET HI5 OF NNTYPTAB (NORM)    OF NNADTAB (MIX)
                TC      Q

DECDSP          TC      GETCOMP
                TC      LEFT5
                MASK    THREE
                TS      DECOUNT         # COMP NUMBER INTO DECOUNT
DSPDCGET        TS      DECTEM          # PICKS UP DATA
                AD      NOUNADD         # DECTEM 1COMP +0, 2COMP +1, 3COMP +2
                INDEX   A
                CS      0
                INDEX   DECTEM
                XCH     XREG            # CANT USE BUF SINCE DMP USES IT.
                CCS     DECTEM
                TC      DSPDCGET        # MORE TO GET
DSPDCPUT        CAF     ZERO            # DISPLAYS DATA
                TS      MPAC +1         # DECOUNT 1COMP +0, 2COMP +1, 3COMP +2
                TS      MPAC +2
                INDEX   DECOUNT
                CAF     R1D1
                TS      DSPCOUNT
                INDEX   DECOUNT
                CS      XREG
                TS      MPAC
                TC      SFCONUM         # 2X (SF CON NUMB) IN A
```

```
                TS      SFTEMP1
                EXTEND                  # SWITCH BANKS TO SF CONSTANT TABLE
                DCA     GTSFOUTL        #       READING ROUTINE.
                DXCH    Z               # LOADS SFTEMP1, SFTEMP2
                INDEX   MIXBR
                TC      +0
                TC      DSPSFNOR
                TC      SFRUTMIX
                TC      DECDSP3

DSPSFNOR        TC      SFRUTNOR
                TC      DECDSP3

                EBANK=  DSPCOUNT
GTSFOUTL        2CADR   GTSFOUT

DSPDCEND        TC      BANKCALL        # ALL SFOUT ROUTINES END HERE
                CADR    DSPDECWD
                CCS     DECOUNT
                TC      +2
                TC      ENTEXIT
                TS      DECOUNT
                TC      DSPDCPUT        # MORE TO DISPLAY

DECDSP3         INDEX   A
                CAF     SFOUTABR
                TC      BANKJUMP

SFOUTABR        CADR    PREDSPAL        # ALARM IF DEC DISP WITH OCTAL ONLY NOUN
                CADR    DSPDCEND
                CADR    DEGOUTSF
                CADR    ARTOUTSF
                CADR    DP1OUTSF
                CADR    DP2OUTSF
                CADR    LRPOSOUT
                CADR    DP3OUTSF
                CADR    HMSOUT
                CADR    M/SOUT
                CADR    DP2OUTSF
                CADR    AROUT1SF
                CADR    2INTOUT
                CADR    360-CDUO
ENDRTOUT        EQUALS

# THE FOLLOWING IS A TYPICAL SF ROUTINE. IT USES MPAC. LEAVES RESULTS
# IN MPAC, MPAC+1.  ENDS WITH TC DSPDCEND
```

```
                SETLOC  BLANKCON +1

                COUNT*  $$/PIN
# DEGOUTSF SCALES BY .18 THE LOW 14 BITS OF ANGLE, ADDING .18 FOR
# NUMBERS IN THE NEGATIVE (AGC) RANGE.

DEGOUTSF        CAF     ZERO
                TS      MPAC +2         # SET INDEX FOR FULL SCALE.
                TC      FIXRANGE
                TC      +2              # NO AUGMENT NEEDED (SFTEMP1 AND 2 ARE 0)
                TC      SETAUG          # SET AUGMENTER ACCORDING TO C(MPAC +2)
                TC      DEGCOM

# 360-CDUD COMPUTES 360 - CDU ANGLE IN MPAC, STORES RESULT IN MPAC AND
# GOES TO DEGOUTSF.

360-CDUO        TC      360-CDU
                TC      DEGOUTSF

360-CDU         CA      MPAC
                MASK    POSMAX          # IF ANGLE IS 0 OR 180 DEGREES, DO NOTHING
                EXTEND
                BZF     360-CDUE
                CS      MPAC            # COMPUTE 360 DEGREES MINUS ANGLE
                AD      ONE
                TS      MPAC
360-CDUE        TC      Q

# LRPOSOUT DISPLAYS +0,1,2, OR 3 (WHOLE) FOR CHANNEL 33, BITS 7-6 = 11,10,
# 01,00 RESPECTIVELY

LRPOSOUT        EXTEND
                READ    CHAN33
                EXTEND
                MP      BIT10           # BITS 7-6 TO BITS 2-1
                COM
                MASK    THREE
                TS      MPAC
                TC      ARTOUTSF        # DISPLAY AS WHOLE

SETAUG          EXTEND                  # LOADS SFTEMP1 AND SFTEMP2 WITH THE
                INDEX   MPAC +2         # DP AUGMENTER CONSTANT
                DCA     DEGTAB
                DXCH    SFTEMP1
                TC      Q

FIXRANGE        CCS     MPAC            # IF MPAC IS + RETURN TO L+1
                TC      Q               # IF MPAC IS - RETURN TO L+2 AFTER
                TC      Q               # MASKING OUT THE SIGN BIT
                TCF     +1
```

```
                CS      BIT15
                MASK    MPAC
                TS      MPAC
                INDEX   Q
                TC      1

DEGCOM          EXTEND                  # LOADS MULTIPLIER, DOES SHORTMP, AND
                INDEX   MPAC +2         # ADDS AUGMENTER.
                DCA     DEGTAB
                DXCH    MPAC            # ADJUSTED ANGLE IN A
                TC      SHORTMP
                DXCH    SFTEMP1
                DAS     MPAC
                TC      SCOUTEND

DEGTAB          OCT     05605           # HI PART OF    .18
                OCT     03656           # LOW PART OF   .18
                OCT     16314           # HI PART OF    .45
                OCT     31463           # LO PART OF    .45

ARTOUTSF        DXCH    SFTEMP1         # ASSUMES POINT AT LEFT OF DP SFCON
                DXCH    MPAC
                TC      PRSHRTMP        # IF C(A) = -0, SHORTMP FAILS TO GIVE -0.
SCOUTEND        TC      POSTJUMP
                CADR    DSPDCEND

AROUT1SF        DXCH    SFTEMP1         # ASSUMES POINT BETWEEN HI AND LO PARTS OF
                DXCH    MPAC            # DP SFCON. SHIFTS RESULTS LEFT 14, BY
                TC      PRSHRTMP        # TAKING RESULTS FROM MPAC+1, MPAC+2.
                TC      L14/OUT

DP1OUTSF        TC      DPOUT           # SCALES MPAC, MPAC +1 BY DP SCALE FACTOR
L14/OUT         XCH     MPAC +2         # IN SFTEMP1, SFTEMP2.  THEN SCALE RESULT
                XCH     MPAC +1         # BY B14
                TS      MPAC
                TC      SCOUTEND

DP2OUTSF        TC      DPOUT           # SCALES MPAC, MPAC +1 BY DP SCALE FACTOR
                TC      SCOUTEND

DP3OUTSF        TC      DPOUT           # ASSUMES POINT BETWEEN BITS 7-8 OF HIGH
                CAF     SIX             # LEFT BY 7, ROUNDS MPAC+2 INTO MPAC+1.
                TC      TPLEFTN         # SHIFT LEFT 7.
                TC      SCOUTEND
```

```
MPAC+6          =       MPAC +6         # USE MPAC +6 INSTEAD OF OVFIND

DPOUT           XCH     Q
                TS      MPAC+6
                TC      READLO          # GET FRESH DATA FOR BOTH HI AND LO.
                TC      TPAGREE         # MAKE DP DATA AGREE
                TC      DMP
                ADRES   SFTEMP1
                TC      MPAC+6

# THE FOLLOWING ROUTINE DISPLAYS TWO CONTIGUOUS SP POSITIVE INTEGERS
# AS TWO POSITIVE DECIMAL INTEGERS IN RXD1-RXD2 AND RXD4-RXD5 (RXD3 IS
# BLANKED).  THE INTEGER IN THE LOWER NUMBERED ADDRESS IS DISPLAYED IN
# RXD1-RXD2.

2INTOUT         TC      5BLANK          # TO BLANK RXD3
                TC      +ON             # TURN ON + SIGN
                CA      MPAC
                TC      DSPDECVN        # DISPLAY 1ST INTEGER (LIKE VERB AND NOUN)
                CS      THREE
                INDEX   DECOUNT
                AD      R1D1            # RXD4
                TS      DSPCOUNT
                TC      READLO          # GET 2ND INTEGER
                CA      MPAC +1
                TC      DSPDECVN        # DISPLAY 2ND INTEGER (LIKE VERB AND NOUN)
                TC      POSTJUMP
                CADR    DSPDCEND +2

# READLO PICKS UP FRESHDATA FOR BOTH HI AND LO AND LEAVES IT IN
# MPAC, MPAC+1.  THIS IS NEEDED FOR TIME DISPLAY.  IT ZEROES MPAC+2, BUT
# DOES NOT FORCE TPAGREE.

READLO          XCH     Q
                TS      TEM4
                INDEX   MIXBR
                TC      +0
                TC      RDLONOR
                INDEX   DECOUNT
                CA      IDAD1TEM        # GET IDADDTAB ENTRY FOR COMP K OF NOUN.
                MASK    LOW11           # E SUBK
                TC      SETEBANK        # SET EB, LEAVE EADRES IN A.
READLO1         EXTEND                  # MIXED                 NORMAL
                INDEX   A               # C(ESUBK)              C(E)
                DCA     0               # C(E SUBK)+1)          C(E+1)
                DXCH    MPAC
                CAF     ZERO
                TS      MPAC    +2
                TC      TEM4
```

```
RDLONOR         CA      NOUNADD         # E
ENDRDLO         TC      READLO1

                BANK    42
                SETLOC  PINBALL3
                BANK

                COUNT*  $$/PIN
HMSOUT          TC      BANKCALL        # READ FRESH DATA FOR HI AND LO INTO MPAC,
                CADR    READLO          # MPAC+1.
                TC      TPAGREE         # MAKE DP DATA AGREE.
                TC      SEPSECNR        # LEAVE FRACT SEC/60 IN MPAC, MPAC+1. LEAVE
                                        # WHOLE MIN IN BIT13 OF LOTEMOUT AND ABOVE
                TC      DMP             # USE ONLY FRACT SEC/60 MOD 60
                ADRES   SECON2          # MULT BY .06
                CAF     R3D1            # GIVES CENTI-SEC/10EXP5 MOD 60
                TS      DSPCOUNT
                TC      BANKCALL        # DISPLAY SEC MOD 60
                CADR    DSPDECWD
                TC      SEPMIN          # REMOVE REST OF SECONDS
                CAF     MINCON2         # LEAVE FRACT MIN/60 IN MPAC+1.  LEAVE
                XCH     MPAC            # WHOLE HOURS IN MPAC.
                TS      HITEMOUT        # SAVE WHOLE HOURS.
                CAF     MINCON2 +1
                XCH     MPAC    +1      # USE ONLY FRACT MIN/60 MOD 60
                TC      PRSHRTMP        # IF C(A) = -0, SHORTMP FAILS TO GIVE -0.
                                        # MULT BY .0006
                CAF     R2D1            # GIVE MIN/10EXP5 MOD 60
                TS      DSPCOUNT
                TC      BANKCALL        # DISPLAY MIN MOD 60
                CADR    DSPDECWD
                EXTEND                  # MINUTES, SECONDS HAVE BEEN REMOVED
                DCA     HRCON1
                DXCH    MPAC
                CA      HITEMOUT        # USE WHOLE HOURS
                TC      PRSHRTMP        # IF C(A) = -0, SHORTMP FAILS TO GIVE -0.
                                        # MULT BY .16384
                CAF     R1D1            # GIVES HOURS/10EXP5
                TS      DSPCOUNT
                TC      BANKCALL        # USE REGULAR DSPDECWD, WITH ROUND OFF.
                CADR    DSPDECWD
                TC      ENTEXIT

SECON1          2DEC*   1.666666666 E-4 B12*    # 2EXP12/6000
SECON2          OCT     01727           # .06 FOR SECONDS DISPLAY
                OCT     01217
MINCON2         OCT     00011           # .0006 FOR MINUTES DISPLAY
                OCT     32445
```

```
MINCON1         OCT     02104           # .066..66 UPPED BY 2EXP-28
                OCT     10422
HRCON1          2DEC    .16384

                OCT     00000
RNDCON          OCT     00062           # .5 SEC

M/SOUT          TC      BANKCALL        # READ FRESH DATA FOR HI AND LO INTO MPAC.
                CADR    READLO          # MPAC+1.
                TC      TPAGREE         # MAKE DP DATA AGREE
                CCS     MPAC            # IF MAG OF (MPAC, MPAC+1) G/ 59 M 59 S.
                TC      +2              # DISPLAY 59B59, WITH PROPER SIGN.
                TC      M/SNORM         # MPAC = +0. L/ 59M58.5S
                AD      M/SCON1         # - HI PART OF (59M58.5S) +1 FOR CCS
                CCS     A               # MAG OF MPAC - HI PART OF (59M58.5S)
                TC      M/SLIMIT        # G/ 59M58.5S
                TC      M/SNORM         # ORIGINAL MPAC = -0. L/ 59M58.5S
                TC      M/SNORM         # L/ 59M58.5S
                CCS     MPAC +1         # MAG OF MPAC = HI PART OF 59M58.5S
                TC      +2
                TC      M/SNORM         # MPAC+1 = +0. L/ 59M58.5S
                AD      M/SCON2         # - LO PART OF (59M58.5S) +1 FOR CCS
                CCS     A               # MAG OF MPAC+1 - LO PART OF (59M58.5S)
                TC      M/SLIMIT        # G/ 59M58.5S
                TC      M/SNORM         # ORIGINAL MPAC+1 = -0. L/ 59M58.5S
                TC      M/SNORM         # L/ 59M58.5S
M/SLIMIT        CCS     MPAC            # = 59M58.5S    LIMIT
                CAF     M/SCON3         # MPAC CANNOT BE +/- 0 AT THIS POINT.
                TC      +LIMIT          # FORCE MPAC, MPAC+1 TO +/- 59M59.5S
                CS      M/SCON3
                TS      MPAC            # WILL DISPLAY 59M59S IN DSPDECNR
                CS      M/SCON3 +1
LIMITCOM        TS      MPAC +1
                CAF     NORMADR         # SET RETURN TO M/SNORM+1.
                TC      SEPSECNR +1
+LIMIT          TS      MPAC
                CAF     M/SCON3 +1
                TC      LIMITCOM
M/SNORM         TC      SEPSEC          # LEAVE FRACT SEC/60 IN MPAC,MPAC+1. LEAVE
                                        # WHOLE MIN IN BIT13 OF LOTEMOUT AND ABOVE
                CAF     HISECON         # USE ONLY FRACT SEC/60 MOD 60
                TC      SHORTMP         # MULT BY .6 + 2EXP-14
                CS      THREE           # GIVES SEC/100 MOD 60
                ADS     DSPCOUNT        # DSPCOUNT ALREADY SET TO RXD1
                TC      BANKCALL        # DISPLAY SEC MOD 60 IN D4D5.
                CADR    DSPDC2NR
                CAF     ZERO
                TS      CODE
                CS      TWO
```

```
                INDEX   DECOUNT
                AD      R1D1            # RXD3
                TS      COUNT
                TC      BANKCALL        # BLANK MIDDLE CHAR
                CADR    DSPIN
                TC      SEPMIN          # REMOVE REST OF SECONDS
                XCH     MPAC +1         # LEAVE FRACT MIN/60 IN MPAC+1
                EXTEND                  # USE ONLY FRACT MIN/60 MOD 60
                MP      HIMINCON        # MULT BY .6 + 2EXP-7
                DXCH    MPAC            # GIVES MIN/100 MOD 60
                INDEX   DECOUNT
                CAF     R1D1            # RXD1
                TS      DSPCOUNT
                TC      BANKCALL        # DISPLAY MIN MOD 60 IN D1D2.
                CADR    DSPDC2NR
                TC      POSTJUMP
                CADR    DSPDCEND +2

HISECON         OCT     23147           # .6 + 2EXP-14
HIMINCON        OCT     23346           # .6 + 2EXP-7

M/SCON1         OCT     77753           # - HI PART OF (59M58.5S) +1
M/SCON2         OCT     41126           # - LO PART OF (59M58.5S) +1
NORMADR         ADRES   M/SNORM +1
M/SCON3         OCT     00025           # 59M 59.5S
                OCT     37016

SEPSEC          CCS     MPAC    +1      # IF +, ROUND BY ADDING .5 SEC
                TCF     POSEC           # IF -, ROUND BY SUBTRACING .5 SEC
                TCF     POSEC           # FINDS TIME IN MPAC, MPAC+1
                TCF     +1              # ROUNDS OFF BY +/- .5 SEC
                EXTEND                  # LEAVES WHOLE MIN IN BIT13 OF
                DCS     RNDCON  -1      # LOTEMOUT AND ABOVE.
SEPSEC1         DAS     MPAC            # LEAVES FRACT SEC/60 IN MPAC, MPAC+1.
                TCF     SEPSECNR
POSEC           EXTEND
                DCA     RNDCON -1
                TCF     SEPSEC1
SEPSECNR        XCH     Q               # THIS ENTRY AVOIDS ROUNDING BY .5 SEC
                TS      SEPSCRET
                TC      DMP             # MULT BY 2EXP12/6000
                ADRES   SECON1          # GIVES FRACT SEC/60 IN BIT12 OF MPAC+1
                EXTEND                  # AND BELOW.
                DCA     MPAC            # SAVE MINUTES AND HOURS
                DXCH    HITEMOUT
                TC      TPSL1
                TC      TPSL1           # GIVES FRACT SEC/60 IN MPAC+1, MPAC+2.
                CAF     ZERO
                XCH     MPAC +2         # LEAVE FRACT SEC/60 IN MPAC, MPAC+1.
```

```
                                XCH     MPAC +1
                                XCH     MPAC
                                TC      SEPSCRET

SEPMIN                          XCH     Q                       # FIND WHOLE MINUTES IN BIT13
                                TS      SEPMNRET                # OF LOTEMOUT AND ABOVE.
                                CA      LOTEMOUT                # REMOVES REST OF SECONDS.
                                EXTEND                          # LEAVES FRACT MIN/60 IN MPAC+1.
                                MP      BIT3                    # LEAVES WHOLE HOURS IN MPAC.
                                EXTEND                          # SR 12, THROW AWAY LP.
                                MP      BIT13                   # SR 2, TAKE FROM LP. = SL 12.
                                LXCH    MPAC +1                 # THIS FORCES BITS 12-1 TO 0 IF +.
                                                                # FORCES BITS 12-1 TO 1 IF -.
                                CA      HITEMOUT
                                TS      MPAC
                                TC      DMP                     # MULT BY 1/15
                                ADRES   MINCON1                 # GIVES FRACT MIN/60 IN MPAC+1.
ENDSPMIN                        TC      SEPMNRET                # GIVES WHOLE HOURS IN MPAC.

# THIS IS A SPECIAL PURPOS VERB FOR DISPLAYING A DOUBLE PRECISION AGC
# WORD AS 10 DECIMAL DIGITS ON THE AGC DISPLAY PANEL.  IT CAN BE USED WITH
# ANY NOUN, EXCEPT MIXED NOUNS. IT DISPLAYS THE CONTENTS
# OF THE REGISTER NOUNADD IS POINTING TO.  IF USED WITH NOUNS WHICH ARE
# INHERENTLY NOT DP SUCH AS THE CDU COUNTERS THE DISPLAY WILL BE GARBAGE.
# DISPLAY IS IN R1 AND R2 ONLY WITH THE SIGN IN R1.

                                SETLOC  ENDRDLO +1

                                COUNT*  $$/PIN
DSPDPDEC                        INDEX   MIXBR
                                TC      +0
                                TC      +2                      # NORMAL NOUN
                                TC      DSPALARM
                                EXTEND
                                INDEX   NOUNADD
                                DCA     0
                                DXCH    MPAC
                                CAF     R1D1
                                TS      DSPCOUNT
                                CAF     ZERO
                                TS      MPAC +2
                                TC      TPAGREE
                                TC      DSP2DEC
ENDDPDEC                        TC      ENTEXIT
```

```
# LOAD VERBS            IF ALARM CONDITION IS DETECTED DURING EXECUTE,
# CHECK FAIL LIGHT IS TURNED ON AND ENDOFJOB.  IF ALARM CONDITION IS
# DETECTED DURING ENTER OF DATA, CHECK FAIL IS TURNED ON AND IT RECYCLES
# TO EXECUTE OF ORIGINAL LOAD VERB.  RECYCLE CAUSED BY  1) DECIMAL MACHINE
# CADR  2) MIXTURE OF OCTAL/DECIMAL DATA  3) OCTAL DATA INTO DECIMAL
# ONLY NOUN  4) DEC DATA INTO OCT ONLY NOUN  5) DATA TOO LARGE FOR SCALE
# 6) FEWER THAN 3 DATA WORDS LOADED FOR HRS, MIN, SEC NOUN.  (2)-(6) ALARM
# AND RECYCLE OCCUR AT FINAL ENTER OF SET. (1) ALARM AND RECYCLE OCCUR AT
# ENTER OF CADR.

                SETLOC  ENDRTOUT

                COUNT*  $$/PIN
ABCLOAD         CS      TWO
                TC      COMPTEST
                TC      NOUNTEST        # TEST IF NOUN CAN BE LOADED.
                CAF     VBSP1LD
                TC      UPDATVB -1
                TC      REQDATX
                CAF     VBSP2LD
                TC      UPDATVB -1
                TC      REQDATY
                CAF     VBSP3LD
                TC      UPDATVB -1
                TC      REQDATZ

PUTXYZ          CS      SIX             # TEST THAT THE 3 DATA WORDS LOADED ARE
                TC      ALLDC/OC        # ALL DEC OR ALL OCT.
                EXTEND
                DCA     LODNNLOC        # SWITCH BANKS TO NOUN TABLE READING
                DXCH    Z               # ROUTINE.
                CAF     ZERO            # X COMP
                TC      PUTCOM
                INDEX   NOUNADD
                TS      0
                CAF     ONE             # Y COMP
                TC      PUTCOM
                INDEX   NOUNADD
                TS      1
                CAF     TWO             # Z COMP
                TC      PUTCOM
                INDEX   NOUNADD
                TS      2
                CS      SEVEN           # IF NOUN 7 HAS JUST BEEN LOADED, SET
                AD      NOUNREG         # FLAG BITS AS SPECIFIED.
                EXTEND
                BZF     +2
                TC      LOADLV
```

```
                CA      XREG            # ECADR OF FLAG WORD.
                TC      SETNCADR +1     # SET EBANK, NOUNADD.
                CA      ZREG            # ZERO TO RESET BITS, NON-ZERO TO SET BITS.
                INHINT
                EXTEND
                BZF     BITSOFF
                INDEX   NOUNADD
                CS      0
                MASK    YREG            # BITS TO BE PROCESSED.
                INDEX   NOUNADD
                ADS     0               # SET BITS.
                TC      BITSOFF1
BITSOFF         CS      YREG            # BITS TO BE PROCESSED.
                INDEX   NOUNADD
                MASK    0
                INDEX   NOUNADD
                TS      0               # RESET BITS.
BITSOFF1        RELINT
                TC      LOADLV

ABLOAD          CS      ONE
                TC      COMPTEST
                TC      NOUNTEST        # TEST IF NOUN CAN BE LOADED.
                CAF     VBSP1LD
                TC      UPDATVB -1
                TC      REQDATX
                CAF     VBSP2LD
                TC      UPDATVB -1
                TC      REQDATY
PUTXY           CS      FIVE            # TEST THAT THE 2 DATA WORDS LOADED ARE
                TC      ALLDC/OC        # ALL DEC OR ALL OCT.
                EXTEND
                DCA     LODNNLOC        # SWITCH BANKS TO NOUN TABLE READING
                DXCH    Z               # ROUTINE.
                CAF     ZERO            # X COMP
                TC      PUTCOM
                INDEX   NOUNADD
                TS      0
                CAF     ONE             # Y COMP
                TC      PUTCOM
                INDEX   NOUNADD
                TS      1
                TC      LOADLV

ALOAD           TC      REQDATX
                EXTEND
                DCA     LODNNLOC        # SWITCH BANKS TO NOUN TABLE READING
                DXCH    Z               # ROUTINE.
                CAF     ZERO            # X COMP
                TC      PUTCOM
```

```
                INDEX   NOUNADD
                TS      0
                TC      LOADLV

BLOAD           CS      ONE
                TC      COMPTEST
                CAF     BIT15           # SET CLPASS FOR PASS0 ONLY
                TS      CLPASS
                TC      REQDATY
                EXTEND
                DCA     LODNNLOC        # SWITCH BANKS TO NOUN TABLE READING
                DXCH    Z               # ROUTINE.
                CAF     ONE
                TC      PUTCOM
                INDEX   NOUNADD
                TS      1
                TC      LOADLV

CLOAD           CS      TWO
                TC      COMPTEST
                CAF     BIT15           # SET CLPASS FOR PASS0 ONLY
                TS      CLPASS
                TC      REQDATZ
                EXTEND
                DCA     LODNNLOC        # SWITCH BANKS TO NOUN TABLE READING
                DXCH    Z               # ROUTINE.
                CAF     TWO
                TC      PUTCOM
                INDEX   NOUNADD
                TS      2
                TC      LOADLV

LOADLV          CAF     ZERO
                TS      DECBRNCH
                CS      ZERO
                TS      LOADSTAT
                TC      RELDSP          # RELEASE FOR PRIORITY DISPLAY PROBLEM.
                CS      VD1             # TO BLOCK NUMERICAL CHARACTERS AND
                TS      DSPCOUNT        # CLEARS AFTER A COMPLETED LOAD
                TC      POSTJUMP        # AFTER COMPLETED LOAD, GO TO RECALTST
                CADR    RECALTST        # TO SEE IF THERE IS RECALL FROM ENDIDLE.

VBSP1LD         DEC     21              # VB21 = ALOAD
VBSP2LD         DEC     22              # VB22 = BLOAD
VBSP3LD         DEC     23              # VB23 = CLOAD

ALLDC/OC        TS      DECOUNT         # TESTS THAT DATA WORDS LOADED ARE EITHER
                CS      DECBRNCH        # ALL DEC OR ALL OCT. ALARMS IF NOT.
                TS      SR
```

```
                CS      SR
                CS      SR              # SHIFTED RIGHT 2
                CCS     A               # DEC COMP BITS IN LOW 3
                TCF     +2              # SOME ONES IN LOW 3
                TC      Q               # ALL ZEROS. ALL OCTAL.  OK
                AD      DECOUNT         # DEC COMP = 7 FOR 3COMP, =6 FOR 2COMP
                EXTEND                  # (BUT IT HAS BEEN DECREMENTED BY CCS)
                BZF     +2              # MUST MATCH 6 FOR 3COMP, 5 FOR 2COMP.
                TC      ALMCYCLE        # ALARM AND RECYCLE.
GOQ             TC      Q               # ALL REQUIRED ARE DEC.  OK

SFRUTNOR        XCH     Q               # GETS SF ROUTINE NUMBER FOR NORMAL CASE
                TS      EXITEM          # CAN'T USE L FOR RETURN. TSTFORDP USES L.
                CAF     MID5
                MASK    NNTYPTEM
                TC      RIGHT5
                TC      EXITEM          # SF ROUTINE NUMBER IN A

SFRUTMIX        XCH     Q               # GETS SF ROUTINE NUMBER FOR MIXED CASE
                TS      EXITEM
                INDEX   DECOUNT
                CAF     DISPLACE        # PUT TC GOQ, TC RIGHT5, OR TC LEFT5 IN L
                TS      L
                INDEX   DECOUNT
                CAF     LOW5            # LOW5, MID5, OR HI5 IN A
                MASK    RUTMXTEM        # GET HI5, MID5, OR LOW5 OF RUTMXTAB ENTRY
                INDEX   L
                TC      0

# DO TC GOQ(DECOUNT=0), DO TC RIGHT5(DECOUNT=1), DO TC LEFT5(DECOUNT=2).
SFRET1          TC      EXITEM          # SF ROUTINE NUMBER IN A

SFCONUM         XCH     Q               # GETS 2X(SF CONSTANT NUMBER)
                TS      EXITEM
                INDEX   MIXBR
                TC      +0
                TC      CONUMNOR        # NORMAL NOUN
                INDEX   DECOUNT         # MIXED NOUN
                CAF     DISPLACE
                TS      L               # PUT TC GOQ, TC RIGHT5, OR TC LEFT5 IN L
                INDEX   DECOUNT
                CAF     LOW5
                MASK    NNTYPTEM
                INDEX   L
                TC      0

# DO TC GOQ(DECOUNT=0), DO TC RIGHT5(DECOUNT=1), DO TC LEFT5(DECOUNT=2).
SFRET           DOUBLE                  # 2X(SF CONSTANT NUMBER) IN A
                TC      EXITEM

DISPLACE        TC      GOQ
```

```
                TC      RIGHT5
                TC      LEFT5

CONUMNOR        CAF     LOW5            # NORMAL NOUN ALWAYS GETS LOW5 OF
                MASK    NNTYPTEM        # NNTYPTAB FOR SF CONUM.
                DOUBLE
                TC      EXITEM          # 2X(SF CONSTANT NUMBER) IN A

PUTCOM          TS      DECOUNT
                XCH     Q
                TS      DECRET
                CAF     ZERO
                TS      MPAC+6
                INDEX   DECOUNT
                XCH     XREGLP
                TS      MPAC +1
                INDEX   DECOUNT
                XCH     XREG
                TS      MPAC
                INDEX   MIXBR
                TC      +0
                TC      PUTNORM         # NORMAL NOUN
# IF MIXNOUN, PLACE ADDRESS FOR COMPONENT K INTO NOUNADD, SET EBANK BITS.
                INDEX   DECOUNT         # GET IDADDTAB ENTRY FOR COMPONENT K
                CA      IDAD1TEM        #         OF NOUN.
                MASK    LOW11           # (ECADR)SUBK FOR CURRENT COMP OF NOUN
                TC      SETNCADR        # ECADR INTO NOUNCADR. SETS EB, NOUNADD.
                EXTEND                  # C(NOUNADD) IN A UPON RETURN
                SU      DECOUNT         # PLACE (ESUBK)-K INTO NOUNADD
                TS      NOUNADD
                CCS     DECBRNCH
                TC      PUTDECSF        # + DEC
                TC      DCTSTCYC        # +0 OCTAL
                TC      SFRUTMIX        # TEST IF DEC ONLY BIT = 1. IF SO,
                TC      DPTEST          # ALARM AND RECYCLE. IF NOT, CONTINUE.
                TC      PUTCOM2         # NO DP
                                        # TEST FOR DP SCALE FOR OCT LOAD. IF SO,
                                        # +0 INTO MAJOR PART. SET NOUNADD FOR
                                        # LOADING OCTAL WORD INTO MINOR PART.
PUTDPCOM        INCR    NOUNADD         # DP (ESUBK)-K+1  OR  E+1
                CA      NOUNADD         # NOUNADD NOW SET FOR MINOR PART
                ADS     DECOUNT         # (ESUBK)+1  OR  E+1  INTO DECOUNT
                CAF     ZERO            # NOUNADD SET FOR MINOR PART
                INDEX   DECOUNT
                TS      0 -1            # ZERO MAJOR PART(ESUBK OR E)
                TC      PUTCOM2

PUTNORM         TC      SETNADD         # ECADR FROM NOUNCADR. SETS EB, NOUNADD.
                CCS     DECBRNCH
```

```
                TC      PUTDECSF        # + DEC
                TC      DCTSTCYC        # +0 OCTAL
                TC      SFRUTNOR        # TEST IF DEC ONLY BIT =1. IF SO,
                TC      DPTEST          # ALARM AND RECYCLE. IF NOT, CONTINUE.
                TC      PUTCOM2 -4      # NO DP
                CAF     ZERO            # DP
                TS      DECOUNT
                TC      PUTDPCOM

                CA      NNADTEM
                AD      ONE             # IF NNADTEM = -1, CHANNEL TO BE SPECIFIED
                EXTEND
                BZF     CHANLOAD
PUTCOM2         XCH     MPAC
                TC      DECRET

                EBANK=  DSPCOUNT
GTSFINLC        2CADR   GTSFIN

CHANLOAD        CS      SEVEN           # DONT LOAD CHAN 7. (IT = SUPERBANK).
                AD      NOUNCADR
                EXTEND
                BZF     LOADLV
                CA      NOUNCADR
                MASK    LOW9
                XCH     MPAC
                EXTEND
                INDEX   MPAC
                WRITE   0
                TC      LOADLV

# PUTDECSF FINDS MIXBR AND DECOUNT STILL SET FROM PUTCOM

PUTDECSF        TC      SFCONUM         # 2X(SF CON NUMB) IN A
                TS      SFTEMP1
                EXTEND                  # SWITCH BANKS TO SF CONSTANT TABLE
                DCA     GTSFINLC        # READING ROUTINE.
                DXCH    Z               # LOADS SFTEMP1, SFTEMP2.
                INDEX   MIXBR
                TC      +0
                TC      PUTSFNOR
                TC      SFRUTMIX
                TC      PUTDCSF2
PUTSFNOR        TC      SFRUTNOR

PUTDCSF2        INDEX   A
                CAF     SFINTABR
```

```
                TC      BANKJUMP        # SWITCH BANKS FOR EXPANSION ROOM
SFINTABR        CADR    GOALMCYC        # ALARM AND RECYCLE IF DEC LOAD
                                        # WITH OCTAL ONLY NOUN.
                CADR    BINROUND
                CADR    DEGINSF
                CADR    ARTHINSF
                CADR    DPINSF
                CADR    DPINSF2
                CADR    DSPALARM        # LRPOSOUT CANT BE LOADED.
                CADR    DPINSF          # SAME AS ARITHDP1
                CADR    HMSIN
                CADR    DSPALARM        # MIN/SEC CANT BE LOADED.
                CADR    DPINSF4
                CADR    ARTIN1SF
                CADR    DSPALARM        # 2INTOUT CANT BE LOADED.
                CADR    DEGINSF         # TESTS AT END FOR 360-CDU
ENDRUTIN        EQUALS

# SCALE FACTORS FOR THOSE ROUTINES NEEDING THEM ARE AVAILABLE IN SFTEMP1.
# ALL SFIN ROUTINES USE MPAC MPAC+1. LEAVE RESULT IN A. END WITH TC DECRET.

                SETLOC  ENDDPDEC +1

                COUNT*  $$/PIN
# DEGINSF APPLIES 1000/180 = 5.55555(10) = 5.43434(8)

DEGINSF         TC      DMP             # SF ROUTINE FOR DEC DEGREES
                ADRES   DEGCON1         # MULT BY 5.5 5(10)X2EXP-3
                CCS     MPAC +1         # THIS ROUNDS OFF MPAC+1 BEFORE SHIFT
                CAF     BIT11           # LEFT 3, AND CAUSES 360.00 TO OF/UF
                TC      +2              # WHEN SHIFTED LEFT AND ALARM.
                CS      BIT11
                AD      MPAC +1
                TC      2ROUND +2
                TC      TPSL1           # LEFT 1
DEGINSF2        TC      TPSL1           # LEFT 2
                TC      TESTOFUF
                TC      TPSL1           # RETURNS IF NO OF/UF (LEFT3)
                CCS     MPAC
                TC      SIGNFIX         # IF +, GO TO SIGNFIX
                TC      SIGNFIX         # IF +0, GO TO SIGNFIX
                COM                     # IF -, USE -MAGNITUDE +1
                TS      MPAC            # IF -0, USE +0
SIGNFIX         CCS     MPAC+6
                TC      SGNTO1          # IF OVERFLOW
                TC      ENDSCALE        # NO OVERFLOW/UNDERFLOW
                CCS     MPAC            # IF UF FORCE SIGN TO 0 EXCEPT -180
                TC      CCSHOLE
```

```
                TC      NEG180
                TC      +1
                XCH     MPAC
                MASK    POSMAX
                TS      MPAC
ENDSCALE        INDEX   MIXBR           # IF ROUTINE NO. IS NOT CDU DEGREES,
                TC      +0              # THEN THIS IS 360 - CDU DEGREES
                TC      +3              # AND ANGLE IN MPAC MUST BE REPLACED
                TC      SFMIXCAL        # BY 360 DEGREES MINUS ITSELF.
MIXBACK         TC      +2
                TC      SFNORCAL
NORBACK         CS      A
                AD      BIT2
                EXTEND
                BZF     +2
                TC      360-CDU
ENDSCAL1        TC      POSTJUMP
                CADR    PUTCOM2

SFMIXCAL        TC      BANKCALL
                CADR    SFRUTMIX
                TC      MIXBACK

SFNORCAL        TC      BANKCALL
                CADR    SFRUTNOR
                TC      NORBACK

NEG180          CS      POSMAX
                TC      ENDSCALE -1

SGNTO1          CS      MPAC            # IF OF FORCE SIGN TO 1
                MASK    POSMAX
                CS      A
                TC      ENDSCALE -1

DEGCON1         2DEC    5.555555555 B-3

ARTHINSF        TC      DMP             # SCALES MPAC, +1 BY SFTEMP1, SFTEMP2.
                ADRES   SFTEMP1         # ASSUMES POINT BETWEEN HI AND LO PARTS
                XCH     MPAC +2         # OF SFCON. SHIFTS RESULTS LEFT BY 14.
                XCH     MPAC +1         # (BY TAKING RESULTS FROM MPAC+1, MPAC+2)
                XCH     MPAC
                EXTEND
                BZF     BINROUND
                TC      ALMCYCLE        # TOO LARGE A LOAD. ALARM AND RECYCLE.
BINROUND        TC      2ROUND
                TC      TESTOFUF
                TC      ENDSCAL1        # RETURNS IF NO OF/UF
```

```
ARTIN1SF        TC      DMP             # SCALES MPAC, +1 BY SFTEMP1, SFTEMP2.
                ADRES   SFTEMP1         # ROUNDS MPAC+1 INTO MPAC.
                TC      BINROUND

DPINSF          TC      DMP             # SCALES MPAC, MPAC +1 BY SFTEMP1,
                ADRES   SFTEMP1         # SFTEMP.  STORES LOW PART OF RESULT
                XCH     MPAC +2         # IN (E SUBK) +1 OR E+1
                DOUBLE
                TS      MPAC +2
                CAF     ZERO
                AD      MPAC +1
                TC      2ROUND +2
                TC      TESTOFUF
                INDEX   MIXBR           # RETURNS IF NO OF/UF
                TC      +0
                TC      DPINORM
                CA      DECOUNT         # MIXED NOUN
DPINCOM         AD      NOUNADD         #       MIXED           NORMAL
                TS      Q               #       E SUBK          E
                XCH     MPAC +1
                INDEX   Q
                TS      1               # PLACE LOW PART IN
                TC      ENDSCAL1        # (E SUBK) +1   MIXED

DPINORM         CAF     ZERO            # E +1          NORMAL
                TC      DPINCOM

DPINSF2         TC      DMP             # ASSUMES POINT BETWEEN BITS 7-8 OF HIGH
                ADRES   SFTEMP1         # PART OF SF CONST. DPINSF2 SHIFTS RESULTS
                CAF     SIX             # LEFT BY 7, ROUNDS MPAC+2 INTO MPAC+1
                TC      TPLEFTN         # SHIFT LEFT 7.
                TC      DPINSF +2

DPINSF4         TC      DMP             # ASSUMES POINT BETWEEN BITS 11-12 OF HIGH
                ADRES   SFTEMP1         # PART OF SF CONST. DPINSF2 SHIFTS RESULTS
                CAF     TWO             # LEFT BY 3, ROUNDS MPAC+2 INTO MPAC+1.
                TC      TPLEFTN         # SHIFT LEFT 3.
                TC      DPINSF +2

TPLEFTN         XCH     Q               # SHIFTS MPAC, +1, +2 LEFT N. SETS OVFIND
                TS      SFTEMP2         # TO +1 FOR OF, -1 FOR UF.
                XCH     Q               # CALL WITH N-1 IN A.
LEFTNCOM        TS      SFTEMP1         #       LOOP TIME .37 MSEC.
                TC      TPSL1
                CCS     SFTEMP1
                TC      LEFTNCOM
```

```
                TC      SFTEMP2

2ROUND          XCH     MPAC     +1
                DOUBLE
                TS      MPAC    +1
                TC      Q               # IF MPAC+1 DOES NOT OF/UF
                AD      MPAC
                TS      MPAC
                TC      Q               # IF MPAC DOES NOT OF/UF
                TS      MPAC+6
2RNDEND         TC      Q

TESTOFUF        CCS     MPAC+6          # RETURNS IF NO OF/UF
                TC      ALMCYCLE        # OF    ALARM AND RECYCLE.
                TC      Q
                TC      ALMCYCLE        # UF    ALARM AND RECYCLE.

                SETLOC  ENDSPMIN +1

                COUNT*  $$/PIN
HMSIN           TC      ALL3DEC         # IF ALL 3 WORDS WERE NOT LOADED, ALARM.
                TC      DMP             # XREG, XREGLP (=HOURS) WERE ALREADY PUT
                ADRES   WHOLECON        # INTO MPAC, MPAC+1.
                TC      RND/TST         # ROUND OFF TO WHOLE HRS IN MPAC+1.
                CAF     ZERO            # ALARM IF MPAC NON ZERO (G/ 16383).
                TS      MPAC    +2
                CAF     HRCON
                TS      MPAC
                CAF     HRCON   +1
                XCH     MPAC    +1
                TC      SHORTMP
                TC      MPACTST         # ALARM IF MPAC NON ZERO (G/ 745)
                DXCH    MPAC    +1      # STORE HOURS CONTRIBUTION
                DXCH    HITEMIN
                CA      YREG            # PUT YREG, YREGLP INTO MPAC, +1.
                LXCH    YREGLP
                DXCH    MPAC
                TC      DMP
                ADRES   WHOLECON
                TC      RND/TST         # ROUND OFF TO WHOLE MIN IN MPAC+1
                CS      59MIN           # ALARM IF MPAC NON ZERO (G/16383)
                TC      SIZETST         # ALARM IF MPAC+1 G/ 59MIN
                XCH     MPAC    +1
                EXTEND
                MP      MINCON          # LEAVES MINUTES CONTRIBUTION IN A,L
                DAS     HITEMIN         # ADD IN MINUTES CONTRIBUTION
                EXTEND                  # IF THIS DAS OVEFLOWS, G/ 745 HR, 39MIN
```

```
                BZF     +2
                TC      ALMCYCLE
                CA      ZREG            # PUT ZREG, ZREGLP INTO MPAC, +1.
                LXCH    ZREGLP
                DXCH    MPAC
                TC      DMP
                ADRES   WHOLECON
                TC      RND/TST         # ROUND OFF TO WHOLE CENTI-SEC IN MPAC+1
                CS      59.99SEC        # ALARM IF MPAC NON ZERO (G/163.83 SEC)
                TC      SIZETST         # ALARM IF MPAC+1 G/59.99 SEC
                DXCH    HITEMIN         # ADD IN SECONDS CONTRIBUTION
                DAS     MPAC            # IF THIS DAS OVERFLOWS,
                EXTEND                  # G/ 745 HR, 39 MIN, 14.59 SEC.
                BZF     +2
                TC      ALMCYCLE        # ALARM AND RECYCLE
                CAF     ZERO
                TS      MPAC +2
                TC      TPAGREE
                DXCH    MPAC
                INDEX   NOUNADD
                DXCH    0
                TC      POSTJUMP
                CADR    LOADLV

WHOLECON        OCT     00006           # (10EXP5/2EXP14)2EXP14
                OCT     03240
HRCON           OCT     00025           # 1 HOUR IN CENTI-SEC
                OCT     37100
MINCON          OCT     13560           # 1 MINUTE IN CENTI-SEC
59MIN           OCT     00073           # 59 AS WHOLE
59.99SEC        OCT     13557           # 5999 CENTI-SEC

RND/TST         XCH     MPAC +2         # ROUNDS MPAC+2 INTO MPAC+1.
                DOUBLE                  # ALARMS IF MPAC NOT 0
                TS      MPAC +2
                CAF     ZERO
                AD      MPAC +1
                TS      MPAC +1
                CAF     ZERO
                AD      MPAC            # CANT OVFLOW
                XCH     MPAC
MPACTST         CCS     MPAC            # ALARM IF MPAC NON ZERO
                TC      ALMCYCLE        # ALARM AND RECYCLE.
                TC      Q
                TC      ALMCYCLE        # ALARM AND RECYCLE.
                TC      Q

SIZETST         TS      MPAC +2         # CALLED WITH - CON IN A
                CCS     MPAC +1         # GET MAG OF MPAC+1
```

```
                AD      ONE
                TCF     +2
                AD      ONE
                AD      MPAC +2
                EXTEND                  # MAG OF MPAC+1 - CON
                BZMF    +2
                TC      ALMCYCLE        # MAG OF MPAC+1 G/ CON. ALARM AND RECYCLE.
                TC      Q               # MAG OF MPAC+1 L/= CON

# ALL3DEC TESTS THAT ALL 3 WORDS ARE LOADED IN DEC (FOR HMSIN).
# ALARM IF NOT. (TEST THAT BITS 3,4,5 OF DECBRNCH ARE ALL = 1).

ALL3DEC         CS      OCT34BAR        # GET BITS 3,4,5 IN A
                MASK    DECBRNCH        # GET BITS 3,4,5 OF DECBRNCH IN A
                AD      OCT34BAR        # BITS 3,4,5 OF DECBRNCH MUST ALL = 1
                CCS     A
                TC      FORCEV25
OCT34BAR        OCT     77743
                TC      FORCEV25
                TC      Q

FORCEV25        CS      OCT31           # FORCE VERB 25 TO BE EXECUTED BY RECYCLE
                TS      VERBSAVE        # IN CASE OPERATOR EXECUTED A LOWER LOAD
                TC      ALMCYCLE        # VERB.  ALARM AND RECYCLE.
ENDHMSS         EQUALS
```

```
# MONITOR ALLOWS OTHER KEYBOARD ACTIVITY. IT IS ENDED BY VERB TERMINATE,
# VERB PROCEED WITHOUT DATA, VERB RESEQUENCE,
# ANOTHER MONITOR, OR ANY NVSUB CALL THAT PASSES THE DSPLOCK (PROVIDED
# THAT THE OPERATOR HAS SOMEHOW ALLOWED THE ENDING OF A MONITOR WHICH
# HE HAS INITIATED THROUGH THE KEYBOARD).
#
# MONITOR ACTION IS SUSPENDED, BUT NOT ENDED, BY ANY KEYBOARD ACTION,
# EXCEPT ERROR LIGHT RESET. IT BEGINS AGAIN WHEN KEY RELEASE IS PERFORMED.
# MONITOR SAVES THE NOUN AND APPROPRIATE DISPLAY VERB IN MONSAVE. IT SAVES
# NOUNCADR IN MONSAVE1, IF NOUN = MACHINE CADR TO BE SPECIFIED. BIT 15 OF
# MONSAVE1 IS THE KILL MONITOR SIGNAL (KILLER BIT). BIT 14 OF MONSAVE1
# INDICATES THE CURRENT MONITOR WAS EXTERNALLY INITIATED (EXTERNAL
# MONITOR BIT). IT IS TURNED OFF BY RELDSP AND KILMONON.
#
# MONSAVE INDICATES IF MONITOR IS ON (+=ON, +0=OFF)
# IF MONSAVE IS +, MONITOR ENTERS NO REQUEST, BUT TURNS KILLER BIT OFF.
# IF MONSAVE IS +0, MONITOR ENTERS REQUEST AND TURNS KILLER BIT OFF.
#
# NVSUB (IF EXTERNAL MONITOR BIT IS OFF), VB=PROCEED WITHOUT DATA,
# VB=RESEQUENCE, AND VB=TERMINATE TURN KILL MONITOR BIT ON.
#
# IF KILLER BIT IS ON, MONREQ ENTERS NO FURTHER REQUESTS, ZEROS MONSAVE
# AND MONSAVE1 (TURNING OFF KILLER BIT AND EXTERNAL MONITOR BIT).
#
# MONITOR DOESNT TEST FOR MATBS SINCE NVSUB CAN HANDLE INTERNAL MATBS NOW.

                SETLOC  ENDRUTIN

                COUNT*  $$/PIN
MONITOR         CS      BIT15/14
                MASK    NOUNCADR
MONIT1          TS      MPAC +1         # TEMP STORAGE
                CS      ENTEXIT
                AD      ENDINST
                CCS     A
                TC      MONIT2
BIT15/14        OCT     60000
                TC      MONIT2
                CAF     BIT14           # EXTERNALLY INITIATED MONITOR.
                ADS     MPAC +1         # SET BIT 14 FOR MONSAVE1.
                CAF     ZERO
                TS      MONSAVE2        # ZERO NVMONOPT OPTIONS
MONIT2          CAF     LOW7
                MASK    VERBREG
                TC      LEFT5
                TS      CYL
                CS      CYL
                XCH     CYL
                AD      NOUNREG
                TS      MPAC            # TEMP STORAGE
                CAF     ZERO
```

```
                TS      DSPLOCK         # +0 INTO DSPLOCK SO MONITOR CAN RUN.
                CCS     CADRSTOR        # TURN OFF KR LITE IF CADRSTOR AND DSPLIST
                TC      +2              # ARE BOTH EMPTY. (LITE COMES ON IF NEW
                TC      RELDSP1         # MONITOR IS KEYED IN OVER OLD MONITOR.)
                INHINT
                CCS     MONSAVE
                TC      +5              # IF MONSAVE WAS +, NO REQUEST
                CAF     ONE             # IF MONSAVE WAS 0, REQUEST MONREQ
                TC      WAITLIST
                EBANK=  DSPCOUNT
                2CADR   MONREQ

                DXCH    MPAC            # PLACE MONITOR VERB AND NOUN INTO MONSAVE
                DXCH    MONSAVE         # ZERO THE KILL MONITOR BIT
                RELINT                  # SET UP EXTERNAL MONITOR BIT
                TC      ENTRET

MONREQ          TC      LODSAMPT        # CALLED BY WAITLIST
                CCS     MONSAVE1        # TIME IS SNATCHED N RUPT FOR NOUN 65
                TC      +4              # IF KILLER BIT = 0, ENTER REQUESTS
                TC      +3              # IF KILLER BIT = 0, ENTER REQUESTS
                TC      KILLMON         # IF KILLER BIT = 1, NO REQUESTS.
                TC      KILLMON         # IF KILLER BIT = 1, NO REQUESTS.
                CAF     MONDEL
                TC      WAITLIST        # ENTER WAITLIST REQUEST FOR MONREQ
                EBANK=  DSPCOUNT
                2CADR   MONREQ

                CAF     CHRPRIO
                TC      NOVAC           # ENTER EXEC REQUEST FOR MONDO
                EBANK=  DSPCOUNT
                2CADR   MONDO

                TC      TASKOVER

KILLMON         CAF     ZERO            # ZERO MONSAVE AND TURN KILLER BIT OFF
                TS      MONSAVE
                TS      MONSAVE1        # TURN OFF KILL MONITOR BIT.
                TC      TASKOVER        # TURN OFF EXTERNAL MONITOR BIT.
MONDEL          OCT     144             # FOR 1 SEC MONITOR INTERVALS

MONDO           CCS     MONSAVE1        # CALLED BY EXEC
                TC      +4              # IF KILLER BIT = 0, CONTINUE
                TC      +3              # IF KILLER BIT = 0, CONTINUE
                TC      ENDOFJOB        # IN CASE TERMINATE CAME SINCE LAST MONREQ
                TC      ENDOFJOB        # IN CASE TERMINATE CAME SINCE LAST MONREQ
                CCS     DSPLOCK
                TC      MONBUSY         # NVSUB IS BUSY
```

```
                CAF     LOW7
                MASK    MONSAVE
                TC      UPDATNN -1      # PLACE NOUN INTO NOUNREG AND DISPLAY IT
                CAF     MID7
                MASK    MONSAVE         # CHANGE MONITOR VERB TO DISPLAY VERB
                AD      MONREF          # -DEC10, STARTING IN BIT8
                TS      EDOP            # RIGHT 7
                CA      EDOP
                TS      VERBREG
                CAF     MONBACK         # SET RETURN TO PASTEVB AFTER DATA DISPLAY
                TS      ENTRET
                CS      BIT15/14
                MASK    MONSAVE1        # PUT ECADR INTO MPAC +2.  INTMCTBS WILL
                TS      MPAC +2         # DISPLAY IT AND SET NOUNCADR, NOUNADD,
ENDMONDO        TC      TESTNN          # EBANK.

                BLOCK   2

                SETLOC  FFTAG8
                BANK

                COUNT*  $$/PIN
PASTEVB         CAF     MID7
                MASK    MONSAVE2        # NVMONOPT PASTE OPTION
                EXTEND
                BZF     +2
                TC      PASTEOPT        # PASTE PLEASE VERB FOR NVMONOPT
                CA      MONSAVE         # PASTE MONITOR VERB - PASTE OPTION IS 0
PASTEOPT        TS      EDOP            # RIGHT 7
                CA      EDOP            # PLACE MONITOR VERB OR PLEASE VERB INTO
                TC      BANKCALL        #  VERBREG AND DISPLAY IT.
                CADR    UPDATVB -1
                CAF     ZERO            # ZERO REQRET SO THAT PASTED VERBS CAN
                TS      REQRET          #  BE EXECUTED BY OPERATOR.
                CA      MONSAVE2
                TC      BLANKSUB        # PROCESS NVMONOPT BLANK OPTION IF ANY
                TC      +1
ENDPASTE        TC      ENDOFJOB

MID7            OCT     37600

                SETLOC  ENDMONDO +1
                COUNT*  $$/PIN
MONREF          OCT     75377           # -DEC10, STARTING IN BIT8
MONBACK         ADRES   PASTEVB

MONBUSY         TC      RELDSPON        # TURN KEY RELEASE LIGHT
                TC      ENDOFJOB
```

```
# DSPFMEM IS USED TO DISPLAY (IN OCTAL) ANY FIXED REGISTER.
# IT IS USED WITH NOUN = MACHINE CADR TO BE SPECIFIED. THE FCADR OF THE
# DESIRED LOCATION IS THEN PUNCHED IN. IT HANDLES F/F (FCADR 4000-7777)
#
# FOR BANKS L/E 27, THIS IS ENOUGH.
#
# FOR BANKS G/E 30, THE THIRD COMPONENT OF NOUN 26 (PRIO, ADRES, BBCON)
# MUST BE PRELOADED WITH THE DESIRED SUPERBANK BITS (BITS 5,6,7).
#       V23N26 SHOULD BE USED.
#
# SUMMARY
# FOR BANKS L/E 27,                                 V27N01E(FCADR)E
# FOR BANKS G/E 30,     V23N26E(SUPERBITS)E         V27N01E(FCADR)E

DSPFMEM         CAF     R1D1            # IF F/F, DATACALL USES BANK 02 OR 03.
                TS      DSPCOUNT
                CA      DSPTEM1 +2      # SUPERBANK BITS WERE PRELOADED INTO
                TS      L               # 3RD COMPONENT OF NOUN 26.
                CA      NOUNCADR        # ORIGINAL FCADR LOADED STILL IN NOUNCADR.
                TC      SUPDACAL        # CALL WITH FCADR IN A, SUPERBITS IN L.
                TC      DSPOCTWO
ENDSPF          TC      ENDOFJOB
```

```
# WORD DISPLAY ROUTINES
                SETLOC  TESTOFUF +4
                COUNT*  $$/PIN
DSPSIGN         XCH     Q
                TS      DSPWDRET
                CCS     MPAC
                TC      +8D
                TC      +7
                AD      ONE
                TS      MPAC
                TC      -ON
                CS      MPAC +1
                TS      MPAC +1
                TC      DSPWDRET
                TC      +ON
                TC      DSPWDRET

DSPRND          EXTEND                          # ROUND BY 5 EXP-6
                DCA     DECROUND -1
                DAS     MPAC
                EXTEND
                BZF     +4
                EXTEND
                DCA     DPOSMAX
                DXCH    MPAC
                TC      Q

# DSPDECWD CONVERTS C(MPAC, MPAC+1) INTO A SIGN AND 5 CHAR DECIMAL
# STARTING IN LOC SPECIFIED IN DSPCOUNT. IT ROUNDS BY 5 EXP-6.

DSPDECWD        XCH     Q
                TS      WDRET
                TC      DSPSIGN
                TC      DSPRND
                CAF     FOUR
DSPDCWD1        TS      WDCNT
                CAF     BINCON
                TC      SHORTMP
TRACE1          INDEX   MPAC
                CAF     RELTAB
                MASK    LOW5
                TS      CODE
                CAF     ZERO
                XCH     MPAC +2
                XCH     MPAC +1
                TS      MPAC
                XCH     DSPCOUNT
TRACE1S         TS      COUNT
                CCS     A                       # DECREMENT DSPCOUNT EXCEPT AT +0
```

```
                TS      DSPCOUNT
                TC      DSPIN
                CCS     WDCNT
                TC      DSPDCWD1
                CS      VD1
                TS      DSPCOUNT
                TC      WDRET

                OCT     00000
DECROUND        OCT     02476

# DSPDECNR CONVERTS C(MPAC,MPAC+1) INTO A SIGN AND 5 CHAR DECIMAL
# STARTING IN LOC SPECIFIED IN DSPCOUNT. IT DOES NOT ROUND

DSPDECNR        XCH     Q
                TS      WDRET
                TC      DSPSIGN
                TC      DSPDCWD1 -1

# DSPDC2NR CONVERTS C(MPAC,MPAC+1) INTO A SIGN AND 2 CHAR DECIMAL
# STARTING IN LOC SPECIFIED IN DSPCOUNT. IT DOES NOT ROUND

DSPDC2NR        XCH     Q
                TS      WDRET
                TC      DSPSIGN
                CAF     ONE
                TC      DSPDCWD1

# DSP2DEC CONVERTS C(MPAC) AND C(MPAC+1) INTO A SIGN AND 10 CHAR DECIMAL
# STARTING IN THE LOC SPECIFIED IN DSPCOUNT.

DSP2DEC         XCH     Q
                TS      WDRET
                CAF     ZERO
                TS      CODE
                CAF     THREE
                TC      11DSPIN         # -R2 OFF
                CAF     FOUR
                TC      11DSPIN         # +R2 OFF
                TC      DSPSIGN
                CAF     R2D1
END2DEC         TC      DSPDCWD1

# DSPDECVN DISPLAYS C(A) UPON ENTRY AS A 2 CHAR DECIMAL BEGINNING IN THE
# DSP LOC SPECIFIED IN DSPCOUNT.
# C(A) SHOULD BE IN FORM N X 2EXP-14.  THIS IS SCALED TO FORM N/100 BEFORE
# DISPLAY CONVERSION.
```

```
DSPDECVN        EXTEND
                MP      VNDSPCON        # MULT BY .01
                LXCH    MPAC            # TAKE RESULTS FROM L. (MULT BY 2EXP14).
                CAF     ZERO
                TS      MPAC +1
                XCH     Q
                TS      WDRET
                TC      DSPDC2NR +3     # NO SIGN, NO ROUND, 2 CHAR

VNDSPCON        OCT     00244           # .01 ROUNDED UP

GOVNUPDT        TC      DSPDECVN        # THIS IS NOT FOR GENERAL USE. REALLY PART
                TC      POSTJUMP        # OF UPDATVB.
                CADR    UPDAT1 +2

ENDECVN         EQUALS

                SETLOC  ENDSPF +1
                COUNT*  $$/PIN
# DSPOCTWD DISPLAYS C(A) UPON ENTRY AS A 5 CHAR OCT STARTING IN THE DSP
# CHAR SPECIFIED IN DSPCOUNT. IT STOPS AFTER 5 CHAR HAVE BEEN DISPLAYED.

DSPOCTWO        TS      CYL
                XCH     Q
                TS      WDRET           # MUST USE SAME RETURN AS DSP2BIT.
                CAF     BIT14           # TO BLANK SIGNS
                ADS     DSPCOUNT
                CAF     FOUR
WDAGAIN         TS      WDCNT
                CS      CYL
                CS      CYL
                CS      CYL
                CS      A
                MASK    DSPMSK
                INDEX   A
                CAF     RELTAB
                MASK    LOW5
                TS      CODE
                XCH     DSPCOUNT
                TS      COUNT
                CCS     A               # DECREMENT DSPCOUNT EXCEPT AT +0
                TS      DSPCOUNT
                TC      POSTJUMP
                CADR    DSPOCTIN
OCTBACK         CCS     WDCNT
                TC      WDAGAIN         # +
DSPLV           CS      VD1             # TO BLOCK NUMERICAL CHARACTERS, CLEARS,
                TS      DSPCOUNT        # AND SIGNS AFTER A COMPLETED DISPLAY.
```

```
                TC      WDRET

DSPMSK          =       SEVEN

# DSP2BIT DISPLAYS C(A) UPON ENTRY AS A 2 CHAR OCT BEGINNING IN THE DSP
# LOC SPECIFIED IN DSPCOUNT BY PRE CYCLING RIGHT C(A) AND USING THE LOGIC
# OF THE 5 CHAR OCTAL DISPLAY

DSP2BIT         TS      CYR
                XCH     Q
                TS      WDRET
                CAF     ONE
                TS      WDCNT
                CS      CYR
                CS      CYR
                XCH     CYR
                TS      CYL
                TC      WDAGAIN +5

# FOR DSPIN PLACE 0/25 OCT INTO COUNT, 5 BIT RELAY CODE INTO CODE.  BOTH
# ARE DESTROYED. IF BIT14 OF COUNT IS 1, SIGN IS BLANKED WITH LEFT CHAR.
# FOR DSPIN1 PLACE 0,1 INTO BIT11 OF CODE, 2 INTO COUNT, REL ADDRESS OF
# DSPTAB ENTRY INTO DSREL.

                SETLOC  ENDECVN

                COUNT*  $$/PIN
DSPIN           XCH     Q               # CANT USE L FOR RETURN, SINCE MANYOF THE
                TS      DSEXIT          # ROTINE CALLING DSPIN USE L AS RETURN.
                CAF     LOW5
                MASK    COUNT
                TS      SR
                XCH     SR
                TS      DSREL
                CAF     BIT1
                MASK    COUNT
                CCS     A
                TC      +2              # LEFT IF COUNT IS ODD
                TC      DSPIN1 -1       # RIGHT IF COUNT IS EVEN
                XCH     CODE
                TC      SLEFT5          # DOES NOT USE CYL
                TS      CODE
                CAF     BIT14
                MASK    COUNT
                CCS     A
                CAF     TWO             # BIT14 = 1, BLANK SIGN
                AD      ONE             # BIT14 = 0, LEAVE SIGN ALONE
                TS      COUNT           # +0 INTO COUNT FOR RIGHT
```

```
                                                # +1 INTO COUNT FOR LEFT (SIGN LEFT ALONE)
                                                # +3 INTO COUNT FOR LEFT (TO BLANK SIGN)
DSPIN1          INHINT
                INDEX   DSREL
                CCS     DSPTAB
                TC      +2              # IF +
                TC      CCSHOLE
                AD      ONE             # IF -
                TS      DSMAG
                INDEX   COUNT
                MASK    DSMSK
                EXTEND
                SU      CODE
                EXTEND
                BZF     DSLV            # SAME
DFRNT           INDEX   COUNT
                CS      DSMSK           # MASK WITH 77740, 76037, 76777, OR 74037
                MASK    DSMAG
                AD      CODE
                CS      A
                INDEX   DSREL
                XCH     DSPTAB
                EXTEND
                BZMF    DSLV            # DSPTAB ENTRY WAS -
                INCR    NOUT            # DSPTAB ENTRY WAS +
DSLV            RELINT
                TC      DSEXIT

DSMSK           OCT     37
                OCT     1740
                OCT     2000
                OCT     3740

# FOR 11DSPIN, PUT REL ADDRESS OF DSPTAB ENTRY INTO A, 1 IN BIT11 OR 0 IN
# BIT11 OF CODE.

11DSPIN         TS      DSREL
                CAF     TWO
                TS      COUNT
                XCH     Q               # MUST USE SAME RETURN AS DSPIN
                TS      DSEXIT
                TC      DSPIN1

DSPOCTIN        TC      DSPIN           # SO DSPOCTWO DOESNT USE SWCALL
                CAF     +2
                TC      BANKJUMP
ENDSPOCT        CADR    OCTBACK
```

```
# DSPALARM FINDS TC NVSUBEND IN ENTRET FOR NVSUB INITIATED ROUTINES
# ABORT WITH 01501.
#
# DSPALARM FINDS TC ENDOFJOB IN ENTRET FOR KEYBOARD INITIATED ROUTINES.
# DC TC ENTRET.

PREDSPAL        CS      VD1
                TS      DSPCOUNT
DSPALARM        CS      NVSBENDL
                AD      ENTEXIT
                EXTEND
                BZF     CHARALRM +2
                CS      MONADR          # IF THIS IS A MONITOR, KILL IT
                AD      ENTEXIT
                EXTEND
                BZF     +2
                TC      CHARALRM
                TC      KILMONON
                TC      FALTON
                TC      PASTEVB         # PUT MONITOR VERB BACK IN VERBREG
CHARALRM        TC      FALTON          # NO NVSUB INITATED. TURN ON OPR ERROR
                TC      ENDOFJOB
                TC      POODOO
                OCT     01501
MONADR          GENADR  PASTEVB
NVSBENDL        TC      NVSUBEND

# ALMCYCLE TURNS ON CHECK FAIL LIGHT, REDISPLAYS THE ORIGINAL VERB THAT
# WAS EXECUTED, AND RECYCLES TO EXECUTE THE ORIGINAL VERB/NOUN COMBINATION
# THAT WAS LAST EXECUTED. USED FOR BAD DATA DURING LOAD VERBS AND BY
# MCTBS.  ALSO BY MMCHANG IF 2 NUMERICAL CHARACTERS WERE NOT PUNCHED IN
# FOR MM CODE.

                SETLOC  MID7 +1
                COUNT*  $$/PIN
ALMCYCLE        TC      FALTON          # TURN ON CHECK FAIL LIGHT.
                CS      VERBSAVE        # GET ORIGINAL VERB THAT WAS EXECUTED
                TS      REQRET          # SET FOR ENTPASO
                TC      BANKCALL        # PUTS ORIGINAL VERB INTO VERBREG AND
                CADR    UPDATVB -1      # DISPLAYS IT IN VERB LIGHTS.
                TC      POSTJUMP
ENDALM          CADR    ENTER

# MMCHANG USES NOUN DISPLAY UNTIL ENTER.  THEN IT USES MODE DISP.
# IT GOES TO MODROUT WITH THE NEW M M CODE IN A, BUT NOT DISPLAYED IN
# MM LIGHTS.
# IT DEMANDS 2 NUMERICAL CHARACTERS BE PUNCHED IN FOR NEW MM CODE.
# IF NOT, IT RECYCLES.
```

```
                SETLOC  DSP2BIT +10D

                COUNT*  $$/PIN
MMCHANG         TC      REQMM           # ENTPASHI ASSUMES THE TC REQMM AT MMCHANG
                                        # IF THIS MOVES AT ALL, MUST CHANGE
                                        # MMADREF AT ENTPASHI.
                CAF     BIT5            # OCT20 = ND2.
                AD      DSPCOUNT        # DSPCOUNT MUST = -ND2.
                EXTEND                  # DEMAND THAT 2 NUM CHAR WERE PUNCHED IN.
                BZF     +2
                TC      ALMCYCLE        # DSPCOUNT NOT= -ND2. ALARM AND RECYCLE.
                CAF     ZERO            # DSPCOUNT = -ND2.
                XCH     NOUNREG
                TS      MPAC
                CAF     ND1
                TS      DSPCOUNT
                TC      BANKCALL
                CADR    2BLANK
                CS      VD1             # BLOCK NUM CHAR IN
                TS      DSPCOUNT
                CA      MPAC
                TC      POSTJUMP
                CADR    MODROUTB        # GO THRU STANDARD LOC.

MODROUTB        =       V37
REQMM           CS      Q
                TS      REQRET
                CAF     ND1
                TS      DSPCOUNT
                CAF     ZERO
                TS      NOUNREG
                TC      BANKCALL
                CADR    2BLANK
                TC      FLASHON
                CAF     ONE
                TS      DECBRNCH        # SET FOR DEC
                TC      ENTEXIT

# VBRQEXEC ENTERS REQUEST TO EXEC FOR ANY ADDRESS WITH ANY PRIORITY.
# IT DOES ENDOFJOB AFTER ENTERING REQUEST.  DISPLAY SYST IS RELEASED.
# IT ASSUMES NOUN 26 HAS BEEN PRELOADED WITH
# COMPONENT 1  PRIORITY (BITS 10-14) BIT1=0 FOR NOVAC, BIT1=1 FOR FINDVAC.
# COMPONENT 2  JOB ADRES (12 BIT)
# COMPONENT 3  BBCON

VBRQEXEC        CAF     BIT1
                MASK    DSPTEM1
                CCS     A
```

```
                TC      SETVAC          # IF BIT1 = 1, FINDVAC
                CAF     TCNOVAC         # IF BIT1 = 0, NOVAC
REQEX1          TS      MPAC            # TC NOVAC OR TC FINDVAC INTO MPAC
                CS      BIT1
                MASK    DSPTEM1
                TS      MPAC +4         # PRIO INTO MPAC+4 AS A TEMP
REQUESTC        TC      RELDSP
                CA      ENDINST
                TS      MPAC +3         # TC ENDOFJOB INTO MPAC+3
                EXTEND
                DCA     DSPTEM1 +1      # JOB ADRES INTO MPAC+1
                DXCH    MPAC +1         # BBCON INTO MPAC+2
                CA      MPAC +4         # PRIO IN A
                INHINT
                TC      MPAC

SETVAC          CAF     TCFINDVC
                TC      REQEX1

# VBRQWAIT ENTERS REQUEST TO WAITLIST FOR ANY ADDRESS WITH ANY DELAY.
# IT DOES ENDOFJOB AFTER ENTERING REQUEST. DISPLAY SYST IS RELEASED.
# IT ASSUMES NOUN 26 HAS BEEN PRELOADED WTIH
# COMPONENT 1  DELAY (LOW BITS)
# COMPONENT 2  TASK ADRES (12 BIT)
# COMPONENT 3  BBCON

VBRQWAIT        CAF     TCWAIT
                TS      MPAC            # TC WAITLIST INTO MPAC
                CA      DSPTEM1         # TIME DELAY
ENDRQWT         TC      REQUESTC -1

# REQUESTC WILL PUT TASK ADRES INTO MPAC+1, BBCON INTO MPAC+2,
# TC ENDOFJOB INTO MPAC+3.  IT WILL TAKE TIME DELAY OUT OF MPAC+4 AND
# LEAVE IT IN A, INHINT AND TC MPAC.

                SETLOC  NVSBENDL +1
                COUNT*  $$/PIN
VBPROC          CAF     ONE             # PROCEED WITHOUT DATA
                TS      LOADSTAT
                TC      KILMONON        # TURN ON KILL MONITOR BIT
                TC      RELDSP
                TC      FLASHOFF
                TC      RECALTST        # SEE IF THERE IS ANY RECALL FROM ENDIDLE

VBTERM          CS      ONE
                TC      VBPROC +1       # TERM VERB SETS LOADSTAT NEG
```

```
# PROCKEY PERFORMS THE SAME FUNCTION AS VBPROC. IT MUST BE CALLED UNDER
# EXECUTIVE CONTROL, WITH CHRPRIO.

PROCKEY         CAF     ZERO            # SET REQRET FOR ENTER PASS 0.
                TS      REQRET
                CS      VD1             # BLOCK NUMERICAL CHARACTERS, SIGNS, CLEAR
                TS      DSPCOUNT
                TC      VBPROC

# VBRESEQ WAKES ENDIDLE AT SAME LINE AS FINAL ENTER OF LOAD (L+3).
# (MAIN USE IS INTENDED AS RESPONSE TO INTERNALLY INITIATED FLASHING
#  DISPLAYS IN ENDIDLE. SHOULD NOT BE USED WITH LOAD VERBS, PLEASE PERFORM,
#  OR PLEASE MARK VERBS BECAUSE THEY ALREADY USE L+3 IN ANOTHER CONTEXT.)

VBRESEQ         CS      ZERO            # MAKE IT LOOK LIKE DATA IN.
                TC      VBPROC +1

# FLASH IS TURNED OFF BY PROCEED WITHOUT DATA, TERMINATE, REQUEQUENCE,
# END OF LOAD.
```

```
# KEY RELEASE ROUTINE
#
# THIS ROUTINE ALWAYS TURNS OFF THE UPACT LIGHT AND ALWAYS CLEARS DSPLOCK.
#
# THE HIGHEST PRIORITY FUNCTION OF THE KEY RELEASE BUTTON IS THE
# UNSUSPENDING OF A SUSPENDED MONITOR WHICH WAS EXTERNALLY INITIATED.
# THIS FUNCTION IS ACCOMPLISHED BY CLEARING DSPLOCK AND TURNING OFF
# THE KEY RELEASE LIGHT IF BOTH DSPLIST AND CADRSTOR ARE EMPTY.
#
# IF NO SUCH MONITOR EXISTS, THEN RELDSP IS EXECUTED TO CLEAR DSPLOCK
# AND THE EXTERNAL MONITOR BIT (FREEING THE DISPLAY SYSTEM FOR INTERNAL
# USE), TURN OFF THE KEY RELEASE LIGHT, AND WAKE UP ANY JOB IN DSPLIST.
#
# IN ADDITION IF THERE IS A JOB IN ENDIDLE, THEN CONTROL IS TRANSFERRED
# TO PINBRNCH (IN DISPLAY INTERFACE ROUTINE) TO RE-EXECUTE THE SERIES OF
# NVSUB CALLS ETC. THAT PRECEDED THE ENDIDLE CALL STILL AWAITING RESPONSE.
# THIS FEATURE IS INTENDED FOR USE WHEN THE OPERATOR HAS BEEN REQUESTED TO
# RESPOND TO SOME INTERNAL ACTION THAT USED ENDIDLE, BUT HE HAS WRITTEN
# OVER THE INFORMATION ON THE DISPLAY PANEL BY SOME DISPLAYS OF HIS OWN
# INITIATION WHICH DO NOT SERVE AS RESPONSES. HITTING KEYRLSE WILL
# RE-ESTABLISH THE DISPLAYS TO THE STATE THEY WERE IN BEOFRE HE OBSCURED
# THEM, SO THAT HE CAN SEE THE WAITING REQUEST. THIS WORKS ONLY FOR
# INTERNAL PROGRAMS THAT USED ENDIDLE THROUGH MARGARETS DISPLAY
# SUBROUTINES.

VBRELDSP        CS      BIT3
                EXTEND
                WAND    DSALMOUT        # TURN OFF UPACT LITE
                CCS     21/22REG        # OLD DSPLOCK
                CAF     BIT14
                MASK    MONSAVE1        # EXTERNAL MONITOR BIT (EMB)
                CCS     A
                TC      UNSUSPEN        # OLD DSPLOCK AND EMB BOTH 1, UNSUSPEND.
TSTLTS4         TC      RELDSP          # NOT UNSUSPENDING EXTERNAL MONITOR,
                CCS     CADRSTOR        #       RELEASE DISPLAY SYSTEM AND
                TC      +2              #       DO RE-ESTABLISH IF CADRSTOR IS FULL.
                TC      ENDOFJOB
                TC      POSTJUMP
                CADR    PINBRNCH
UNSUSPEN        CAF     ZERO            # EXTERNAL MONITOR IS SUSPENDED,
                TS      DSPLOCK         #       JUST UNSUSPEND IT BY CLEARING DSPLOCK.
                CCS     CADRSTOR        #       TURN KEY RELEASE LIGHT OFF IF BOTH
                TC      ENDOFJOB        #       CADRSTOR AND DSPLIST ARE EMPTY.
                TC      RELDSP1
                TC      ENDOFJOB

ENDRELDS        EQUALS
```

```
# NVSUB IS USED FOR SUBROUTINE CALLS FROM WITHIN COMPUTER. IT CAN BE
# USED TO CALL THE COMBINATION OF ANY DISPLAY, LOAD, OR MONITOR VERB
# TOGETHER WITH ANY NOUN AVAILABLE TO THE KEYBOARD.
# PLACE OVVVVVVVNNNNNNN INTO A.
# V'S ARE THE 7-BIT VERB CODE.  N'S ARE THE 7-BIT NOUN CODE.
#
# IF NVSUB IS CALLED WTIH THE FOLLOWING NEGATIVE NUMBERS (RATHER THAN THE
# VERB-NOUN CODE) IN A, THEN THE DISPLAY IS BLANKED AS FOLLOWS -
#  -4 FULL BLANK, -3 LEAVE MODE, -2 LEAVE MODE AND VERB, -1 BLANK R'S ONLY.
#
# NVSUB CAN BE USED WTIH MACH CADR TO BE SPEC BY PLACING THE CADR INTO
# MPAC+2 BEFORE THE STANDARD NVSUB CALL.
#
# NVSUB RETURNS TO 2+ CALLING LOC AFTER PERFORMING TASK, IF DISPLAY
# SYSTEM IS AVAIALBLE. THE NEW NOUN AND VERB CODES ARE DISPLAYED.
# IF V'S =0, THE NEW NOUN CODE IS DISPLAYED ONLY (RETURN WITH NO FURTHER
# ACTION). IF N'S =0, THE NEW VERB CODE IS DISPLAYED ONLY (RETURN WITH NO
# FURTHER ACTION).
#
# IT RETURNS TO 1+ CALLING LOC WITHOUT PERFORMING TASK, IF DISPLAY
# SYSTEM IS BLOCKED (NOTHING IS DISPLAYED IN THIS CASE).
# IT DOES TC ABORT (WITH OCT 01501) IF IT ENCOUNTERS A DISPLAY PROGRAM
# ALARM CONDITION BEFORE RETURN TO CALLER.
#
# THE DISPLAY SYSTEM IS BLOCKED BY THE DEPRESSION OF ANY
# KEY, EXCEPT ERROR LIGHT RESET.
# IT IS RELEASED BY THE KEY RELEASE BUTTON, ALL EXTENDED VERBS,
# PROCED WITHOUT DATA, TERMINATE, RESEQUENCE, INITIALIZE EXECUTIVE,
# RECALL PART OF RECALTST IF ENDIDLE WAS USED,
# VB = REQUEST EXECUTIVE, VB = REQUEST WAITLIST,
# MONITOR SET UP.
#
# THE DISPLAY SYSTEM IS ALSO BLOCKED BY THE EXTERNAL MONITOR BIT, WHICH
# INDICATES AND EXTERNALLY INITIATED MONITOR IS RUNNING (SEE MONITOR).
#
# A NVSUB CALL THAT PASSES DSPLOCK AND THE EXTERNAL MONITOR BIT ENDS OLD
# MONITOR.
#
# DSPLOCK IS THE INTERLOCK FOR USE OF KEYBOARD AND DISPLAY SYSTEM WHICH
# LOCKS OUT INTERNAL USE WHENEVER THERE IS EXTERNAL KEYBOARD ACTION.
#
# NVSUB SHOULD BE USED TWICE IN SUCCESSION FOR 'PLEASE PERFORM' SITUATIONS
# (SIMILARLY FOR PLEASE MARK).  FIRST PLACE THE CODED NUMBER FOR WHAT
# ACTION IS DESIRED OF OPERATOR INTO THE REGISTERS REFERRED TO BY THE
# 'CHECKLIST' NOUN. GO TO NVSUB WITH A DISPLAY VERB AND THE 'CHECKLIST'
# NOUN.  GO TO NVSUB AGAIN WTIH THE 'PLEASE PERFORM' VERB AND ZEROS IN THE
# LOW 7 BITS. THIS 'PASTES UP' THE 'PLEASE PERFORM' VERB INTO THE VERB
# LIGHTS.
#
# NVMONOPT IS AN ENTRY SIMILAR TO NVSUB, BUT REQUIRING AN ADDITIONAL
```

```
# PARAMETER IN L. IT SHOULD BE USED ONLY WITH A MONITOR VERB-NOUN CODE IN
# A. AFTER EACH MONITOR DISPLAY A *PLEASE* VERB WILL BE PASED INT THE VERB
# LIGHTS OR DATA WILL BE BLANKED (OR BOTH) ACCORDING TO THE OPTIONS
# SPECIFIED IN L. IF BITS 8-14 OF L ARE OTHER THAN ZERO, THEN THEY WILL
# BE INTERPRETED AS A VERB CODE AND PASTED IN THE VERB LIGHTS. (THIS VERB
# CODE SHOULD DESIGNATE ONE OF THE *PLEASE* VERBS.)  IF BITS 1-3 OF L ARE
# OTHER THAN ZERO, THEN THEY WILL BE USED TO BLANK DATA BY BEING FED TO
# BLANKSUB.  IF NVMONOPT IS USED WITH A VERB OTHER THAN A MONITOR VERB,
# THE PARAMETER IN L HAS NO EFFECT.
#
# NVSUB IN FIXED-FIXED PLACES 2+CALLING LOC INTO NVQTEM, TC NVSUBEND INTO
# ENTRET. (THIS WILL RESTORE OLD CALLING BANK BITS)

                SETLOC  ENDALM +1

                COUNT*  $$/PIN
NVSUB           LXCH    7               # ZERO NVMONOPT OPTIONS
NVMONOPT        TS      NVTEMP
                CAF     BIT14
                MASK    MONSAVE1        # EXTERNAL MONITOR BIT
                AD      DSPLOCK
                CCS     A
                TC      Q               # DSP SYST BLOCKED, RET TO 1+ CALLING LOC
                CAF     ONE             # DSP SYST AVAILABLE.
NVSBCOM         AD      Q
                TS      NVQTEM          # 2+ CALLING LOC INTO NVQTEM
                LXCH    MONSAVE2        # STORE NVMONOPT OPTIONS
                TC      KILMONON        # TURN ON KILL MONITOR BIT
NVSUBCOM        CAF     NVSBBBNK

                XCH     BBANK
                EXTEND                  # SAVE OLD SUPERBITS
                ROR     SUPERBNK
                TS      NVBNKTEM
                CAF     PINSUPBT
                EXTEND
                WRITE   SUPERBNK
                TC      NVSUBB          # GO TO NVSUB1 THRU STANDARD LOC
                EBANK=  DSPCOUNT
NVSBBBNK        BBCON   NVSUB1

PINSUPBT        =       NVSBBBNK        # CONTAINS THE PINBALL SUPERBITS.

NVSUBEND        DXCH    NVQTEM          # NVBNKTEM MUST = NVQTEM+1
                TC      SUPDXCHZ        # DTCB WITH SUPERBIT SWITCHING

                SETLOC  ENDRQWT +1

                COUNT*  $$/PIN

# BLANKDSP BLANKS DISPLAY ACCORDING TO OPTION NUMBER IN NVTEMP AS FOLLOWS
```

```
#  -4 FULL BLANK, -3 LEAVE MODE, -2 LEAVE MODE AND VERB, -1 BLANK R'S ONLY.

BLANKDSP        AD      SEVEN           # 7,8,9, OR 10 (A HAD 0,1,2,OR 3)
                INHINT
                TS      CODE            # BLANK SPECIFIED DSPTABS
                CS      BIT12
                INDEX   CODE
                XCH     DSPTAB
                CCS     A
                INCR    NOUT
                TC      +1
                CCS     CODE
                TC      BLANKDSP +2
                RELINT
                INDEX   NVTEMP
                TC      +5
                TC      +1              # NVTEMP HAS    -4 (NEVER TOUCH MODREG)
                TS      VERBREG         #               -3
                TS      NOUNREG         #               -2
                TS      CLPASS          #               -1
                CS      VD1
                TS      DSPCOUNT
                TC      FLASHOFF        # PROTECT AGAINS INVISIBLE FLASH
                TC      ENTSET -2       # ZEROS REQRET

NVSUB1          CAF     ENTSET          # IN BANK
                TS      ENTRET          # SET RETURN TO NVSUBEND
                CCS     NVTEMP          # WHAT NOW
                TC      +4              # NORMAL NVSUB CALL (EXECUTE VN OR PASTE)
                TC      GODSPALM
                TC      BLANKDSP        # BLANK DISPLAY AS SPECIFIED
                TC      GODSPALM
                CAF     LOW7
                MASK    NVTEMP
                TS      MPAC +3         # TEMP FOR NOUN (CANT USE MPAC. DSPDECVN
                CA      NVTEMP          #               USES MPAC, +1, +2).
                TS      EDOP            # RIGHT 7
                CA      EDOP
                TS      MPAC +4         # TEMP FOR VERB (CANT USE MPAC+1. DSPDECVN
                                        #               USES MPAC, +1, +2).
                CCS     MPAC +3         # TEST NOUN
                TC      NVSUB2          # IF NOUN NOT +0, GO ON
                CA      MPAC +4
                TC      UPDATVB -1      # IF NOUN = +0, DISPLAY VERB, THEN RETURN
                CAF     ZERO            # XERO REQRET SO THAT PASTED VERBS CAN
                TS      REQRET          # BE EXECUTED BY OPERATOR.
ENTSET          TC      NVSUBEND
NVSUB2          CCS     MPAC +4         # TEST VERB
                TC      +4              # IF VERB NOT +0, GO ON
                CA      MPAC +3
```

```
                TC      UPDATNN -1      # IF VERB = +0, DISPLAY NOUN, THEN RETURN
                TC      NVSUBEND
                CA      MPAC +2         # TEMP FOR MACH CADR TO BE SPEC. (DSPDECVN
                TS      MPAC +5         #       USES MPAC, +1, +2)
                CA      MPAC +4
                TC      UPDATVB -1      # IF BOTH NOUN AND VERB NOT +0, DISPLAY
                CA      MPAC +3         # BOTH AND GO TO ENTPASO.
                TC      UPDATNN -1
                CAF     ZERO
                TS      LOADSTAT        # SET FOR WAITING FOR DATA CONDITION
                TS      CLPASS
                TS      REQRET          # SET REQRET FOR PASS 0.
                CA      MPAC +5         # RESTORES MACH CADR TO BE SPEC TO MPAC+2
                TS      MPAC +2         # FOR USE IN INTMCTBS (IN ENTPASO).
ENDNVSB1        TC      ENTPASO

# IF INTERNAL MACH CADR TO BE SPECIFIED, MPAC+2 WILL BE PLACED INTO
# NOUNCADR IN ENTPASO (INTMCTBS).

                SETLOC  NVSUBEND +2
                COUNT*  $$/PIN
                                        # FORCE BIT 15 OF MONSAVE1 TO 1.
KILMONON        CAF     BIT15           #       THIS IS THE KILL MONITOR BIT.
                TS      MONSAVE1        # TURN OFF BIT 14, THE EXTERNAL
                                        #       MONITOR BIT.
                TC      Q

# LOADSTAT      +0      INACTIVE (WAITING FOR DATA). SET BY NVSUB
#               +1      PROCEED NO DATA. SET BY SPECIAL VERB
#               -1      TERMINATE. SET BY SPECIAL VERB.
#               -0              DATA IN         SET BY END OF LOAD ROUTINE
#                       OR      RESEQUENCE      SET BY VERB 32
#
# L TO ENDIDLE (FIXED FIXED)
# ROUTINES THAT REQUEST LOADS THROUGH NVSUB SHOULD USE ENDIDLE WHILE
# WAITING FOR THE DATA TO BE LOADED. ENDIDLE PUTS CURRENT JOB TO SLEEP.
# ENDIDLE CANNOT BE CALLED FROM ERASABLE OR F/F MEMORY,
# SINCE JOB SLEEP AND JOBWAKE CAN HANDLE ONLY FIXED BANKS.
# RECALTST TESTS LOADSTAT AND WAKES JOB UP TO,
#       L+1     FOR TERMINATE
#       L+2     FOR PROCEED WITHOUT DATA
#       L+3     FOR DATA IN, OR RESEQUENCE
# IT DOES NOTHING IF LOADSTAT INDICATES WAITING FOR DTA.
#
# ENDIDLE ABORTS (WITH CODE 1206) IF A SECOND JOB ATTEMPTS TO GO TO SLEEP
```

```
# IN PINBALL. IN PARTICULAR, IF AN ATTEMPT IS MADE TO GO TO ENDIDLE WHEN
# 1)     CADRSTOR NOT= +0. THIS IS THE CASE WHERE THE CAPACITY OF ENDIDLE IS
#        EXCEEDED. (+-NZ INDICATES A JOB IS ALREADY ASLEEP DUE TO ENDIDDLE.)
# 2)     DSPLIST NOT= +0. THIS INDICATES A JOB IS ALREADY ASLEEP DUE TO
#        NVSUBUSY.

ENDIDLE         LXCH    Q               # RETURN ADDRESS INTO L.
                TC      ISCADR+0        # ABORT IF CADRSTOR NOT= +0
                TC      ISLIST+0        # ABORT IF DSPLIST NOT= +0
                CA      L               # DONT SET DSPLOC TO 1 SO CAN USE
                MASK    LOW10           # ENDIDLE WITH NVSUB INITIATED MONITOR.
                AD      FBANK           # SAME STRATEGY FOR CADR AS MAKECADR.
                TS      CADRSTOR
                TC      JOBSLEEP

ENDINST         TC      ENDOFJOB

ISCADR+0        CCS     CADRSTOR        # ABORTS (CODE 01206) IF CADRSTOR NOT= +0.
                TC      DSPABORT        # RETURNS IF CADRSTOR = +0.
                TC      Q
                TC      DSPABORT

ISLIST+0        CCS     DSPLIST         # ABORTS (CODE 01206) IF DSPLIST NOT= +0.
                TC      DSPABORT        # RETURNS IF DSPLIST = +0.
                TC      Q
DSPABORT        TC      POODOO
                OCT     01206

# JAMTERM ALLOWS PROGRAMS TO PERFORM THE TERMINATE FUNCTION.
# IT DOES ENDOFJOB.

JAMTERM         CAF     PINSUPBT
                EXTEND
                WRITE   SUPERBNK
                CAF     34DEC
                TS      REQRET          # LEAVE ENTER SET FOR ENTPASSO.
                CS      VD1
                TS      DSPCOUNT
                TC      POSTJUMP
                CADR    VBTERM

34DEC           DEC     34

# JAMPROC ALLOWS PROGRAMS TO PERFORM THE PROCEED/PROCEED WITHOUT DATA
# FUNCTION. IT DOES ENDOFJOB.
```

```
JAMPROC         CAF     PINSUPBT
                EXTEND
                WRITE   SUPERBNK
                CAF     33DEC
                TS      REQRET          # LEAVE ENTER SET FOR ENTPASS0.
                CS      VD1
                TS      DSPCOUNT
                TC      POSTJUMP
                CADR    VBPROC

33DEC           DEC     33

# BLANKSUB BLANKS ANY COMBINATION OF R1, R2, R3.
# CALL WITH BLANKING CODE IN A.
# BIT1=1 BLANKS R1, BIT2=1 BLANKS R2, BIT3=1 BLANKS R3.
# ANY COMBINATION OF THESE BITS IS ACCEPTED.
#
# DSPCOUNT IS RESTORED TO STATE IT WAS IN BEFORE BLANKSUB WAS EXECUTED.

BLANKSUB        MASK    SEVEN
                TS      NVTEMP          # STORE BLANKING CODE IN NVTEMP.
                CAF     BIT14
                MASK    MONSAVE1        # EXTERNAL MONITOR BIT
                AD      DSPLOCK
                CCS     A
                TC      Q               # DSP SYST BLOCKED. RET TO 1+ CALLING LOC
                INCR    Q               # DSP SYST AVAILABLE
                                        # SET RETURN FOR 2+ CALLING LOC
                CCS     NVTEMP
                TCF     +2
                TC      Q               # NOTHING TO BLANK. RET TO 2+ CALLING LOC
                LXCH    Q               # SET RETURN FOR 2 + CALLING LOC
                CAF     BLNKBBNK
                XCH     BBANK
                EXTEND
                ROR     SUPERBNK        # SAVE OLD SUPERBITS.
                DXCH    BUF
                CAF     PINSUPBT
                EXTEND
                WRITE   SUPERBNK
                TC      BLNKSUB1

                EBANK=  DSPCOUNT
BLNKBBNK        BBCON   BLNKSUB1
ENDBLFF         EQUALS

                SETLOC  ENDRELDS
                COUNT*  $$/PIN
BLNKSUB1        CA      DSPCOUNT        # SAVE OLD DSPCOUNT FOR LATER RESTORATION
```

```
                TS      BUF +2
                CAF     BIT1            # TEST BIT1. SEE IF R1 TO BE BLANKED.
                TC      TESTBIT
                CAF     R1D1
                TC      5BLANK -1
                CAF     BIT2            # TEST BIT2. SEE IF R2 TO BE BLANKED.
                TC      TESTBIT
                CAF     R2D1
                TC      5BLANK -1
                CAF     BIT3            # TEST BIT3. SEE IF R3 TO BE BLANKED.
                TC      TESTBIT
                CAF     R3D1
                TC      5BLANK -1
                CA      BUF +2          # RESTORE DSPCOUNT TO STATE IT HAD
                TS      DSPCOUNT        #       BEFORE BLANKSUB.
                DXCH    BUF             # CALL L+2 DIRECTLY.
                TC      SUPDXCHZ +1     # DTCB WITH SUPERBIT SWITCHING

TESTBIT         MASK    NVTEMP          # NVTEMP CONTAINS BLANKING CODE.
                CCS     A
                TC      Q               # IF CURRENT BIT = 1, RETURN TO L+1.
                INDEX   Q               # IF CURRENT BIT = 0, RETURN TO L+3.
                TC      2

ENDBSUB1        EQUALS

# DSPMM DOES NOT DISPLAY MODREG DIRECTLY. IT PUTS IN EXEC REQUEST WITH
# PRIO 30000 FOR DSPMMJB AND RETURNS TO CALLER.
#
# IF MODREG CONTAINS -0, DSPMMJB BLANKS THE MODE LIGHTS.
#
# DSPMM MUST BE IN BANK 27 OR LOWER, SO IT CAN BE CALLED VIA BANKCALL.

                BANK    7
                SETLOC  PINBALL4
                BANK

                COUNT*  $$/PIN
DSPMM           XCH     Q
                TS      MPAC
                INHINT
                CAF     CHRPRIO
                TC      NOVAC
                EBANK=  DSPCOUNT
                2CADR   DSPMMJB

                RELINT
ENDSPMM         TC      MPAC
```

```
# DSPMM  PLACE MAJOR MODE CODE INTO MODREG

                SETLOC  ENDBSUB1

                COUNT*  $$/PIN
DSPMMJB         CAF     MD1             # GETS HERE THRU DSPMM
                XCH     DSPCOUNT
                TS      DSPMMTEM        # SAVE DSPCOUNT
                CCS     MODREG
                AD      ONE
                TC      DSPDECVN        # IF MODREG IS + OR +0, DISPLAY MODREG
                TC      +2              # IF MODREG IS -NZ, DO NOTHING
                TC      2BLANK          # IF MODREG IS -0, BLANK MM
                XCH     DSPMMTEM        # RESTORE DSPCOUNT
                TS      DSPCOUNT
                TC      ENDOFJOB

# RECALTST IS ENTERED DIRECTLY AFTER DATA IS LOADED (OR RESEQUENCE VERB IS
# EXECUTED), TERMINATE VERB IS EXECUTED, OR PROCEED WITHOUT DATA VERB IS
# EXECUTED. IT WAKES UP JOB THAT DID TC ENDIDLE.
#
# IF CADRSTOR NOT= +0, IT PUTS +0 INTO DSPLOCK, AND TURNS OFF KEY RLSE
# LIGHT IF DSPLIST IS EMPTY (LEAVES KEY RLSE LIGHT ALONE IF NOT EMPTY).

RECALTST        CCS     CADRSTOR
                TC      RECAL1
                TC      ENDOFJOB        # NORMAL EXIT IF KEYBOARD INITIATED
RECAL1          CAF     ZERO
                XCH     CADRSTOR
                INHINT
                TC      JOBWAKE
                CCS     LOADSTAT
                TC      DOPROC          # + PROCEED WITHOUT DATA
                TC      ENDOFJOB        # PATHALOGICAL CASE EXIT
                TC      DOTERM          # - TERMINATE
                CAF     TWO             # -0 DATA IN OR RESEQUENCE
RECAL2          INDEX   LOCCTR
                AD      LOC             # LOC IS + FOR BASIC JOBS
                INDEX   LOCCTR
                TS      LOC
                CA      NOUNREG         # SAVE VERB IN MPAC, NOUN IN MPAC+1 AT
                TS      L               # TIME OF RESPONSE TO ENDIDLE FOR
                CA      VERBREG         # POSSIBLE LATER TESTING BY JOB THAT HAS
                INDEX   LOCCTR          # BEEN WAKED UP.
                DXCH    MPAC
                RELINT
RECAL3          TC      RELDSP
                TC      ENDOFJOB
```

```
DOTERM          CAF     ZERO
                TC      RECAL2

DOPROC          CAF     ONE
                TC      RECAL2
```

```
# MISCELLANEOUS SERVICE ROUTINES IN FIXED/FIXED

                SETLOC  ENDBLFF

                COUNT*  $$/PIN

# SETNCADR      E CADR ARRIVES IN A. IT IS STORED IN NOUNCADR. EBANK BITS
#               ARE SET.  E ADRES IS DERIVED AND PUT INTO NOUNADD.

SETNCADR        TS      NOUNCADR        # STORE ECADR
                TS      EBANK           # SET EBANK BITS
                MASK    LOW8
                AD      OCT1400
                TS      NOUNADD         # PUT E ADRES INTO NOUNADD
                TC      Q

# SETNADD       GETS E CADR FROM NOUNCADR, SETS EBANK BITS, DERIVES
#               E ADRES AND PUTS IT INTO NOUNADD.

SETNADD         CA      NOUNCADR
                TCF     SETNCADR +1

# SETEBANK      E CADR ARRIVES IN A. EBANK BITS ARE SET. E ADRES IS
#               DERIVED AND LEFT IN A.

SETEBANK        TS      EBANK           # SET EBANK BITS
                MASK    LOW8
                AD      OCT1400         # E ADRES LEFT IN A
                TC      Q

R1D1            OCT     16              # THESE 3 CONSTANTS FORM A PACKED TABLE.
R2D1            OCT     11              # DONT SEPARATE.
R3D1            OCT     4

RIGHT5          TS      CYR
                CS      CYR
                CS      CYR
                CS      CYR
                CS      CYR
                XCH     CYR
                TC      Q

LEFT5           TS      CYL
                CS      CYL
                CS      CYL
                CS      CYL
                CS      CYL
```

```
                XCH     CYL
                TC      Q

SLEFT5          DOUBLE
                DOUBLE
                DOUBLE
                DOUBLE
                DOUBLE
                TC      Q

LOW5            OCT     37              # THESE 3 CONSTANTS FORM A PACKED TABLE.
MID5            OCT     1740            # DONT SEPARATE.
HI5             OCT     76000           # MUST STAY HERE

TCNOVAC         TC      NOVAC
TCWAIT          TC      WAITLIST
TCTSKOVR        TC      TASKOVER
TCFINDVC        TC      FINDVAC

CHRPRIO         OCT     30000           # EXEC PRIORITY OF CHARIN

LOW11           OCT     3777
B12-1           EQUALS  LOW11
LOW8            OCT     377

VD1             OCT     23              # THESE 3 CONSTANTS FORM A PACKED TABLE.
ND1             OCT     21              # DONT SEPARATE.
MD1             OCT     25

BINCON          DEC     10

FALTON          CA      BIT7            # TURN ON OPERATOR ERROR LIGHT.
                EXTEND
                WOR     DSALMOUT        # BIT 7 OF CHANNEL 11
                TC      Q

FALTOF          CS      BIT7            # TURN OFF OPERATOR ERROR LIGHT
                EXTEND
                WAND    DSALMOUT        # BIT 7 OF CHANNEL 11
                TC      Q

RELDSPON        CAF     BIT5            # TURN ON KEY RELEASE LIGHT
                EXTEND
                WOR     DSALMOUT        # BIT 5 OF CHANNEL 11
                TC      Q
```

```
LODSAMPT        EXTEND
                DCA     TIME2
                DXCH    SAMPTIME
                TC      Q

TPSL1           EXTEND                  # SHIFTS MPAC, +1, +2 LEFT 1
                DCA     MPAC +1         # LEAVES OVFIND SET TO +/- 1 FOR OF/UF
                DAS     MPAC +1
                AD      MPAC
                ADS     MPAC
                TS      7               # TS A DOES NOT CHANGE A ON OF/UF.
                TC      Q               # NO NET OF/UF
                TS      MPAC+6          # MPAC +6 SET TO +/- 1 FOR OF/UF
                TC      Q

# IF MPAC, +1 ARE EACH +NZ OR +0 AND C(A)=-0, SHORTMP WRONGLY GIVES +0.
# IF MPAC, +1 ARE EACH -NZ OR -0 AND C(A)=+0, SHORTMP WRONGLY GIVES +0.
# PRSHRTMP FIXES FORST CASE ONLY, BY MERELY TESTING C(A) AND IF IT = -0,
# SETTING RESULT TO -0.
#  (DO NOT USE PRSHRTMP UNLESS MPAC, +1 ARE EACH +NZ OR +0, AS THEY ARE
#  WHEN THEY CONTAIN THE SF CONSTANTS.)

PRSHRTMP        TS      MPTEMP
                CCS     A
                CA      MPTEMP          # C(A) +, DO REGULAR SHORTMP
                TCF     SHORTMP +1      # C(A) +0, DO REGULAR SHORTMP
                TCF     -2              # C(A) -, DO REGULAR SHORTMP
                CS      ZERO            # C(A) -0, FORCE RESULT TO -0 AND RETURN.
                TS      MPAC
                TS      MPAC +1
                TS      MPAC +2
                TC      Q

FLASHON         CAF     BIT6            # TURN ON V/N FLASH
                EXTEND                  # BIT 6 OF CHANNEL 11
                WOR     DSALMOUT
                TC      Q

FLASHOFF        CS      BIT6            # TURN OFF V/N FLASH
                EXTEND
                WAND    DSALMOUT
                TC      Q
```

```
# INTERNAL USE OF KEYBOARD AND DISPLAY PROGRAM.
#
# USER MUST SCHEDULE CALLS TO NVSUB SO THAT THERE IS NO CONFLICT OF USE OR
# CONFUSION TO OPERATOR. THE OLD GRABLOCK (INTERNAL/INTERNAL INTERLOCK)
# HAS BEEN REMOVED AND THE INTERNAL USER NO LONGER HAS THE PROTECTION THIS
# OFFERED.
#
# THERE ARE TWO WAYS A JOB CAN BE PUT TO SLEEP BY THE KEYBOARD + DISPLAY
# PROGRAM.      1) BY ENDIDLE
#               2) BY NVSUBUSY
# THE BASIC CONVENTION IS THAT ONLY ONE JOB WILL BE PERMITTED ASLEEP VIA
# THE KEYBOARD + DISPLAY PROGRAM AT A TIME. IF A JOB ATTEMPTS TO GO TO
# SLEEP BY MEANS OF (1) OR (2) AND THERE IS ALREADY A JOB ASLEEP THAT WAS
# PUT TO SLEEP BY (1) OR (2), THEN AN ABORT IS CAUSED.
#
# THE CALLING SEQUENCE FOR NVSUB IS
#                       CAF             V/N
#       L               TC              NVSUB
#       L+1             RETURN HERE IF OPERATOR HAS INTERVENED
#       L+2             RETURN HERE AFTER EXECUTION
#
# A ROUTINE CALLED NVSUBUSY IS PROVIDED (USE IS OPTIONAL) TO PUT
# YOUR JOB TO SLEEP UNTIL THE OPERATOR RELEASES THE KEYBOARD + DISPLAY
# SYSTEM. NVSUBUSY ALSO TURNS ON THE KEY RELEASE LIGHT.
# NVSUBUSY CANNOT BE CALLED FROM ERASABLE OR F/F MEMORY,
# SINCE JOBSLEEP AND JOBWAKE CAN HANDLE ONLY FIXED BANKS.
#
# THE CALLING SEQUENCE IS
#       CAF     WAKEFCADR
#       TC      NVSUBUSY
#
#
# .
#
# NVSUBUSY IS INTENDED FOR USE WHEN AN INTERNAL PROGRAM FINDS THE OPERATOR
# IS NOT USING THE KEYBOARD + DISPLAY PROGRAM (BY HIS OWN INITIATION). IT IS
# NOT INTENDED FOR USE WHEN ONE INTERNAL PROGRAM FINDS ANOTHER INTERNAL
# PROGRAM USING THE KEYBOARD + DISPLAY PROGRAM.
#
# NVSUBUSY ABORTS (WITH CODE 01206) IF A SECOND JOB ATTEMPTS TO GO TO
# SLEEP IN PINBALL. IN PARTICULAR, IF AN ATTEMPT IS MADE TO GO TO NVSUBUSY
# WHEN
# 1)    DSPLIST NOT= +0.  THIS IS THE CASE WHERE THE CAPACITY OF THE DSPLIST
#       IS EXCEEDED.
# 2)    CADRSTOR NOT= +0.  THIS INDICATES THAT A JOB IS ALREADY USING
```

```
# ENDIDLE.  (+-NZ INDICATE A JOB IS ALREADY ASLEEP DUE TO ENDIDLE.)

PRENVBSY        CS      2K+3            # SPECIAL ENTRANCE FOR ROUTINES IN FIXED
                AD      Q               # BANKS ONLY DESIRING THE FCADR OF (LOC
                AD      FBANK           # FROM WHICH THE TC PRENVBSY WAS DONE) -2
NVSUBUSY        TC      POSTJUMP        # TO BE ENTERED.
                CADR    NVSUBSY1
2K+3            OCT     2003

# NVSUBSY1 MUST BE IN BANK 27 OR LOWER, SO IT WILL PUT CALLER TO SLEEP
# WITH HIS PROPER SUPERBITS.

                SETLOC  ENDSPMM +1
                COUNT*  $$/PIN
NVSUBSY1        TS      L
                TC      ISCADR+0        # ABORT IF CADRSTOR NOT= +0.
                TC      ISLIST+0        # ABORT IF DSPLIST NOT= +0.
                TC      RELDSPON
                CA      L
                TS      DSPLIST
ENDNVBSY        TC      JOBSLEEP

# NVSBWAIT IS A SPECIAL ENTRANCE FOR ROUTINES IN FIXED BANKS ONLY. IF
# SYSTEM IS NOT BUSY, IT EXECUTES V/N AND RETURNS TO L+1 (L= LOC FROM
# WHICH THE TC NVSBWAIT WAS DONE). IF SYSTEM IS BUSY, IT PUTS CALLING JOB
# TO SLEEP WITH L-1 GOING INTO LIST FOR EVENTUAL WAKING UP WHEN SYSTEM
# IS NOT BUSY.

                SETLOC  NVSUBUSY +3
                COUNT*  $$/PIN
NVSBWAIT        LXCH    7               # ZERO NVMONOPT OPTIONS
                TS      NVTEMP
                CAF     BIT14
                MASK    MONSAVE1        # EXTERNAL MONITOR BIT
                AD      DSPLOCK
                CCS     A
                TCF     NVSBWT1         # BUSY
                TCF     NVSBCOM         # FREE. NVSUB WILL SAVE L+1 FOR RETURN
                                        # AFTER EXECUTION.
NVSBWT1         INCR    Q               # L+2. PRENVBSY WILL PUT L-1 INTO LIST AND
                TCF     PRENVBSY        # GO TO SLEEP.

# RELDSP IS USED BY VBPROC, VBTERM, VBRQEXEC, VBRQWAIT, VBRELDSP, EXTENDED
# VERB DISPATCHER, VBRESEQ, RECALTST.
# RELDSP1 IS USED BY MONITOR SET UP, VBRELDSP.

RELDSP          XCH     Q               # SET DSPLOCK TO +0, TURN RELDSP LIGHT
                TS      RELRET          # OFF, SEARCH DSPLIST
                CS      BIT14
```

```
                  INHINT
                  MASK     MONSAVE1
                  TS       MONSAVE1        # TURN OFF EXTERNAL MONITOR BIT
                  CCS      DSPLIST
                  TC       +2
                  TC       RELDSP2         # LIST EMPTY
                  CAF      ZERO
                  XCH      DSPLIST
                  TC       JOBWAKE
RELDSP2           RELINT
                  CS       BIT5            # TURN OFF KEY RELEASE LIGHT
                  EXTEND                   # (BIT 5 OF CHANNEL 11)
                  WAND     DSALMOUT
                  CAF      ZERO
                  TS       DSPLOCK
                  TC       RELRET
RELDSP1           XCH      Q               # SET DSPLOCK TO +0. NO DSPLIST SEARCH.
                  TS       RELRET          # TURN KEY RLSE LIGHT OFF IF DSPLIST IS
                                           # EMPTY. LEAVE KEY RLSE LIGHT ALONE IF
                                           # DSPLIST IS NOT EMPTY.
                  CCS      DSPLIST
                  TC       +2              # +  NOT EMPTY. LEAVE KEY RLSE LIGHT ALONE.
                  TC       RELDSP2         # +0 EMPTY. TURN OFF KEY RLSE LIGHT
                  CAF      ZERO            # -  NOT EMPTY. LEAVE KEY RLSE LIGHT ALONE
                  TS       DSPLOCK
                  TC       RELRET

ENDPINBF          EQUALS
```

```
# PINTEST IS NEEDED FOR AUTO CHECK OF PINBALL.

PINTEST         EQUALS   LST2FAN
```

```
# VBTSTLTS TURNS ON ALL DISPLAY PANEL LIGHTS. AFTER 5 SEC, IT TURNS
# OFF THE CAUTION AND STATUS LIGHTS.

                SETLOC  ENDNVSB1 +1

                COUNT*  $$/PIN
VBTSTLTS        INHINT
                CS      BIT1            # SET BIT 1 OF IMODES33 SO IMUMON WONT
                MASK    IMODES33        # TURN OUT ANY LAMPS.
                AD      BIT1
                TS      IMODES33

                CAF     TSTCON1         # TURN ON UPLINK ACTIVITY, TEMP, KEY RLSE,
                EXTEND                  # V/N FLASH, OPERATOR ERROR.
                WOR     DSALMOUT
                CAF     TSTCON2         # TURN ON NO ATT, GIMBAL LOCK, TRACKER,
                TS      DSPTAB +11D     # PROG ALM.
                CAF     BIT10           # TURN ON TEST ALARM OUTBIT
                EXTEND
                WOR     CHAN13
                CAF     TEN
TSTLTS1         TS      ERCNT
                CS      FULLDSP
                INDEX   ERCNT
                TS      DSPTAB
                CCS     ERCNT
                TC      TSTLTS1
                CS      FULLDSP1
                TS      DSPTAB +1       # TURN ON 3 PLUS SIGNS
                TS      DSPTAB +4
                TS      DSPTAB +6
                CAF     ELEVEN
                TS      NOUT
                RELINT
                CAF     SHOLTS
                INHINT
                TC      WAITLIST
                EBANK=  DSPTAB
                2CADR   TSTLTS2

                TC      ENDOFJOB        # DSPLOCK IS LEFT BUSY (FROM KEYBOARD
                                        # ACTION) UNTIL TSTLTS3 TO INSURE THAT
                                        # LIGHTS TEST WILL BE SEEN.

FULLDSP         OCT     05675           # DISPLAY ALL 8'S
FULLDSP1        OCT     07675           # DISPLAY ALL 8'S AND +
TSTCON1         OCT     00175
                                        # UPLINK ACTIVITY, TEMP. KEY RLSE,
                                        # V/N FLASH, OPERATOR ERROR.
```

```
TSTCON2         OCT     40674           # DSPTAB+11D BITS 3,4,5,6,8,9 LR LITES,
                                        # NO ATT, GIMBAL LOCK, TRACKER, PROG ALM.
TSTCON3         OCT     00115           # CHAN 11 BITS 1, 3, 4, 7.
                                        # UPLINK ACITIVY, TEMP, OPERATOR ERROR.
SHOLTS          OCT     764             # 5 SEC

TSTLTS2         CAF     CHRPRIO         # CALLED BY WAITLIST
                TC      NOVAC
                EBANK=  DSPTAB
                2CADR   TSTLTS3

                TC      TASKOVER

TSTLTS3         CS      TSTCON3         # CALLED BY EXECUTIVE
                INHINT
                EXTEND                  # TURN OFF  UPLINK ACTIVITY, TEMP,
                WAND    DSALMOUT        # OPERATOR ERROR.
                CS      BIT10           # TURN OFF  TEST ALARM OUTBIT
                EXTEND
                WAND    CHAN13
                CAF     BIT4            # MAKE NO ATT FOLLOW BIT 4 OF CHANNEL 12
                EXTEND                  #   (NO ATT LIGHT ON IF IN COARSE ALIGN)
                RAND    CHAN12
                AD      BIT15           # TURN OFF AUTO, HOLD, FREE, SPARE,
                TS      DSPTAB +11D     # GIMBAL LOCK, SPARE, TRACKER, PROG ALM
                CS      13-11,1         # SET BITS TO INDICATE ALL LAMPS OUT. TEST
                MASK    IMODES33        # LIGHTS COMPLETE.
                AD      PRIO16
                TS      IMODES33

                CS      OCT55000
                MASK    IMODES30
                AD      PRIO15          # 15000.
                TS      IMODES30

                CS      RFAILS2
                MASK    RADMODES
                AD      RCDUFBIT
                TS      RADMODES

                RELINT

                TC      BANKCALL        # REDISPLAY C(MODREG)
                CADR    DSPMM
                TC      KILMONON        # TURN ON KILL MONITOR BIT.
                TC      FLASHOFF        # TURN OFF V/N FLASH.
                TC      POSTJUMP        # DOES RELDSP AND GOES TO PINBRNCH IF
                CADR    TSTLTS4         #  ENDIDLE IS AWAITING OPERATOR RESPONSE.
```

```
13-11,1         OCT     16001
RFAILS2         OCT     330             # RADAR CDU AND DATA FAIL FLAGS.
OCT55000        OCT     55000
ENDPINS2        EQUALS
```

```
# ERROR LIGHT RESET (RSET) TURNS OFF:
# UPLINK ACTIVITY, AUTO, HOLD, FREE, OPERATOR ERROR,
# PROG ALM, TRACKER FAIL.
# LEAVES GIMBAL LOCK AND NO ATT ALONE.
# IT ALSO ZEROS THE 'TEST ALARM' OUT BIT, WHICH TURNS OFF STBY, RESTART.
# IT ALSO SETS 'CAUTION RESET' TO 1.
# IT ALSO FORCES BIT 12 OF ALL DSPTAB ENTRIES TO 1.

                SETLOC  DOPROC +2
                COUNT*  $$/PIN
ERROR           XCH     21/22REG        # RESTORE ORIGINAL C(DSPLOCK). THUS ERROR
                TS      DSPLOCK         # LIGHT RESET LEAVES DSPLOCK UNCHANGED.
                INHINT
                CAF     BIT10           # TURN ON 'CAUTION RESET' OUTBIT
                EXTEND
                WOR     DSALMOUT        # BIT10 CHAN 11
                CAF     GL+NOATT        # LEAVE GIMBAL LOCK AND NO ATT INTACT,
                MASK    DSPTAB +11D     # TURNING OFF AUTO, HOLD, FREE,
                AD      BIT15           # PROG ALARM, AND TRACKER.
                TS      DSPTAB +11D
                CS      PRIO16          # RESET FAIL BITS WHICH GENERATE PROG
                MASK    IMODES33        # ALARM SO THAT IF THE FAILURE STILL
                AD      PRIO16          # EXISTS, THE ALARM WILL COME BACK.
                TS      IMODES33
                CS      BIT10
                MASK    IMODES30
                AD      BIT10
                TS      IMODES30

                CS      RFAILS
                MASK    RADMODES
                AD      RCDUFBIT
                TS      RADMODES

                CS      BIT10           # TURN OFF 'TEST ALARM' OUTBIT.
                EXTEND
                WAND    CHAN13
                CS      ERCON           # TURN OFF UPLINK ACTIVITY,
                EXTEND                  # OPERATOR ERROR.
                WAND    DSALMOUT
TSTAB           CAF     BINCON          # (DEC 10)
                TS      ERCNT           # ERCNT = COUNT
                INHINT
                INDEX   ERCNT
                CCS     DSPTAB
                AD      ONE
                TC      ERPLUS
                AD      ONE
ERMINUS         CS      A
                MASK    NOTBIT12
```

```
                TC      ERCOM
ERPLUS          CS      A
                MASK    NOTBIT12
                CS      A               # MIGHT WANT TO RESET CLPASS, DECBRNCH,
ERCOM           INDEX   ERCNT           # ETC.
                TS      DSPTAB
                RELINT
                CCS     ERCNT
                TC      TSTAB   +1
                CAF     ZERO
                TS      FAILREG
                TS      FAILREG +1
                TS      FAILREG +2
                TS      SFAIL
                TC      ENDOFJOB

ERCON           OCT     104             # CHAN 11 BITS 3,7.
                                        # UPLINK ACTIVITY, AND OPERATOR ERROR.
RFAILS          OCT     330             # RADAR CDU AND DATA FAIL FLAGS.
GL+NOATT        OCT     00050           # NO ATT AND GIMBAL LOCK LAMPS
NOTBIT12        OCT     73777

ENDPINS1        EQUALS

                SBANK=  LOWSUPER
```

```
# MOD NO: 0                        DATE: 1 MAY 1968
# MOD BY: DIGITAL DEVEL GROUP    LOG SECTION R60,R62
#
# FUNCTIONAL DESCRIPTION:
#
# CALLED AS A GENERAL SUBROUTINE TO MANEUVER THE LM TO A SPECIFIED
# ATTITUDE.
#
# 1. IF THE 3-AXIS FLAG IS NOT SET THE FINAL CDU ANGLES ARE
# CALCULATED (VECPOINT).
#
# 2. THE FDAI BALL ANGLES (NOUN 18) ARE CALCULATED (BALLANGS).
#
# 3. REQUEST FLASHING DISPLAY V50 N18 PLEASE PERFORM AUTO MANEUVER.
#
# 4. IF PRIORITY DISPLAY FLAG IS SET DO A PHASECHANGE. THEN AWAIT
# ASTRONAUT RESPONSE.
#
# 5. DISPLAY RESPONSE RETURNS:
#
#     A. ENTER - RESET 3-AXIS FLAG AND RETURN TO CLIENT.
#
#     B. TERMINATE - IF IN P00 GO TO STEP 5A. OTHERWISE CHECK IF R61 IS
#        THE CALLING PROGRAM. IF IN R61 AN EXIT IS MADE TO GOTOV56. IF
#        NOT IN R61 AN EXIT IS DONE VIA GOTOPOOH.
#
#     C. PROCEED - CONTINUE WITH PROGRAM AT STEP 6.
#
# 6. IF THE 3-AXISFLAG IS NOT SET, THE FINAL CDU ANGLES ARE CALCULATED
#    (VECPOINT).
#
# 7. THE FDAI BALL ANGLES (NOUN 18) ARE CALCULATED (BALLANGS).
#
# 8. IF THE G+N SWITCH IS NOT SET GO BACK TO STEP 3.
#
# 9. IF THE AUTO SWITCH IS NOT SET GO BACK TO STEP 3.
#
# 10. NONFLASHING DISPLAY V06N18 (FDAI ANGLES).
#
# 11. DO A PHASECHANGE.
#
# 12. DO A MANEUVER CALCULATION AND ICDU DRIVE ROUTINE TO ACHIEVE FINAL
#
#     GIMBAL ANGLES (GOMANUR).
# 13. AT END OF MANEUVER GO TO STEP 3.
#
#          IF SATISFACTORY MANEUVER STEP 5A EXITS R60.
#          FOR FURTHER ADJUSTMENT OF THE VEHICLE ATTITUDE ABOUT THE
#          DESIRED VECTOR, THE ROUTINE MAY BE PERFORMED AGAIN STARTING AT
```

```
#          STEP 5C.
#
# CALLING SEQUENCE:  TC BANKCALL
#                    CADR R60LEM
#
# ERABLE INITIALIZATION REQUIRED : SCAXIS, POINTVSM (FOR VECPOINT)
#                                        3AXISFLG.
#
# SUBROUTINES CALLED: VECPOINT, BALLANGS, GOPERF2R, LINUS, GODSPER,
#                    GOMANUR, DOWNFLAG, PHASCHNG, UPFLAG
#
# NORMAL EXIT MODES: CAE TEMPR60    (CALLERS RETURN ADDRESS)
#                    TC  BANKJUMP
#
# ALARMS: NONE
#
# OUTPUT: NONE
#
# DEBRIS: CPHI, CTHETA, CPSI, 3AXISFLG, TBASE2

                BANK    34
                SETLOC  MANUVER
                BANK

                EBANK=  TEMPR60

                COUNT*  $$/R06
R60LEM          TC      MAKECADR
                TS      TEMPR60

REDOMANN        CAF     3AXISBIT
                MASK    FLAGWRD5        # IS 3-AXIS FLAG SET
                CCS     A
                TCF     TOBALL          # YES
                TC      INTPRET
                CALL
                        VECPOINT        # TO COMPUTE FINAL ANGLES
                STORE   CPHI            # STORE FINAL ANGLES - CPHI,CTHETA,CPSI
                EXIT

TOBALL          TC      BANKCALL
                CADR    BALLANGS        # TO CONVERT ANGLES TO FDAI
TOBALLA         CAF     V06N18
                TC      BANKCALL
                CADR    GOPERF2R        # DISPLAY PLEASE PERFORM AUTO MANEUVER
                TC      R61TEST
                TC      REDOMANC        # PROCEED
                TC      ENDMANU1        # ENTER I.E. FINISHED WITH R60
```

```
                TC      CHKLINUS        # TO CHECK FOR PRIORITY DISPLAYS
                TC      ENDOFJOB

REDOMANC        CAF     3AXISBIT
                MASK    FLAGWRD5        # IS 3-AXIS FLAG SET
                CCS     A
                TCF     TOBALLC         # YES
                TC      INTPRET
                CALL
                        VECPOINT        # TO COMPUTE FINAL ANGLES
                STORE   CPHI            # STORE ANGLES
                EXIT

TOBALLC         TC      BANKCALL
                CADR    BALLANGS        # TO CONVERT ANGLES TO FDAI
                TC      G+N,AUTO        # CHECK AUTO MODE
                CCS     A
                TCF     TOBALLA         # NOT AUTO, GO REREQUEST AUTO MANEUVER.

AUTOMANV        CAF     V06N18          # STATIC DISPLAY DURING AUTO MANEUVER
                TC      BANKCALL
                CADR    GODSPR
                TC      CHKLINUS        # TO CHECK FOR PRIORITY DISPLAYS

STARTMNV        TC      BANKCALL        # PERFORM MANEUVER VIA KALCMANU
                CADR    GOMANUR

ENDMANUV        TCF     TOBALLA         # FINISHED MANEUVER.
ENDMANU1        TC      DOWNFLAG        # RESET 3-AXIS FLAG
                ADRES   3AXISFLG
                CAE     TEMPR60
                TC      BANKJUMP

CHKLINUS        CS      FLAGWRD4
                MASK    PDSPFBIT        # IS PRIORITY DISPLAY FLAG SET?
                CCS     A
                TC      Q               # NO - EXIT
                CA      Q
                TS      MPAC +2         # SAVE RETURN
                CS      THREE           # OBTAIN LOCATION FOR RESTART
                AD      BUF2            # HOLDS Q OF LAST DISPLAY
                TS      TBASE2

                TC      PHASCHNG
                OCT     00132

                CAF     BIT7
                TC      LINUS           # GO SET BITS FOR PRIORITY DISPLAY
                TC      MPAC +2
```

```
RELINUS         CAF     PRIO26          # RESTORE ORIGINAL PRIORITY
                TC      PRIOCHNG

                CAF     TRACKBIT        # DON'T CONTINUE R60 UNLESS TRACKFLAG ON.
                MASK    FLAGWRD1
                CCS     A
                TCF     RER60

                CAF     RNDVZBIT        # IS IT P20?
                MASK    FLAGWRD0
                CCS     A
                TC      +4              # YES
                TC      PHASCHNG        # NO, MUST BE P25, SET 2.11 SPOT
                OCT     40112

                TC      ENDOFJOB

                TC      PHASCHNG        # SET 2.7 SPOT FOR P20
                OCT     40072

                TC      ENDOFJOB

RER60           TC      UPFLAG          # SET PRIO DISPLAY FLAG AFTER RESTART
                ADRES   PDSPFLAG

                TC      TBASE2

R61TEST         CA      MODREG          # IF WE ARE IN P00 IT MUST BE V49 OR V89
                EXTEND
                BZF     ENDMANU1        # THUS WE GO TO ENDEXT VIA USER

                CA      FLAGWRD4        # ARE WE IN R61  (P20 OR P25)
                MASK    PDSPFBIT
                EXTEND
                BZF     GOTOPOOH        # NO
                TC      GOTOV56         # YES

BIT14+7         OCT     20100
OCT203          OCT     203
V06N18          VN      0618

# SUBROUTINE TO CHECK FOR G+N CONTROL. AUTO STABILIZATION
#
# RETURNS WITH C(A) = +  IF NOT SET FOR G+N, AUTO
# RETURNS WITH C(A) = +0 IF SWITCHES ARE SET

G+N,AUTO        EXTEND
                READ    CHAN30
                MASK    BIT10
                CCS     A
                TC      Q               # NOT IN G+N  C(A) = +
```

```
ISITAUTO        EXTEND                          # CHECK FOR AUTO MODE
                READ    CHAN31
                MASK    BIT14
                TC      Q                       # (+) = NOT IN AUTO,  (+0) = AOK
```

```
# PROGRAM DESCRIPTION BALLANGS
# MOD NO.         LOG SECTION  R60,R62
#
# WRITTEN BY RAMA M.AIYAWAR
# FUNCTIONAL DESCRIPTION
#
# COMPUTES LM FDAI BALL DISPLAY ANGLES
# CALLING SEQUENCE
#
#               TC      BALLANGS
# NORMAL EXIT MODE
#
#               TC      BALLEXIT        # (SAVED Q)
#
# ALARM OR EXIT MODE  NIL
# SUBROUTINES CALLED
#               CD*TR*G
#               ARCTAN
#
# INPUT
#
# CPHI,CTHETA,CPSI  ARE  THE ANGLES CORRESPONDING TO AOG,AIG,AMG. THEY ARE
# SP,2S COMPLIMENT SCALED TO HALF REVOLUTION.
# OUTPUT
#
# FDAIX,FDAIY,FDAIZ ARE THE REQUIRED BALL ANGLES SCALED TO HALF REVOLUTION
# SP,2S COMPLIMENT.
# THESE ANGLES WILL BE DISPLAYED AS DEGREES AND HUNDREDTHS. IN THE ORDER  ROLL, PITCH, YAW, USING NOUNS 18 & 19.
#
# ERASABLE INITIALIZATION REQUIRED
#
# CPHI,CTHETA,CPSI EACH A SP REGISTER
# DEBRIS
#
# A,L,Q,MPAC,SINCDU,COSCDU,PUSHLIS,BALLEXIT
#
#
# NOMENCLATURE:  CPHI, CTHETA, & CPSI REPRESENT THE OUTER, INNER, & MIDDLE GIMBAL ANGLES, RESPECTIVELY; OR
# EQUIVALENTLY, CDUX, CDUY, & CDUZ.
#
# NOTE:  ARCTAN CHECKS FOR OVERFLOW AND SHOULD BE ABLE TO HANDLE ANY SINGULARITIES.

                SETLOC  BAWLANGS
                BANK

                COUNT*  $$/BALL
BALLANGS        TC      MAKECADR
                TS      BALLEXIT
                CA      CPHI
```

```
                TS      CDUSPOT +4
                CA      CTHETA
                TS      CDUSPOT
                CA      CPSI
                TS      CDUSPOT +2

                TC      INTPRET
                SETPD   CALL
                        OD
                        CD*TR*G

                DLOAD   DMP
                        SINCDUX         # SIN (OGA)
                        COSCDUZ         # COS (MGA)

                SL1     DCOMP           # SCALE
                ARCSIN  PDDL            # YAW = ARCSIN(-SXCZ) INTO 0 PD
                        SINCDUZ
                STODL   SINTH           # (SINTH = 18D IN PD)
                        COSCDUZ
                DMP     SL1             # RESCALE
                        COSCDUX
                STCALL  COSTH           # (COSTH= 16D IN PD)
                        ARCTAN
                PDDL    DMP             # ROLL = ARCTAN(SZ/CZCX) INTO 2 PD
                        SINCDUZ
                        SINCDUX
                SL2     PUSH            # SXSZ INTO 4 PD
                DMP     PDDL            # SXSZCY INTO 4 PD
                        COSCDUY
                DMP     PDDL            # SXSZSY INTO 6 PD
                        SINCDUY
                        COSCDUX
                DMP     SL1             # CXCY
                        COSCDUY
                DSU     STADR           # PULL UP FROM 6 PD
                STODL   COSTH           # COSTH = CXCY - SXSZSY
                        SINCDUY
                DMP     SL1
                        COSCDUX         # CXSY
                DAD     STADR           # PULL UP FROM 4 PD
                STCALL  SINTH           # SINTH = CXSY + SXSZCY
                        ARCTAN          # RETURNS WITH D(MPAC) = PITCH
                PDDL    VDEF            # PITCH INTO 2 PD, ROLL INTO MPAC FROM 2PD
                RTB                     # VDEF MAKES V(MPAC) = ROLL, PITCH, YAW
                        V1STO2S
                STORE   FDAIX           # MODE IS TP
                EXIT

ENDBALL         CA      BALLEXIT
```

```
TC BANKJUMP
```

```
# PROGRAM DESCRIPTION - VECPOINT
#
#
#          THIS INTERPRETIVE SUBROUTINE MAY BE USED TO POINT A SPACECRAFT AXIS IN A DESIRED DIRECTION.  THE AXIS
# TO BE POINTED MUST APPEAR AS A HALF UNIT DOUBLE PRECISION VECTOR IN SUCCESSIVE LOCATIONS OF ERASABLE MEMORY
# BEGINNING WITH THE LOCATION CALLED SCAXIS.  THE COMPONENTS OF THIS VECTOR ARE GIVEN IN SPACECRAFT COORDINATES.
# THE DIRECTION IN WHICH THIS AXIS IS TO BE POINTED MUST APPEAR AS A HALF UNIT DOUBLE PRECISION VECTOR IN
# SUCCESSIVE LOCATIONS OF ERASABLE MEMORY BEGINNING WITH THE ADDRESS CALLED POINTVSM.  THE COMPONENTS OF THIS
# VECTOR ARE GIVEN IN STABLE MEMBER COORDINATES.  WITH THIS INFORMATION VECPOINT COMPUTES A SET OF THREE GIMBAL
# ANGLES (2S COMPLEMENT) CORESPONDING TO THE CROSS-PRODUCT ROTATION BETWE EN SCAXIS AND POINTVSM AND STORES THEM
# IN T(MPAC) BEFORE RETURNING TO THE CALLER.
#          THIS ROTATION, HOWEVER, MAY BRING THE S/C INTO GIMBAL LOCK.  WHEN POINTING A VECTOR IN THE Y-Z PLANE,
# THE TRANSPONDER AXIS, OR THE AOT FOR THE LEM, THE PROGRAM WILL CORRECT THIS PROBLEM BY ROTATING THE CROSS-
# PRODUCT ATTITUDE ABOUT POINTVSM BY A FIXED AMOUNT SUFFICIENT TO ROTATE THE DESIRED S/C ATTITUDE OUT OF GIMBAL
# LOCK.  IF THE AXIS TO BE POINTED IS MORE THAN 40.6 DEGREES BUT LESS THAN 60.5 DEG FROM THE +X (OR-X) AXIS,
# THE ADDITIONAL ROTATION TO AVOID GIMAL LOCK IS 35 DEGREES.  IF THE AXIS IS MORE THAN 60.5 DEGEES FROM +X (OR -X)
# THE ADDITIONAL ROTATION IS 35 DEGREES.  THE GIMBAL ANGLES CORRESPONDING TO THIS ATTITUDE ARE THEN COMPUTED AND
# STORED AS 2S COMPLIMENT ANGLES IN T(MPAC) BEFORE RETURNING TO THE CALLER.
#          WHEN POINTING THE X-AXIS, OR THE THRUST VECTOR, OR ANY VECTOR WITHIN 40.6 DEG OF THE X-AXIS, VECPOINT
# CANNOT CORRECT FOR A CROSS-PRODUCT ROTATION INTO GIMBAL LOCK.  IN THIS CASE A PLATFORM REALIGNMENT WOULD BE
# REQUIRED TO POINT THE VECTOR IN THE DESIRED DIRECTION.  AT PRESENT NO INDICATION IS GIVEN FOR THIS SITUATION
# EXCEPT THAT THE FINAL MIDDLE GIMBAL ANGLE IN MPAC +2 IS GREATER THAN 59 DEGREES.
#
#          CALLING SEQUENCE -
#              1) LOAD SCAXIS, POINTVSM
#              2) CALL
#                          VECPOINT
#
#          RETURNS WITH
#
#              1) DESIRED OUTER  GIMBAL ANGLE IN MPAC
#              2) DESIRED INNER  GIMBAL ANGLE IN MPAC +1
#              3) DESIRED MIDDLE GIMBAL ANGLE IN MPAC +2
#
#          ERASABLES USED -
#
#              1) SCAXIS           6
#              2) POINTVSM         6
#              3) MIS             18
#              4) DEL             18
#              5) COF              6
#              6) VECQTEMP         1
#              7) ALL OF VAC AREA 43
#
#                           TOTAL     99

                SETLOC  VECPT
                BANK
```

```
                COUNT*  $$/VECPT

                EBANK=  BCDU

VECPNT1         STQ     BOV             # THIS ENTRY USES DESIRED CDUS
                        VECQTEMP        # NOT PRESENT-ENTER WITH CDUD'S IN MPAC
                        VECPNT2
VECPNT2         AXC,2   GOTO
                        MIS
                        STORANG
VECPOINT        STQ     BOV             # SAVE RETURN ADDRESS
                        VECQTEMP
                        VECLEAR         # AND CLEAR OVFIND
VECLEAR         AXC,2   RTB
                        MIS             # READ THE PRESENT CDU ANGLES AND
                        READCDUK        # STORE THEM IN PD25, 26, 27
STORANG         STCALL  25D
                        CDUTODCM        # S/C AXES TO STABLE MEMBER AXES (MIS)
                VLOAD   VXM
                        POINTVSM        # RESOLVE THE POINTING DIRECTION VF INTO
                        MIS             # INITIAL S/C AXES ( VF = POINTVSM)
                UNIT
                STORE   28D
                                        # PD 28 29 30 31 32 33
                VXV     UNIT            # TAKE THE CROSS PRODUCT VF X VI
                        SCAXIS          # WHERE VI = SCAXIS
                BOV     VCOMP
                        PICKAXIS
                STODL   COF             # CHECK MAGNITUDE
                        36D             # OF CROSS PRODUCT
                DSU     BMN             # VECTOR, IF LESS
                        DPB-14          # THAN B-14 ASSUME
                        PICKAXIS        # UNIT OPERATION
                VLOAD   DOT             #           INVALID.
                        SCAXIS
                        28D
                SL1     ARCCOS
COMPMATX        CALL                    # NOW COMPUTE THE TRANSFORMATION FROM
                        DELCOMP         # FINAL S/C AXES TO INITIAL S/C AXES  MFI
                AXC,1   AXC,2
                        MIS             # COMPUTE THE TRANSFORMATION FROM FINAL
                        KEL             # S/C AXES TO STABLE MEMBER AXES
                CALL                    # MFS = MIS MFI
                        MXM3            # (IN PD LIST)

                DLOAD   ABS
                        6               # MFS6 = SIN(CPSI)                      $2
                DSU     BMN
                        SINGIMLC        # = SIN(59 DEGS)                        $2
                        FINDGIMB        # /CPSI/ LESS THAN 59 DEGS
```

```
                                         # I.E. DESIRED ATTITUDE NOT IN GIMBAL LOCK

                DLOAD   ABS              # CHECK TO SEE IF WE ARE POINTING
                        SCAXIS           # THE THRUST AXIS
                DSU     BPL
                        SINVEC1          # SIN 49.4 DEGS                         $2
                        FINDGIMB         # IF SO, WE ARE TRYING TO POINT IT INTO
                VLOAD                    # GIMBAL LOCK, ABORT COULD GO HERE
                STADR
                STOVL   MIS +12D
                STADR                    # STORE MFS (IN PD LIST) IN MIS
                STOVL   MIS +6
                STADR
                STOVL   MIS
                        MIS +6           # INNER GIMBAL AXIS IN FINAL S/C AXES
                BPL     VCOMP            # LOCATE THE IG AXIS DIRECTION CLOSEST TO
                        IGSAMEX          # FINAL X S/C AXIS

IGSAMEX         VXV     BMN              # FIND THE SHORTEST WAY OF ROTATING THE
                        SCAXIS           # S/C OUT OF GIMBAL LOCK BY A ROTATION
                        U=SCAXIS         # ABOUT +- SCAXIS, I.E. IF (IG (SGN MFS3)
                                         # X SCAXIS . XF) LESS THAN 0, U = SCAXIS
                                         # OTHERWISE U = -SCAXIS

                VLOAD   VCOMP
                        SCAXIS
                STCALL  COF              # ROTATE ABOUT -SCAXIS
                        CHEKAXIS
U=SCAXIS        VLOAD
                        SCAXIS
                STORE   COF              # ROTATE ABOUT + SCAXIS
CHEKAXIS        DLOAD   ABS
                        SCAXIS           # SEE IF WE ARE POINTING THE AOT
                DSU     BPL
                        SINVEC2          # SIN 29.5 DEGS                         $2
                        PICKANG1         # IF SO, ROTATE 50 DEGS ABOUT +- SCAXIS
                DLOAD   GOTO             # IF NOT, MUST BE POINTING THE TRANSPONDER
                        VECANG2          # OR SOME VECTOR IN THE Y, OR Z PLANE
                        COMPMFSN         # IN THIS CASE ROTATE 35 DEGS TO GET OUT
                                         # OF GIMBAL LOCK (VECANG2  $360)
PICKANG1        DLOAD
                        VECANG1          # = 50 DEGS                           $ 360
COMPMFSN        CALL
                        DELCOMP          # COMPUTE THE ROTATION ABOUT SCAXIS TO
                AXC,1   AXC,2            # BRING MFS OUT OF GIMBAL LOCK
                        MIS
                        KEL
                CALL                     # COMPUTE THE NEW TRANSFORMATION FROM
                        MXM3             # DESIRED S/C AXES TO STABLE MEMBER AXES
                                         # WHICH WILL ALIGN VI WITH VF AND AVOID
```

```
                                        # GIMBAL LOCK
FINDGIMB        AXC,1   CALL
                        0               # EXTRACT THE COMMANDED CDU ANGLES FROM
                        DCMTOCDU        # THIS MATRIX
                RTB     SETPD
                        V1STO2S         # CONVERT TO 2:S COMPLEMENT
                        0
                GOTO
                        VECQTEMP        # RETURN TO CALLER

PICKAXIS        VLOAD   DOT             # IF VF X VI = 0,  FIND VF . VI
                        28D
                        SCAXIS
                BMN     TLOAD
                        ROT180
                        25D
                GOTO                    # IF VF = VI, CDU DESIRED = PRESENT CDU
                        VECQTEMP        # PRESENT CDU ANGLES

                BANK    35
                SETLOC  MANUVER1
                BANK
ROT180          VLOAD   VXV             # IF VF, VI ANTIPARALLEL, 180 DEG ROTATION
                        MIS     +6      # IS REQUIRED. Y STABLE MEMBER AXIS IN
                        HIDPHALF        # INITIAL S/C AXES.
                UNIT    VXV             # FIND Y(SM) X X(I)
                        SCAXIS          # FIND UNIT(VI X UNIT(Y(SM) X X(I)))
                UNIT    BOV             # I.E. PICK A VECTOR IN THE PLANE OF X(I),
                        PICKX           # Y(SM) PERPENDICULAR TO VI
                STODL   COF
                        36D             # CHECK MAGNITUDE
                DSU     BMN             # OF THIS VECTOR.
                        DPB-14          # IF LESS THAN B-14,
                        PICKX           # PICK X-AXIS.
                VLOAD
                        COF
XROT            STODL   COF
                        HIDPHALF
                GOTO
                        COMPMATX
PICKX           VLOAD   GOTO            # PICK THE XAXIS IN THIS CASE
                        HIDPHALF
                        XROT
SINGIMLC        2DEC    .4285836003     # =SIN(59)              $2

SINVEC1         2DEC    .3796356537     # =SIN(49.4)            $2

SINVEC2         2DEC    .2462117800     # =SIN(29.5)            $2

VECANG1         2DEC    .1388888889     # = 50 DEGREES                      $360
```

```
VECANG2         2DEC    .09722222222    # = 35 DEGREES                  $360

1BITDP          OCT     0               # KEEP THIS BEFORE DPB(-14)    *********
DPB-14          OCT     00001
                OCT     00000
```

```
# ROUTINE FOR INITIATING AUTOMATIC MANEUVER VIA KEYBOARD (V49)

                BANK    34
                SETLOC  R62
                BANK
                EBANK=  BCDU

                COUNT*  $$/R62

R62DISP         EQUALS  R62FLASH

R62FLASH        CAF     V06N22          # FLASH V06N22 AND
                TC      BANKCALL        # ICDU ANGLES
                CADR    GOFLASH
                TCF     ENDEXT          # TERMINATE
                TCF     GOMOVE          # PROCEED
                TCF     R62FLASH        # ENTER

                                        # ASTRONAUT MAY LOAD NEW ICDUS AT THIS
                                        # POINT
GOMOVE          TC      UPFLAG          # SET FOR 3-AXIS MANEUVER
                ADRES   3AXISFLG

                TC      BANKCALL
                CADR    R60LEM
                TCF     ENDEXT          # END R62
```

```
# SUBROUTINE NAME: R05 - S-BAND ANTENNA FOR LM
#
# MOD0 BY T. JAMES
# MOD1 BY P. SHAKIR
#
# FUNCTIONAL DESCRIPTION
#
#     THE S-BAND ANTENNA ROUTINE, R05, COMPUTES AND DISPLAYS THE PITCH AND
# YAW ANTENNA GIMBAL ANGLES REQUIRED TO POINT THE LM STEERABLE ANTENNA
# TOWARD THE CENTER OF THE EARTH. THIS ROUTINE IS SELECTED BY THE ASTRO-
# NAUT VIA DSKY ENTRY DURING COASTING FLIGHT OR WHEN THE LM IS ON THE MOON
# SURFACE. THE EARTH OR MOON REFERENCE COORDINATE SYSTEM IS USED DEPENDING
# ON WHETHER THE LM IS ABOUT TO ENTER OR HAS ALREADY ENTERED THE MOON
# SPHERE OF INFLUENCE, RESPECTIVELY
#
# TO CALL SUBROUTINE, ASTRONAUT KEYS IN V 64 E
#
# SUBROUTINES CALLED-
#       R02BOTH
#       INTPRET
#       LOADTIME
#       LEMCONIC
#       LUNPOS
#       CDUTRIG
#       *SMNB*
#       BANKCALL
#       B50OFF
#       ENDOFJOB
#       BLANKET
#
# RETURNS WITH
#       PITCH ANGLE IN PITCHANG         REV. B0
#       YAW ANGLE IN YAWANG             REV. B0
#
# ERASABLES USED
#       PITCHANG
#       YAWANG
#       RLM
#       VAC AREA

                BANK    41
                SETLOC  SBAND
                BANK

                EBANK=  WHOCARES
                COUNT*  $$/R05
SBANDANT        TC      BANKCALL
```

```
                CADR    RO2BOTH         # CHECK IF IMU IS ON AND ALIGNED
                TC      INTPRET
                SETPD   RTB
                        OD
                        LOADTIME        # PICK UP CURRENT TIME
                STCALL  TDEC1           # ADVANCE INTEGRATION TO TIME IN TDEC1
                        LEMCONIC        # USING CONIC INTEGRATION
                SLOAD   BHIZ
                        X2              # X2 =0 EARTH SPHERE, X2 =2 MOON SPHERE
                        CONV4
                VLOAD
                        RATT
                STODL   RLM
                        TAT
CONV3           CALL
                        LUNPOS          # UNIT POSITION VECTOR FROM EARTH TO MOON
                VLOAD   VXSC
                        VMOON
                        REMDIST         # MEAN DISTANCE FROM EARTH TO MOON
                VSL1    VAD
                        RLM
                GOTO
                        CONV5
CONV4           VLOAD
                        RATT            # UE = -UNIT(RATT)  EARTH SPHERE
CONV5           SETPD   UNIT            # UE = -UNIT((REM)(UEM) + RL)  MOON SPHERE
                        OD              # SET PL POINTER TO 0
                VCOMP   CALL
                        CDUTRIG         # COMPUTE SINES AND COSINES OF CDU ANGLES
                MXV     VSL1            # TRANSFORM REF. COORDINATE SYSTEM TO
                        REFSMMAT        # STABLE MEMBER  B-1 X B-1 X B+1 = B-1
                PUSH    DLOAD           # 8D
                        HI6ZEROS
                STORE   PITCHANG
                STOVL   YAWANG          # ZERO OUT ANGLES
                CALL
                        *SMNB*
                STODL   RLM             # PRE-MULTIPLY RLM BY (NBSA) MATRIX(BO)
                        RLM     +2
                PUSH    DSU
                        RLM
                DMP
                        1OVSQRT2
                STODL   RLM     +2
                DAD     DMP
                        RLM
                        1OVSQRT2
                STOVL   RLM             # R  B-1
                        RLM
                UNIT    PDVL
```

```
                        RLM
                VPROJ   VSL2            # PROJECTION OF R ONTO LM XZ PLANE
                        HIUNITY
                BVSU    BOV             # CLEAR OVERFLOW INDICATOR IF ON
                        RLM
                        COVCNV
COVCNV          UNIT    BOV             # EXIT ON OVERFLOW
                        SBANDEX
                PUSH    VXV             # URP VECTOR  B-1
                        HIUNITZ
                VSL1    VCOMP           # UZ X URP = -(URP X UZ)
                STORE   RLM             # X VEC  B-1
                DOT     PDVL            # SGN(X.UY) UNSCALED
                        HIUNITY
                        RLM
                ABVAL   SIGN
                ASIN                    # ASIN((SGN(X.UY))ABV(X)) REV BO
                STOVL   PITCHANG
                        URP
                DOT     BPL
                        HIUNITZ
                        NOADJUST        # YES, -90 TO +90
                DLOAD   DSU
                        HIDPHALF
                        PITCHANG
                STORE   PITCHANG
NOADJUST        VLOAD   VXV
                        UR              # Z = (UR X URP)
                        URP
                VSL1
                STODL   RLM             # Z VEC  B-1
                        PITCHANG
                SIN     VXSC
                        HIUNITZ
                PDDL    COS
                        PITCHANG
                VXSC    VSU
                        HIUNITX         # (UX COS ALPHA) - (UZ SIN ALPHA)
                DOT     PDVL            # YAW.Z
                        RLM
                        RLM
                ABVAL   SIGN
                ASIN
                STORE   YAWANG
SBANDEX         EXIT
                CA      EXTVBACT
                MASK    BIT5            # IS BIT5 STILL ON
                EXTEND
                BZF     ENDEXT          # NO
                CAF     PRIO5
```

```
                TC      PRIOCHNG
                CAF     V06N51          # DISPLAY ANGLES
                TC      BANKCALL
                CADR    GOMARKFR
                TC      B5OFF           # TERMINATE
                TC      B5OFF           # PROCEED
                TC      ENDOFJOB        # RECYCLE
                CAF     BIT3            # IMMEDIATE RETURN
                TC      BLANKET         # BLANK R3
                CAF     PRIO4
                TC      PRIOCHNG
                TC      SBANDANT +2     # YES, CONTINUE DISPLAYING ANGLES
V06N51          VN      0651
1OVSQRT2        2DEC    .7071067815     # 1/SQRT(2)

UR              EQUALS  0D
URP             EQUALS  6D
                SBANK=  LOWSUPER

# *** END OF LNYAIDE .001 ***
```

```
                BANK    25
                SETLOC  RRLEADIN
                BANK

                EBANK=  RSTACK

# RADAR SAMPLING LOOP.

                COUNT*  $$/RLEAD
RADSAMP         CCS     RSAMPDT         # TIMES NORMAL ONCE-PER-SECOND SAMPLING.
                TCF     +2
                TCF     TASKOVER        # +0 INSERTED MANUALLY TERMINATES TEST.

                TC      WAITLIST
                EBANK=  RSTACK
                2CADR   RADSAMP

                CAF     PRIO25
                TC      NOVAC
                EBANK=  RSTACK
                2CADR   DORSAMP

                CAF     BIT14           # FOR CYCLIC SAMPLING, RTSTDEX =
                EXTEND                  # RTSTLOC/2 + RTSTBASE
                MP      RTSTLOC
                AD      RTSTBASE        # 0 FOR RR, 2 FOR LR.
                TS      RTSTDEX
                TCF     TASKOVER

# DO THE ACTUAL RADAR SAMPLE.

DORSAMP         TC      VARADAR         # SELECTS VARIABLE RADAR CHANNEL.
                TC      BANKCALL
                CADR    RADSTALL

                INCR    RFAILCNT        # ADVANCE FAIL COUNTER BUT ACCEPT BAD DATA

DORSAMP2        INHINT
                CA      FLAGWRD5        # DON'T UPDATE RSTACK IF IN R77.
                MASK    R77FLBIT
                CCS     A
                TCF     +4

                DXCH    SAMPLSUM
                INDEX   RTSTLOC
                DXCH    RSTACK

                CS      RTSTLOC         # CYCLE RTSTLOC.
                AD      RTSTMAX
                EXTEND
```

```
                BZF     +3
                CA      RTSTLOC
                AD      TWO             # STORAGE IS DP
                TS      RTSTLOC
                TCF     ENDOFJOB        # CONTINUOUS SAMPLING AND 2N TRIES - GONE.

# VARIABLE RADAR DATA CALLER FOR ONE MEASUREMENT ONLY.

VARADAR         CAF     ONE             # WILL BE SENT TO RADAR ROUTINE IN A BY
                TS      BUF2            # SWCALL.
                INDEX   RTSTDEX
                CAF     RDRLOCS
                TCF     SWCALL          # NOT TOUCHING Q.

RDRLOCS         CADR    RRRANGE         # =0
                CADR    RRRDOT          # =1
                CADR    LRVELX          # =2
                CADR    LRVELY          # =3
                CADR    LRVELZ          # =4
                CADR    LRALT           # =5
```

```
# RENDEZVOUS NAVIGATION PROGRAM 20
#
# PROGRAM DESCRIPTION
#
#       MOD NO -- 2
#       BY P. VOLANTE
#
# FUNCTIONAL DESCRIPTION
#
#       THE PURPOSE OF THIS PROGRAM IS TO CONTROL THE RENDEZVOUS RADAR FROM
#       STARTUP THROUGH ACQUISITION AND LOCKON TO THE CSM AND TO UPDATE EITHER
#       THE LM OR CSM STATE VECTOR (AS SPECIFIED BY THE ASTRONAUT BY DSKY ENTRY)
#       ON THE BASIS OF THE RR TRACKING DATA.
#
# CALLING SEQUENCE --
#
#       ASTRONAUT REQUEST THROUGH DSKY V37E20E
#
# SUBROUTINES CALLED
#
#       R02BOTH (IMU STATUS CHECK)                       FLAGUP
#       GOFLASH (PINBALL-DISPLAY)                        FLAGDOWN
#       R23LEM  (MANUAL ACQUISITION)                     BANKCALL
#       LS201   (LOS DETERMINATION)                      TASKOVER
#       LS202   (RANGE LIMIT TEST)
#       R61LEM  (PREFERRED TRACKING ATTITUDE)
#       R21LEM  (RR DESIGNATE)                           ENDOFJOB
#       R22LEM  (DATA READ)                              GOPERF1
#       R31LEM  (RENDEZVOUS PARAMETER DISPLAY)
#       PRIOLARM (PRIORITY DISPLAY)
#
# NORMAL EXIT MODES --
#
#       P20 MAY BE TERMINATED IN TWO WAYS -- ASTRONAUT SELECTION OF IDLING
#       PROGRAM (P00) BY KEYING V37E00E OR BY KEYING IN V56E
#
# ALARM OR ABORT EXIT MODES --
#
#       RANGE GREATER THAN 400 NM DISPLAY
#
# OUTPUT
#
#       TRKMKCNT = NO OF RENDEZVOUS TRACKING MARKS TAKEN (COUNTER)
#
# ERASABLE INITIALIZATION REQUIRED
#
# FLAGS SET + RESET
#
#       SRCHOPT, RNDVZFLG, ACMODFLG, VEHUPFLG, UPDATFLG, TRACKFLG
#
# DEBRIS
#
#       CENTRALS -- A,Q,L

                SBANK=  LOWSUPER        # FOR LOW 2CADR'S.

                BANK    33
                SETLOC  P20S
                BANK
```

```
                EBANK=  LOSCOUNT
                COUNT*  $$/P20
PROG22          =       PROG20
PROG20          TC      2PHSCHNG
                OCT     4
                OCT     05022
                OCT     26000           # PRIORITY 26
                TC      LUNSFCHK        # CHECK IF ON LUNAR SURFACE
```

```
                TC      ORBCHGO         # YES
                TC      PROG20A -2      # NO -- CONTINUE WITH P20
ORBCHGO         TC      UPFLAG          # SET VEHUPFLG -- CSM STATE
                ADRES   VEHUPFLG        # VECTOR TO BE UPDATED
                CAF     ONE             # SET R2 FOR OPTION CSM WILL NOT
                TS      OPTION2         # CHANGE PRESENT ORBIT
                CAF     OCT00012
                TC      BANKCALL        # DISPLAY ASSUMED CSM ORBIT OPTION
                CADR    GOPERF4
                TC      GOTOPOOH        # TERMINATE
                TC      ORBCHG1         # PROCEED VALUE OF ASSUMED OPTION OK
                TC      -5              # R2 LOADED THRU DSKY
ORBCHG1         CS      P22ONE
                AD      OPTION2
                EXTEND
                BZF     PROG20A
                CAF     V06N33*
                TC      BANKCALL        # FLASH VERB-NOUN TO REQUEST ESTIMATED
                CADR    GOFLASH         # TIME OF LAUNCH
                TC      GOTOPOOH        # TERMINATE
                TC      ORBCHG2         # PROCEED VALUES OK
                TC      -5              # TIME LOADED THRU DSKY
ORBCHG2         TC      INTPRET
                GOTO
                        ORBCHG3
                BANK    32
                SETLOC  P20S4
                BANK
                COUNT*  $$/P20

ORBCHG3         CALL
                        INTSTALL
                DLOAD
                        TIG
                STORE   LNCHTM
                STORE   TDEC1           # ESTIMATED LAUNCH TIME
                CLEAR   CLEAR
                        VINTFLAG        # LM INTEGRATION
                        INTYPFLG        # PRECISION -- ENCKE
                CLEAR   CLEAR
                        DIMOFLAG        # NO W-MATRIX
                        D6OR9FLG
                CALL
                        INTEGRV         # PLANETARY INERTIAL ORIENTATION
                CALL
                        GRP2PC
                VLOAD
                        RATT1
                STODL   RSUBL           # SAVE LM POSITION
                        TAT
```

```
                STCALL  TDEC1
                        INTSTALL
                SET     CLEAR
                        VINTFLAG        # CSM INTEGRATION
                        INTYPFLG
                CLEAR   BOFF
                        DIMOFLAG
                        RENDWFLG        # W MATRIX VALID
                        NOWMATX         # NO
                SET     SET             # YES -- SET FOR W MATRIX
                        DIMOFLAG
                        D6OR9FLG
NOWMATX         CALL
                        INTEGRV         # CSM INTEGRATION
                CALL
                        GRP2PC
                VLOAD
                        VATT1
                STOVL   VSUBC           # SAVE CSM POSITION
                        RATT1
                STORE   RSUBC           # SAVE CSM POSITION
                VXV     UNIT            # COMPUTE NORMAL TO CSM ORBITAL PLANE
                        VSUBC           # NSUB1=UNIT(R(CM) CROSS V(CM))
                STOVL   20D             # SAVE NSUB1
                        RSUBL           # COMPUTE ESTIMATED ORBITAL
                VXV     UNIT            # PLANE CHANGE
                        20D             # UCSM = UNIT(R(LM) CROSS NSUB1)
                STOVL   UCSM
                        RSUBC           # COMPUTE ANGLE BETWEEN UCSM
                UNIT    DOT             # AND RSUBC
                        UCSM            # COS A = UCSM DOT UNIT (R(CM))
                SL1
                STORE   CSTH            # SAVE DOE TIME-THETA SUBROUTINE
                DSQ     BDSU            # COMPUTE SINE A
                        ONEB-2
                SQRT
                STOVL   SNTH            # SAVE FOR TIME-THETA SUBROUTINE
                        RSUBC           # POSITION OF CSM AT EST. LAUNCH
                STOVL   RVEC            # TIME FOR TIME-THETHA  B-27
                        VSUBC           # VELOCITY OF CSM AT EST. LAUNCH.
                VCOMP
                STORE   VVEC            # TIME FOR TIME THETA   B-5
                CLEAR   CALL
                        RVSW
                        TIMETHET
                VCOMP
                STORE   NEWVEL          # TERMINAL VELOCITY OF CSM
                DLOAD
                        T
                STOVL   TRANSTM         # TRANSFER TIME
```

```
                        NEWVEL
                ABVAL
                STOVL   20D
                        0D
                STORE   NEWPOS          # TERMINAL POSITION OF CSM
                VXV     UNIT            # COMPUTE NORMAL TO SCM ORBITAL PLANE
                        RSUBL           # NSUB2 = UNIT(NEWPOS CROSS R(LM))
                VXV     UNIT            # ROTATE TERMINAL VEL INTO DESIRED
                        NEWPOS          # ORBITAL PLANE
                VXSC    VSL1            # VSUBC = ABVAL(NEWVEL) $ UNIT (NSUB2
                        20D
                STCALL  NCSMVEL         # NEW CSM VELOCITY
                        GRP2PC
                CALL
                        INTSTALL
                DLOAD   BDSU
                        TRANSTM         # LAUNCH TIME -- TRANSFER TIME
                        LNCHTM
                STOVL   TET
                        NEWPOS
                STORE   RCV
                STOVL   RRECT
                        NCSMVEL
                STCALL  VRECT
                        MINIRECT
                AXT,2   CALL
                        2
                        ATOPCSM
                CALL
                        INTWAKE0
                EXIT
                TC      BANKCALL
                CADR    PROG20A

                BANK    24
                SETLOC  P20S
                BANK
                COUNT*  $$/P20

                TC      DOWNFLAG        # RESET VEHUPFLG -- LM STATE VECTOR
                ADRES   VEHUPFLG        # TO BE UPDATED
PROG20A         TC      BANKCALL
                CADR    R02BOTH
                TC      UPFLAG
                ADRES   UPDATFLG        # SET UPDATE FLAG
                TC      UPFLAG
                ADRES   TRACKFLG        # SET TRACK FLAG
                TC      UPFLAG
                ADRES   RNDVZFLG        # SET RENDEZVOUS FLAG
                TC      DOWNFLAG
                ADRES   SRCHOPTN        # INSURE SEARCH OPTION OFF
```

```
                TC      DOWNFLAG        # ALSO MANUAL ACQUISITION FLAG RESET
                ADRES   ACMODFLG
                TC      DOWNFLAG        # TURN OFF R04FLAG TO ENSURE GETTING
                ADRES   R04FLAG         # ALARM 521 IF CAN'T READ RADAR
                TC      DOWNFLAG        # ENSURE R25 GIMBAL MONITOR IS ENABLED
                ADRES   NORRMON         # (RESET NORRMON FLAG)
                TC      DOWNFLAG        # RESET LOS BEING COMPUTED FLAG
                ADRES   LOSCMFLG
                TC      CLRADMOD
P20LEM1         TC      PHASCHNG
                OCT     04022
                CAF     ZERO            # ZERO MARK COUNTER
                TS      MARKCTR
                TC      INTPRET         # LOS DETERMINATION ROUTINE
                RTB
                        LOADTIME
                STCALL  TDEC1
                        LPS20.1
                CALL
                        LPS20.2         # TEST RANGE R/UTINE
                EXIT
                INDEX   MPAC
                TC      +1
                TC      P20LEMA         # NORMAL RETURN WITHIN 400 N M
526ALARM        CAF     ALRM526         # ERROR EXIT -- RANGE > 400 N. MI.
                TC      BANKCALL
                CADR    PRIOLARM
                TC      GOTOV56         # TERMINATE EXITS P20 VIA V56 CODING
                TC      -4              # PROC (ILLEGAL)
                TC      P20LEM1         # ENTER RECYCLE
                TC      ENDOFJOB

P20LEMA         TC      PHASCHNG
                OCT     04022
                TC      LUNSFCHK        # CHECK LUNAR SURFACE FLAG (P22 FLAG)
                TC      P20LEMB
                TC      BANKCALL
                CADR    R61LEM          # PREFERRED TRACKING ATTITUDE ROUTINE
P20LEMB         TC      PHASCHNG
                OCT     05022           # RESTART AT PRIORITY 10 TO ALLOW V37
                OCT     10000           # REQUESTED PROGRAM TO RUN FIRST
                CAF     PRIO26          # RESTORE PRIORITY 26
                TC      PRIOCHNG
                CA      FLAGWRD1        # IS THE TRACK FLAG SET
                MASK    TRACKBIT
                EXTEND
                BZF     P20LEMWT        # BRANCH -- NO -- WAIT FOR IT TO BE SET
P20LEMB7        CAF     BIT2            # IS RR AUTO MODE DISCRETE PRESENT
                EXTEND
```

```
                RAND    CHAN33
                EXTEND
                BZF     P20LEMB3        # YES -- DO AUTOMATIC ACQUISITION (R21)

P20LEMB5        CS      OCT24           # RADAR NOT IN AUTO CHECK IF
                AD      MODREG          # MAJOR MODE IS 20
                EXTEND
                BZF     P20LEMB6        # BRANCH -- YES -- OKAY TO DO PLEASE PERFORM

                AD      NEG2            # ALSO CHECK FOR P22
                EXTEND
                BZF     P20LEMB6        # BRANCH -- YES OK TO DO PLEASE PERFORM
                CAF     ALRM514         # TRACK FLAG SET -- FLASH PRIORITY ALARM 514 --
                TC      BANKCALL        # RADAR GOES OUT OF AUTO MODE WHILE IN USE
                CADR    PRIOLARM
                TC      GOTOV56         # TERMINATE EXITS VIA V56
                TC      P20LEMB         # PROCEED AND ENTER BOTH GO BACK
                TC      P20LEMB         # TO CHECK AUTO MODE AGAIN
                TC      ENDOFJOB
P20LEMB6        CAF     OCT201          # REQUEST RR AUTO MODE SELECTION
                TC      BANKCALL
                CADR    GOPERF1
                TC      GOTOV56         # TERMINATE EXITS P20 VIA V56 CODING
                TC      P20LEMB         # PROCEED CHECKS AUTO MODE DISCRETE AGAIN
                TC      LUNSFCHK        # ENTER INDICATES MANUAL ACQUISITION (R23)
                TC      P20LEMB2        # YES -- R23 NOT ALLOWED -- TURN ON OPR ERROR
                TC      R23LEM          # NO -- DO MANUAL ACQUISITION

P20LEMB1        TC      UPFLAG          # RETURN FROM R23 -- LOCKON ACHIEVED
                ADRES   ACMODFLG        # SET MANUAL FLAG AND GO BACK TO CHECK
                TC      P20LEMB         # RR AUTO MODE

P20LEMB2        TC      FALTON          # TURNS ON OPERATOR ERROR LIGHT ON DSKY
                TC      P20LEMB         # AND GOES BACK TO CHECK AUTO MODE

P20LEMB3        CS      RADMODES        # ARE RR CDUS BEING ZEROED
                MASK    RCDUOBIT
                EXTEND
                BZF     P20LEMB4        # BRANCH -- YES -- WAIT
                CAF     BIT13-14        # IS SEARCH OR MANUAL ACQUISITION FLAG SET
                MASK    FLAGWRD2
                EXTEND
                BZF     P20LEMC3        # ZERO MEANS AUTOMATIC RR ACQUISITION
                TC      DOWNFLAG        # RESET TO AUTO MODE
                ADRES   SRCHOPTN
```

```
                TC      DOWNFLAG
                ADRES   ACMODFLG
                TC      P20LEMWT        # WAIT 2.5 SECONDS THEN GO TO RR DATA READ

P20LEMB4        CAF     250DEC
                TC      BANKCALL        # WAIT 2.5 SECONDS WHILE RR CDUS ARE BEING
                CADR    DELAYJOB        # ZEROED -- THEN GO BACK AND CHECK AGAIN
                TC      P20LEMB3

P20LEMC3        TC      INTPRET
                RTB
                        LOADTIME
                STCALL  TDEC1
                        UPPSV
P20LEMC4        EXIT
P20LEMC         TC      PHASCHNG
                OCT     04022
                CAE     FLAGWRD0        # IS THE RENDEZVOUS FLAG SET
                MASK    RNDVZBIT
                EXTEND
                BZF     ENDOFJOB        # NO -- EXIT P20
                CAE     FLAGWRD1        # IS TRACK FLAG SET (BIT 5 FLAGWORD 1)
                MASK    TRACKBIT
                EXTEND
                BZF     P20LEMD         # BRANCH -- TRACK FLAG NOT ON -- WAIT 15 SECONDS
P20LEMF         TC      R21LEM

P20LEMWT        CAF     250DEC
                TC      TWIDDLE         # USE INSTEAD OF WAITLIST SINCE SAME BANK
                ADRES   P20LEMC1        # WAIT 2.5 SECONDS
                CAE     FLAGWRD1        # IS TRACK FLAG SET
                MASK    TRACKBIT
                EXTEND
                BZF     ENDOFJOB        # NO -- EXIT WITHOUT DOING 2.7 PHASE CHANGE
P20LMWT1        TC      PHASCHNG
                OCT     40072
                TC      ENDOFJOB

P20LEMC1        CAE     FLAGWRD0        # IS RENDEZVOUS FLAG SET
                MASK    RNDVZBIT
                EXTEND
                BZF     TASKOVER        # NO -- EXIT P20/R22
                CAE     FLAGWRD1        # IS TRACK FLAG SET
                MASK    TRACKBIT
                EXTEND
                BZF     P20LEMC2        # NO -- DON'T SCHEDULE R22 JOB
```

```
                CAF     PRIO26          # YES -- SCHEDULE R22 JOB (RR DATA READ)
                TC      FINDVAC
                EBANK=  LOSCOUNT
                2CADR   R22LEM42

                TC      TASKOVER

P20LEMC2        TC      FIXDELAY        # TRACK FLAG NOT SET, WAIT 15 SECONDS
                DEC     1500            # AND CHECK AGAIN

                TC      P20LEMC1

P20LEMD         CAF     1500DEC
                TC      TWIDDLE         # WAITLIST FOR 15 SECONDS
                ADRES   P20LEMD1
                TC      ENDOFJOB

P20LEMD1        CAE     FLAGWRD1        # IS TRACK FLAG SET
                MASK    TRACKBIT
                CCS     A
                TCF     P20LEMD2        # YES -- SCHEDULE DESIGNATE JOB
                TC      FIXDELAY        # NO -- WAIT 15 SECONDS
                DEC     1500
                TC      P20LEMD1

P20LEMD2        CAF     PRIO26          # SCHEDULE JOB TO DO R21
                TC      FINDVAC
                EBANK=  LOSCOUNT
                2CADR   P20LEMC3        # START AT PERM.  MEMORY INTEGRATION

                TC      TASKOVER

250DEC          DEC     250
ALRM526         OCT     00526
OCT201          OCT     00201
ALRM514         OCT     514
MAXTRIES        DEC     60
OCT00012        OCT     00012
P22ONE          OCT     00001
ONEB-2          2DEC    1.0 B-2

V06N33*         VN      0633
UPPSV           STQ     CALL            # UPDATES PERMANENT STATE VECTORS
                        LS21X           #        TO PRESENT TIME
                        INTSTALL
                CALL
```

```
                                SETIFLGS
                      BOF       SET             # IF W-MATRIX INVALID, DON'T INTEGRATE IT
                                RENDWFLG
                                UPPSV1
                                DIMOFLAG        # SET DIMOFLAG TO INTEGRATE W-MATRIX
                      BON       SET
                                SURFFLAG        # IF ON LUNAR SURFACE W IS 6X6
                                UPPSV5
                                D6OR9FLG        # OTHERWISE 9X9
        UPPSV5        BOF
                                VEHUPFLG
                                UPPSV3
        UPPSV1        SET
                                VINTFLAG
                      CALL
                                INTEGRV
                      CALL                      # GROUP 2 PHASE CHANGE
                                GRP2PC          # TO PROTECT INTEGRATION
                      CALL
                                INTSTALL
                      DLOAD     CLEAR           # GET TETCSM TO STORE IN TDEC FOR LM INT.
                                TETCSM
                                VINTFLAG
        UPPSV4        CALL                      # INTEGRATE OTHER VEHICLE
                                SETIFLGS        #       WITHOUT W-MATRIX
                      STCALL    TDEC1
                                INTEGRV
                      BOFF      VLOAD
                                SURFFLAG
                                P20LEMC4
                                RCVLEM
                      VSR2
                      STOVL     LMPOS
                                VCVLEM
                      VSR2
                      STORE     LMVEL
                      GOTO
                                LS21X

        UPPSV3        CLEAR     CALL
                                VINTFLAG
                                INTEGRV
                      CALL
                                GRP2PC
                      CALL
                                INTSTALL
                      SET       DLOAD
                                VINTFLAG
                                TETLEM          # GET TETLEM TO STORE IN TDEC FOR CSM INT.
```

```
                         GOTO
                                 UPPSV4
                         EBANK=  LOSCOUNT
                         COUNT*  $$/P22
```

```
# PROGRAM DESCRIPTION
#
#       PREFERRED TRACKING ATTITUDE PROGRAM P25
#       MOD NO -- 3
#       BY P. VOLANTE
#
# FUNCTIONAL DESCRIPTION
#
#       THE PURPOSE OF THIS PROGRAM IS TO COMPUTE THE PREFERRED TRACKING
#       ATTITUDE OF THE LM TO CONTINUOUSLY POINT THE LM TRTACKING BEACON AT THE
#       CSM AND TO PERFORM THE MANEUVER TO THE PREFERRED TRACKING ATTITUDE AND
#       CONTINUOUSLY MAINTAIN THIS ATTITUDE WITHIN PRESCRIBED LIMITS.
#
# CALLING SEQUENCE --
#
#       ASTRONAUT REQUEST THROUGH DSKY V37E25E
#
# SUBROUTINES CALLED --
#
#       BANKCALL                                FLAGUP
#       R02BOTH (IMU STATUS CHECK)              ENDOFJOB
#       R61LEM  (PREF TRK ATT ROUT)             WAITLIST
#       TASKOVER                                FINDVAC
#
# NORMAL EXIT MODES --
#
#       P25 MAY BE TERMINATED IN TWO WAYS -- ASTRONAUT SELECTION OF IDLING
#       PROGRAM (P00) BY KEYING V37E00E OR BY KEYING IN V56E
#
# ALARM OR ABORT EXIT MODES --
#
#       NONE
#
# OUTPUT
#
# ERASABLE INITIALIZATION REQUIRED
#
# FLAGS SET + RESET
#
#       TRACKFLG, P25FLAG
#
# DEBRIS
#
#       NONE

                EBANK=  LOSCOUNT
                COUNT*  $$/P25
PROG25          TC      2PHSCHNG
                OCT     4               # MAKE GROUP 4 INACTIVE (VERB 37)
                OCT     05022
                OCT     26000           # PRIORITY 26

                TC      BANKCALL
                CADR    R02BOTH         # IMU STATUS CHECK
                TC      UPFLAG
                ADRES   TRACKFLG        # SET TRACK FLAG
                TC      UPFLAG
                ADRES   P25FLAG         # SET P25FLAG
P25LEM1         TC      PHASCHNG
```

```
		OCT	04022
		CAF	P25FLBIT
		MASK	STATE		# IS P25FLAG SET
		EXTEND
		BZF	ENDOFJOB
		CAF	TRACKBIT	# IS TRACKFLAG SET?
		MASK	STATE +1
		EXTEND
```

```
                BZF     P25LMWT1        # NO -- SKIP PHASE CHANGE AND WAIT 1 MINUTE
                CAF     SEVEN           # CALL R65 -- FINE PREFERRED
                TS      R65CNTR
                TC      BANKCALL        # TRACKING ATTITUDE ROUTINE
                CADR    R65LEM
                TC      P25LEM1         # THEN GO CHECK FLAGS
P25LEMWT        TC      PHASCHNG
                OCT     00112
P25LMWT1        CAF     60SCNDS
                TC      TWIDDLE         # WAIT ONE MINUTE THEN CHECK AGAIN
                ADRES   P25LEM2
                TC      ENDOFJOB
P25LEM2         CAF     PRIO14
                TC      FINDVAC
                EBANK=  LOSCOUNT
                2CADR   P25LEM1

                TC      TASKOVER
60SCNDS         DEC     6000
```

```
# DATA READ ROUTINE 22 (LEM)
# PROGRAM DESCRIPTION
#
#       MOD NO -- 2
#       BY P. VOLANTE
#
# FUNCTIONAL DESCRIPTION
#
#       TO PROCESS AUTOMATIC RR MARK DATA TO UPDATE THE STATE VECTOR OF EITHER
#       LM OR CSM AS DEFINED IN THE RENDEZVOUS NAVIGATION PROGRAM (P20)
#
# CALLING SEQUENCE --
#
#       TC      BANKCALL
#       CADR    R22LEM
#
# SUBROUTINES CALLED --
#
#       LSR22.1         GOFLASH         WAITLIST
#       LSR22.2         PRIOLARM        BANKCALL
#       LSR22.3         R61LEM
#
# NORMAL EXIT MODES --
#
#       R22 WILL CONTINUE TO RECYCLE, UPDATING STATE VECTORS WITH RADAR DATA
#       UNTIL P20 CEASES TO OPERATE (RENDEZVOUS FLAG SET TO ZERO) AT WHICH TIME
#       R22 WILL TERMINATE SELF.
#
# ALARM OR ABORT EXIT MODES --
#
#       PRIORITY ALARM
#       PRIORITY ALARM 525 LOS NOT WITHIN 3 DEGREE LIMIT
#
# OUTPUT
#
#       SEE OUTPUT FROM LSR22.3
#
# ERASABLE INITIALIZATION REQUIRED
#
#       SEE LSR22.1, LSR22.2, LSR22.3
#
# FLAGS SET + RESET
#
#       NOANGFLG
#
# DEBRIS
#
#       SEE LSR22.1, LSR22.2, LSR22.3

                EBANK=  LRS22.1X
                COUNT*  $$/R22
R22LEM          TC      PHASCHNG
                OCT     04022
                CAF     RNDVZBIT        # IS RENDEZVOUS FLAG SET?
                MASK    STATE
                EXTEND
                BZF     ENDOFJOB        # NO -- EXIT R22 AND P20
                CAF     TRACKBIT        # IS TRACKFLAG SET?
                MASK    STATE +1
```

```
                EXTEND
                BZF     R22WAIT         # NO WAIT
R22LEM12        CAF     BIT14           # IS RR AUTO TRACK ENABLE DISCRETE STILL
                EXTEND                  # ON (A MONITOR REPOSITION BY R25 CLEARS IT)
                RAND    CHAN12
                EXTEND
                BZF     P20LEMA         # NO -- RETURN TO P20
                CAF     BIT2            # YES
                EXTEND                  # IS RR AUTO MODE DISCRETE PRESENT
                RAND    CHAN33
```

```
                EXTEND
                BZF      +2              # YES CONTINUE
                TC       P20LEMB5        # NO -- SET IT
                CS       RADMODES        # ARE RR CDUS BEING ZEROED
                MASK     RCDUOBIT
                EXTEND
                BZF      R22LEM42        # CDUS BEING ZEROED
                TC       PHASCHNG        # IF A RESTART OCCURS, AND EXTRA RADAR
                OCT      00152           # READING IS TAKEN, SO BAD DATA ISN'T USED
                TC       BANKCALL        # YES READ DATA + CALCULATE LOS
                CADR     LRS22.1         # DATA READ SUBROUTINE
                INDEX    MPAC
                TC       +1
                TC       R22LEM2         # NORMAL RETURN (GOOD DATA)
                TC       P20LEMC         # COULD NOT READ RADAR -- TRY TO REDESIGNATE
                CAF      ALRM525         # RR LOS NOT WITHIN 3 DEGREES (ALARM)
                TC       BANKCALL
                CADR     PRIOLARM
                TC       GOTOV56         # TERMINATE EXITS P20 VIA V56 CODING
                TC       R22LEM1         # PROC (DISPLAY DELTA THETA)
                TC       -5              # ENTER (ILLEGAL OPTION)
                TC       ENDOFJOB

R22LEM1         TC       PHASCHNG
                OCT      04022
                CAF      V06N05          # DISPLAY DELTA THETA
                TC       BANKCALL
                CADR     PRIODSP
                TC       GOTOV56         # TERMINATE EXITS P20 VIA V56 CODING
                TC       R22LEM2         # PROC (OK CONTINUE)
                TC       P20LEMC         # ENTER (RECYCLE)
R22LEM2         TC       PHASCHNG
                OCT      04022
                TC       LUNSFCHK        # CHECK IF ON LUNAR SURFACE (P22FLAG SET)
                TC       R22LEM3         # YES -- BYPASS FLAG CHECKS AND LRS22.2
                CA       FLAGWRD1        # IS TRACK FLAG SET
                MASK     TRACKBIT
                EXTEND
                BZF      R22WAIT         # NO -- WAIT
                TC       BANKCALL        # YES
                CADR     LRS22.2         # CHECKS RR BORESIGHT WITHIN 30 DEG OF +Z
                INDEX    MPAC
                TC       +1
                TC       R22LEM3         # NORMAL RETURN (LOS WITHIN 30 OF Z-AXIS)
                TC       BANKCALL
                CADR     R61LEM
                TC       R22WAIT         # NOT WITHIN 30 DEG OF Z-AXIS
R22LEM3         CS       FLAGWRD1        # SHOULD WE BYPASS STATE VECTOR UPDATE
                MASK     NOUPFBIT        # (IS NO UPDATE FLAG SET?)
```

```
                EXTEND
                BZF     R22LEM42        # BRANCH -- YES
                CA      FLAGWRD1        # IS UPDATE FLAG SET
                MASK    UPDATBIT
                EXTEND
                BZF     R22LEM42        # UPDATE FLAG NOT SET
                CAF     PRIO26          # INSURE HIGH PRIO IN RESTART
                TS      PHSPRDT2

                TC      INTPRET
                GOTO
                        LSR22.3
R22LEM93        EXIT                    # NORMAL EXIT FROM LSR22.3
                TC      PHASCHNG        # PHASE CHANGE TO PROTECT AGAINST
                OCT     04022           # CONFLICT WITH GRP2PC ERASEABLE
                TCF     R22LEM44
R22LEM96        EXIT
                CAF     ZERO            # SET N49FLAG = ZERO TO INDICATE
                TS      N49FLAG         # V06 N49 DISPLAY HASN'T BEEN ANSWERED
                TC      PHASCHNG
                OCT     04022           # TO PROTECT DISPLAY
                CAF     PRIO27          # PROTECT DISPLAY
                TC      NOVAC
                EBANK=  N49FLAG
                2CADR   N49DSP

                TC      INTPRET
                SLOAD
                        N49FLAG
                BZE     BMN             # LOOP TO CHECK IF FLAG
                        -3              # SETTING CHANGED -- BRANCH -- NO
                        R22LEM7         # PROCEED
                EXIT                    # DISPLAY ANSERED BY RECYCLE
                TC      LUNSFCHK        # ARE WE ON LUNAR SURFACE
                TC      R22WAIT         # YES -- 15 SECOND DELAY
                CA      ZERO            # NO -- SET R65COUNTER = 0, DO FINE
                TC      R22LEM45        # TRACKING TAKE ANOTHER RADAR READING
R22LEM7         CALL                    # PROCEED
                        GRP2PC          # PHASE CHANGE AND
                GOTO                    # GO TO INCOPORATE DATA.
                        ASTOK
R22LEM44        INCR    MARKCTR         # INCREMENT COUNT OF MARKS INCORPORATED.
                TC      LUNSFCHK        # ARE WE ON LUNAR SURFACE
                TC      R22LEM46        # YES -- WAIT 2 SECONDS
                CA      FIVE            # NOT ON LUNAR SURFACE
                TC      R22LEM45        # R65COUNTER = 5
R22LEM42        TC      LUNSFCHK        # CHECK IF ON LUNAR SURFACE (P22FLAG SET)
                TC      R22LEM46        # YES -- WAIT 2 SECONDS
                CA      TWO             # NO -- SET R65COUNTER = 2
R22LEM45        TS      R65CNTR
```

```
                TC      BANKCALL
                CADR    R65LEM          # FINE PREFERRED TRACKING ATTITUDE
                TC      R22LEM
R22WAIT         CAF     1500DEC
                TC      P20LEMWT +1

R22LEM46        CAF     2SECS
                TC      BANKCALL        # WAIT 2 SECONDS AND TAKE ANOTHER MARK
                CADR    DELAYJOB
                TC      R22LEM

N49DSP          CAF     V06N49NB
                TC      BANKCALL        # EXCESSIVE STATE VECTOR UPDATE -- FLASH
                CADR    PRIODSP         # VERB 06 NOUN 49 R1=DELTA R, R2=DELTA V
                TC      GOTOV56         # TERMINATE -- EXIT R22 AND P20
                CS      ONE             # PROCEED -- N49FLAG = -1
                TS      N49FLAG         # RECYCLE -- N49FLAG = + VALUE
                TC      ENDOFJOB
R22RSTRT        TC      PHASCHNG        # IF A RESTART OCCURS WHILE READING RADAR
                OCT     00152           # COME HERE TO TAKE A RANGE-RATE READING
                TC      BANKCALL        # WHICH ISN'T USED TO PREVENT TAKING A BAD
                CADR    RRRDOT          # READING AND TRYING TO INCORPORATE THE
                TC      BANKCALL        # BAD DATA
                CADR    RADSTALL        # WAIT FOR READ COMPLETE
                TC      P20LEMC         # COULD NOT READ RADAR -- TRY TO REDISGNATE
                TC      R22LEM          # READ SUCCESSFUL -- CONTINUE AT R22

ALRM525         OCT     00525
V06N05          VN      00605
V06N49NB        VN      00649
1500DEC         DEC     1500

# LUNSFCHECK -- CLOSED SUBROUTINE TO CHECK IF ON LUNAR SURFACE (P22FLAG)
#       RETURNS TO CALLER +1 IF P22FLAG SET
#               TO CALLER +2 IF P22FLAG NOT SET

                COUNT*  $$/P22
LUNSFCHK        CS      FLAGWRD8        # CHECK IF ON LUNAR SURFACE
                MASK    SURFFBIT        # IS SURFFLAG SET?
                CCS     A               # BRANCH -- P22FLAG SET
                INCR    Q               # NOT SET
                TC      Q               # RETURN
```

```
# RR DESIGNATE ROUTINE (R21LEM)
# PROGRAM DESCRIPTION
#
#       MOD NO -- 2
#       BY P. VOLANTE
#
# FUNCTIONAL DESCRIPTION
#
#       TO POINT THE RENDEZVOUS RADAR AT THE CSM UNTIL AUTOMATIC ACQUISITION
#       OF THE CSM IS ACCOMPLISHED BY THE RADAR.  ROUTINE IS CALLED BY P20.
#
# CALLING SEQUENCE --
#
#       TC      BANKCALL
#       CADR    R21LEM
#
# SUBROUTINES CALLED --
#
#       FINDVAC         FLAGUP          ENDOFJOB        PRIOLARM
#       NOVAC           INTPRET         LPS20.1         PHASCHNG
#       WAITLIST        JOBSLEEP        JOBWAKE         FLAGDOWN
#       TASKOVER        BANKCALL        RADSTALL        RRDESSM
#
# NORMAL EXIT MODES
#
#       WHEN LOCK-ON IS ACHIEVED, BRANCH WILL BE TO P20 WHERE R22 (DATA READ
#       WILL BE SELECTED OR A NEED FOR A MANEUVER (BRANCH TO P20LEMA)
#
# ALARM OR ABORT EXIT MODES --
#
#       PRIORITY ALARM 503 WHEN LOCK-ON HASN'T BEEN ACHIEVED AFTER 30SECS --
#       THIS REQUIRES ASTRONAUT INTERFACE: SELECTION OF SEARCH OPTION OF
#       ACQUISITION
#
# OUTPUT
#
#       SEE LPS20.1, RRDESSM
#
# ERASABLE INITIALIZATION REQUIRED
#
#       RRTARGET, RADMODES ARE USED BY LPS20.1 AND RRDESSM
#
# FLAGS SET + RESET
#
#       LOSCMFLG        LOKONSW
#
# DEBRIS
#
#       SEE LPS20.1, RRSESSM

                EBANK=  LOSCOUNT
                COUNT*  $$/R21
R21LEM          CS      BIT14           # REMOVE RR SELF TRACK ENABLE
                EXTEND
                WAND    CHAN12
                TC      LUNSFCHK
                TC      R21LEM5
                CAF     ZERO            # COMMAND ANTENNA TO MODE CENTER
                TS      TANG            # IF NOT ON SURFACE -- MODE 1 -- (T=0,S=0)
```

```
                TS      TANG +1
                TC      R21LEM6
R21LEM5         CAF     BIT12
                MASK    RADMODES
                CCS     A
                TC      R21LEM10
                CAF     BIT15
                TS      TANG
                CS      HALF
                TS      TANG +1
```

```
R21LEM6         TC      DOWNFLAG
                ADRES   LOKONSW
                TC      BANKCALL
                CADR    RRDESNB
                TC      +1
                TC      BANKCALL
                CADR    RADSTALL
                TC      R21-503         # BAD RETURN FROM DESIGNATE -- ISSUE ALARM
R21LEM10        TC      UPFLAG
                ADRES   LOSCMFLG        # EVERY FOURTH PASS THRU DODES
                CAF     MAXTRIES        # ALLOW 60 PASSES (APPROX 45 SECONDS)
                TS      DESCOUNT        # TO DESIGNATE AND LOCK ON
R21LEM2         CAF     THREE
                TS      LOSCOUNT
R21LEM1         TC      INTPRET
                RTB     DAD
                        LOADTIME
                        HALFSEC         # EXTRAPOLATE TO PRESENT TIME + .5 SEC.
                STCALL  TDEC1           # LOS DETERMINATION ROUTINE
                        LPS20.1
                EXIT
R21LEM3         TC      UPFLAG          # SET LOKONSW TO RADAR -- ON DESIRED
                ADRES   LOKONSW
                TC      DOWNFLAG
                ADRES   NORRMON
                TC      INTPRET
                CALL                    # INPUT (RRTARGET UPDATED BY LPS20.1)
                        RRDESSM         # DESIGNATE ROUTINE
                EXIT
                TC      R21LEM4         # LOS NOT IN MODE 2 COVERAGE
                                        # ON LUNAR SURFACE
                TC      P20LEMA         # VEHICLE MANEUVER REQUIRED.
                TC      BANKCALL        # NO VEHICLE MANEUVER REQUIRED
                CADR    RADSTALL        # WAIT FOR DESIGNATE COMPLETE -- LOCKON OR
                TC      +2              # BAD END -- LOCKON NOT ACHIEVED IN 60 TRIES
                TC      R21END          # EXIT ROUTINE RETURN TO P20 (LOCK-ON)
R21-503         CAF     ALRM503         # ISSUE ALARM 503
                TC      BANKCALL
                CADR    PRIOLARM
                TC      GOTOV56         # TERMINATE EXITS P20 VIA V56 CODING
                TC      R21SRCH         # PROC
                TC      P20LEMC3
                TC      ENDOFJOB
R21END          TC      DOWNFLAG
                ADRES   LOSCMFLG        # RESET LOSCMFLG
                TC      R21DISP         # PUT UP VERIFY MAIN LOBE LOCKON DISPLAY
R21SRCH         TC      PHASCHNG
                OCT     04022
                TC      R24LEM          # SEARCH ROUTINE
ALRM503         OCT     00503
```

```
ALRM527         OCT     527

R21LEM4         CAF     MAXTRIES        # SET UP COUNTER FOR
                TS      REPOSCNT        # 60 PASSES (APPROX 600 SECS.)
                TC      UPFLAG
                ADRES   FSPASFLG        # SET FIRST PASS FLAG
                TC      DOWNFLAG        # RESET LOS BEING
                ADRES   LOSCMFLG        # COMPUTED FLAG
                TC      INTPRET
R21LEM12        RTB
                        LOADTIME
                DAD
                        TENSEC          # TIME T = T + 10 SECS.
                STORE   REPOSTM         # SAVE FOR LONGCALL AND UPPSV
                STCALL  TDEC1
                        LPS20.1         # COMPUTE LOS AT TIME T
                CALL
                        RRDESSM
                EXIT
                TC      R21LEM13        # LOS NOT IN MODE 2 COVERAGE
                TC      ENDOFJOB        # VEHICLE MANEUVER REQUIRED
                TC      KILLTASK
                CADR    BEGDES
                TC      INTPRET
                BOF
                        FSPASFLG        # FIRST PASS THRU REPOSITION
                        R21LEMB         # NO -- GO TO CONTINUOUS DESIGNATE
                CLRGO
                        FSPASFLG        # YES -- RESET FIRST PASS FLAG
                        R21LEM7 +1
R21LEM13        CCS     REPOSCNT        # HAVE WE TRIED 60 TIMES?
                TC      R21LEM7         # NO -- ADD 10 SECS.  RECOMPUTE LOS
                TC      R21LEM11        # YES -- PUT OUT ALARM 530
R21LEM7         TS      REPOSCNT
                TC      INTPRET
                DLOAD   GOTO
                        REPOSTM
                        R21LEM12 +2
R21LEMB         DLOAD
                        REPOSTM
                STCALL  TDEC1
                        UPPSV
                EXIT
                TC      UPFLAG          # SET RADMODES BIT 15 FOR
                ADRES   CDESFLAG        # CONTINUOUS DESIGNATION
                TC      DOWNFLAG
                ADRES   LOKONSW
                TC      UPFLAG
                ADRES   NORRMON
```

```
                TC      BANKCALL
                CADR    RRDESNB
                TC      +1
                TC      INTPRET
                RTB     BDSU
                        LOADTIME        # COMPUTE DELTA TIME
                        REPOSTM         # FOR LONGCALL
                STORE   DELTATM
                EXIT
                EXTEND
                DCA     DELTATM
                TC      LONGCALL
                EBANK=  LOSCOUNT
                2CADR   R21LEM9

                TC      ENDOFJOB
R21LEM9         TC      KILLTASK
                CADR    STDESIG
                TC      CLRADMOD
                CAF     PRIO26
                TC      FINDVAC
                EBANK=  LOSCOUNT
                2CADR   R21LEM10

                TC      TASKOVER
R21LEM11        CAF     ALRM530         # ALARM 530 -- LOS NOT IN COVERAGE
                TC      BANKCALL        # AFTER TRYING TO DESIGNATE FOR
                CADR    PRIOLARM        # 600 SECS.
                TC      GOTOV56
                TC      GOTOV56
                TC      GOTOV56
                TC      ENDOFJOB
ALRM530         OCT     00530
TENSEC          2DEC    1000 B-28

HALFSEC         2DEC    50

R21DISP         TC      PHASCHNG
                OCT     04022
                CAF     V06N72PV        # FLASH V 50 N 72 -- PLEASE PERFORM RR
                TC      BANKCALL        # MAIN LOBE LOCKON VERIFICATION
                CADR    GOPERF2R
                TC      GOTOV56         # TERMINATE EXITS VIA V 56
                TC      P20LEMWT        # PROCEED CONTINUES TO R22
                TC      -5              # ENTER ILLEGAL
                CAF     BIT7
                TC      LINUS           # SET BITS TO MAKE THIS A PRIORITY DISPLAY
                TC      ENDOFJOB
```

V06N72PV VN 00672

```
# MANUAL ACQUISITION ROUTINE R23LEM
# PROGRAM DESCRIPTION
#
#       MOD NO -- 2
#       BY P. VOLANTE
#
# FUNCTIONAL DESCRIPTION
#
#       TO ACQUIRE THE CSM BY MANUAL OPERATION OF THE RENDEZVOUS RADAR
#
# CALLING SEQUENCE --
#
#       TC      R23LEM
#
# SUBROUTINES CALLED
#
#       BANKCALL        R61LEM
#       SETMINDB        GOPERF1
#
# NORMAL EXIT MODES --
#
#       IN RESPONSE TO THE GOPERF1,     SELECTION OF ENTER WILL RECYCLE R23
#                                       SELECTION OF PROC WILL CONTINUE R23
#                                       SELECTION OF TERM WILL TERMINATE R23 + P20
#
# ALARM OR ABORT EXIT MODES --
#
#       SEE NORMAL EXIT MODES ABOVE
#
# OUTPUT
#
#       N.A.
#
# ERASABLE INITIALIZATION REQUIRED --
#
#       ACMODFLG MUST BE SET TO 1 (MANUAL MODE)

                EBANK=  GENRET
                COUNT*  $$/R23
R23LEM          TC      UPFLAG          # SET NO ANGLE MONITOR FLAG
                ADRES   NORRMON
                INHINT
                TC      IBNKCALL        # SELECT MINIMUM DEADBAND
                CADR    SETMINDB
                RELINT
R23LEM1         CAF     BIT14           # ENABLE TRACKER
                EXTEND
                WOR     CHAN12
                CAF     OCT205
                TC      BANKCALL
                CADR    GOPERF1
                TC      R23LEM2         # TERMINATE
                TC      R23LEM11        # PROCEDE
                TC      R23LEM3         # ENTER -- DO ANOTHER MANEUVER
R23LEM11        INHINT
                TC      RRLIMCHK        # YES -- CHECK IF ANTENNA IS WITHIN LIMITS
                ADRES   CDUT
                TC      OUTOFLIM        # NOT WITHIN LIMITS
                TC      IBNKCALL        # RESTORE DEADBAND TO
```

```
                    CADR     RESTORDB         # ASTRONAUT SELECTED VALUE
                    RELINT
                    TC       DOWNFLAG         # CLEAR NO ANGLE MONITOR FLAG
                    ADRES    NORRMON
                    TC       P20LEMB1         # RADAR IS LOCKED ON CONTINUE IN P20
OUTOFLIM            RELINT
```

```
                CAF     OCT501PV
                TC      BANKCALL        # ISSUE ALARM -- RR ANTENNA NOT WITHIN
                CADR    PRIOLARM        # LIMITS
                TC      R23LEM2         # TERMINATE -- EXIT R23 TO R00 (GO TO POOH)
                TC      OUTOFLIM +1     # PROCEED ILLEGAL
                TC      R23LEM3         # RECYCLE -- TO ANOTHER MANEUVER
                TC      ENDOFJOB
R23LEM2         TC      DOWNFLAG        # CLEAR NO ANGLE MONITOR FLAG
                ADRES   NORRMON
                TC      GOTOV56         # AND EXIT VIA V56
R23LEM3         TC      BANKCALL
                CADR    R61LEM
                TC      R23LEM1

OCT501PV        OCT     501
OCT205          OCT     205
```

```
# SEARCH ROUTINE R24LEM
# PROGRAM DESCRIPTION
#
#       MOD NO -- 2
#       BY P. VOLANTE
#
# FUNCTIONAL DESCRIPTION
#
#       TO ACQUIRE THE CSM BY A SEARCH PATTERN WHEN THE RENDEZVOUS RADAR HAS
#       FAILED TO ACQUIRE TEH CSM IN THE AUTOMATIC TRACKING MODE AND TO ALLOW
#       THE ASTRONAUT TO CONFIRM THAT REACQUISITION HAS NOT BEEN IN SIDELOBE.
#
# CALLING SEQUENCE
#
#       CAF     PRIONN
#       TC      FINDVAC
#       EBANK=  DATAGOOD
#       2CADR   R24LEM
#
# SUBROUTINES CALLED
#
#       FLAGUP          FLAGDOWN        BANKCALL
#       R61LEM          GOFLASHR        FINDVAC
#       ENDOFJOB        NOVAC           LSR24.1
#
# NORMAL EXIT MODES --
#
#       ASTRONAUT RESPONSE TO DISPLAY OF OMEGA AND DATAGOOD.  HE CAN EITHER
#       REJECT BY TERMINATING (SEARCH OPTION AND RESELECTING P20) OR ACCEPT BY
#       PROCEEDING (EXIT ROUTINE AND RETURN TO AUTO MODE IN P20)
#
# ALARM OR ABORT EXIT MODES --
#
#       SEE NORMAL EXIT MODES ABOVE
#
# OUTPUT --
#
#       SEE OUTPUT FROM LSR24.1 + R61LEM
#
# ERASABLE INITIALIZATION REQUIRED
#
#       SET INPUT FOR LSR24.1
#
# FLAGS SET + RESET
#
#       SRCHOPT, ACMODFLG

                EBANK=  DATAGOOD
                COUNT*  $$/R24
R24LEM          TC      UPFLAG
                ADRES   SRCHOPTN        # SET SRCHOPT FLAG
                TC      DOWNFLAG        # RESET LOS BEING COMPUTED FLAG TO MAKE
                ADRES   LOSCMFLG        # SURE DODES DOESN'T GO TO R21
R24LEM1         CAF     ZERO
                TS      DATAGOOD        # ZERO OUT DATA INDICATOR
                TS      OMEGAD          # ZERO OMEGA DISPLAY REGS
                TS      OMEGAD +1       # ZERO OMEGA DISPLAY REGS
R24LEM2         TC      PHASCHNG
                OCT     04022
```

```
         CAF     V16N80
         TC      BANKCALL
         CADR    PRIODSPR
         TC      GOTOV56
         TC      R24END          # PROCEED EXIT R24 TO P20LEM1

         TC      R24LEM3         # RECYCLE -- CALL R61 TO MANEUVER S/C
```

```
                TC      BANKCALL
                CADR    LRS24.1
R24END          TC      KILLTASK
                CADR    CALLDGCH
                TC      CLRADMOD        # CLEAR BITS 10 & 15 OF RADMODES.
                TCF     P20LEM1         # AND GO TO 400 MI. RANGE CHECK IN P20

                BLOCK   3
                SETLOC  FFTAG6
                BANK
                COUNT*  $$/R24

CLRADMOD        CS      BIT10+15
                INHINT
                MASK    RADMODES
                TS      RADMODES
                CS      BIT2            # DISABLE RR ERROR COUNTERS
                EXTEND
                WAND    CHAN12          # USER WILL RELINT

                TC      Q

BIT10+15        OCT     41000
                BANK    24
                SETLOC  P20S
                BANK
                COUNT*  $$/R24

R24LEM3         TC      PHASCHNG
                OCT     04022
                TC      KILLTASK
                CADR    CALLDGCH        # KILL WAITLIST FOR NEXT POINT IN PATTERN
                TC      CLRADMOD        # CLEAR BITS 10 + 15 OF RADMODES TO KILL
                RELINT                  # HALF SECOND DESIGNATE LOOP
                CAF     .5SEC
                TC      BANKCALL        # WAIT FOR DESIGNATE LOOP TO DIE
                CADR    DELAYJOB
                TC      LUNSFCHK        # CHECK IF ON LUNAR SURFACE
                TC      R24LEM4         # YES -- DON'T DO ATTITUDE MANEUVER
                TC      BANKCALL        # CALL R61 TO DO PREFERRED TRACKING
                CADR    R61LEM          # ATTITUDE MANEUVER
R24LEM4         CAF     ZERO            # ZERO OUT RADCADR (WHICH WAS SET BY
                TS      RADCADR         # ENDRADAR WHEN DESIGNATE STOPPED) SO THAT
                                        # RRDESSM WILL RETURN TO CALLER
                TC      R24LEM2         # AND GO BACK TO PUT UP V16 N80 DISPLAY

V16N80          VN      01680
```

```
# PREFERRED TRACKING ATTITUDE ROUTINE R61LEM
# PROGRAM DESCRIPTION
#
#       MOD NO: 3               DATE: 4-11-67
#       MOD BY: P. VOLANTE, SDC
#
# FUNCTIONAL DESCRIPTION --
#
#       TO COMPUTE THE PREFERRED TRACKING ATTITUDE OF THE LM TO ENABLE RR
#       TRACKING OF THE CSM AND TO PERFORM THE MANEUVER TO THE PREFERRED
#       ATTITUDE.
#
# CALLING SEQUENCE --
#
#       TC      BANKCALL
#       CADR    R61LEM
#
# SUBROUTINES CALLED
#
#       LPS20.1         VECPOINT
#       KALCMAN3
#
# NORMAL EXIT MODES --
#
#       NORMAL RETURN IS TO CALLER + 1
#
# ALARM OR ABORT EXIT MODES --
#
#       TERMINATE P20 + R61 BY BRANCHING TO P20END IF BOTH TRACKFLAG +
#       RENDEZVOUS FLAG ARE NOT SET.
#
# OUTPUT --
#
#       SEE OUTPUT FOR LPS20.1 + ATTITUDE MANEUVER ROUTINE (R60)
#
# ERASABLE INITIALIZATION REQUIRED
#
#       GENRET USED TO SAVE Q FOR RETURN
#
# FLAGS SET + RESET
#
#       3AXISFLG
#
# DEBRIS
#
#       SEE SUBROUTINES

                SETLOC  R61
                BANK
                EBANK=  LOSCOUNT
                COUNT*  $$/R61
R61LEM          TC      MAKECADR
                TS      GENRET
                TC      UPFLAG          # SET R61 FLAG
                ADRES   R61FLAG
                TC      R61C+L01
R65LEM          TC      MAKECADR
                TS      GENRET
                TC      DOWNFLAG        # RESET R61 FLAG
```

```
                 ADRES   R61FLAG
R61C+L01         CAF     TRACKBIT         # TRACKFLAG
                 MASK    STATE +1
                 EXTEND
                 BZF     R65WAIT          # NOT SET
R61C+L03         TC      INTPRET
                 VLOAD
```

```
                        HIUNITZ
                STORE   SCAXIS          # TRACK AXIS UNIT VECTOR
R61LEM1         RTB     DAD
                        LOADTIME        # EXTRAPOLATE FORWARD TO CENTER
                        3SECONDS        # SIX SECOND PERIOD.
                STCALL  TDEC1
                        LPS20.1         # LOS DETERMINATION + VEH ATTITUDE
                VLOAD
                        RRTARGET
                STORE   POINTVSM
                RTB     CALL            # GET DESIRED CDU'S FOR VECPNT1
                        READCDUD
                        VECPNT1         # COMPUTES FINAL ANGLES FROM PRESENT CDUDS
                STORE   CPHI            # STORE FINAL ANGLES -- CPHI, CTHETA, CPSI
                EXIT
                TC      PHASCHNG
                OCT     04022
                CAF     TRACKBIT        # IS TRACK FLAG SET
                MASK    FLAGWRD1
                EXTEND
                BZF     R65WAIT
                TC      BANKCALL
                CADR    G+N,AUTO        # CHECK FOR AUTO MODE
                CCS     A
                TC      R61C+L04        # NOT IN AUTO
                TC      INTPRET
                VLOAD   CALL
                        RRTARGET
                        CDU*SMNB
                DLOAD   DSU             # GET PHI -- ARCCOS OF Z-COMPONENT OF LOS
                        MPAC +5
                        COS15DEG
R61LEM2         BMN     EXIT            # BRANCH -- PHI > 15 DEGREES
                        R61C+L05        # PHI GRE 10DEG
                EBANK=  CDUXD
                CAF     EBANK6
                TS      EBANK
                INHINT
                EXTEND
                DCA     CPHI
                DXCH    CDUXD
                CA      CPSI
                TS      CDUZD
                RELINT
                EBANK=  LOSCOUNT
                CAF     EBANK7
                TS      EBANK
                TC      R61C+L06
R61C+L05        EXIT
                INHINT
```

```
                TC      IBNKCALL
                FCADR   ZATTEROR
                TC      IBNKCALL
                FCADR   SETMINDB        # REDUCE ATTITUDE ERROR
                TC      DOWNFLAG
                ADRES   3AXISFLG
                TC      UPFLAG
                ADRES   PDSPFLAG        # SET PRIORITY DISPLAY FLAG
                TC      BANKCALL
                CADR    R60LEM
                INHINT
                TC      IBNKCALL
                FCADR   RESTORDB
                TC      PHASCHNG
                OCT     04022
                TC      DOWNFLAG
                ADRES   PDSPFLAG        # RESET PRIORITY DISPLAY FLAG
R61C+L06        CA      FLAGWRD1
                MASK    R61FLBIT
                CCS     A
                TC      R61C+L4
                CCS     R65CNTR
                TC      +2
                TC      R61C+L4         # R65CNTR = 0 - EXIT ROUTINE
                TS      R65CNTR
                CAF     06SEC
                TC      TWIDDLE
                ADRES   R61C+L2
                TC      ENDOFJOB
R61C+L2         CAF     PRIO26
                TC      FINDVAC
                EBANK=  LOSCOUNT
                2CADR   R61C+L01

                TC      TASKOVER
R61C+L04        TC      BANKCALL        # TO CONVERT ANGLES TO FDAI
                CADR    BALLANGS
                TC      R61C+L06
R61C+L4         CAE     GENRET
                TCF     BANKJUMP        # EXIT R61
R61C+L1         CAF     BIT7+9PV        # IS RENDEZVOUS OR P25FLAG SET
                MASK    STATE
                EXTEND
                BZF     ENDOFJOB        # NO -- EXIT ROUTINE AND PROGRAM.
                TC      R61C+L06        # YES EXIT ROUTINE
R65WAIT         TC      POSTJUMP
                CADR    P20LEMWT

BIT7+9PV        OCT     00500
```

```
COS15DEG        2DEC    0.96593 B-1

06SEC           DEC     600
PHI             EQUALS  20D
READCDUD        INHINT                  # READS DESIRED CDU'S AND STORES IN
                CAF     EBANK6          # MPAC TP EXITS WITH MODE SET TO TP
                XCH     EBANK
                TS      RUPTREG1
                EBANK=  CDUXD
                CA      CDUXD
                TS      MPAC
                EXTEND
                DCA     CDUYD
                DXCH    MPAC +1
                CA      RUPTREG1
                TS      EBANK
                RELINT
                TCF     TMODE
                BLOCK   02
                SETLOC  RADARFF
                BANK

                EBANK=  LOSCOUNT
                COUNT*  $$/RRSUB
```

```
# THE FOLLOWING SUBROUTINE RETURNS TO CALLER +2 IF THE ABSOLUTE VALUE OF VALUE OF C(A) IS GREATER THAN THE
# NEGATIVE OF THE NUMBER AT CALLER +1.  OTHERWISE IT RETURNS TO CALLER +3.  MAY BE CALLED IN RUPT OR UNDER EXEC.

MAGSUB          EXTEND
                BZMF    +2
                TCF     +2
                COM

                INDEX   Q
                AD      0
                EXTEND
                BZMF    Q+2             # ABS(A) <= CONST GO TO L+3
                TCF     Q+1             # ABS(A) > CONST GO TO L+2
```

```
# PROGRAM NAME: RRLIMCHK
#
# FUNCTIONAL DESCRIPTION:
#
#       RRLIMCHK CHECKS RR DESIRED GIMBAL ANGLES TO SEE IF THEY ARE WITHIN
#       THE LIMITS OF THE CURRENT MODE.  INITIALLY THE DESIRED TRUNNION AND
#       SHAFT ANGLES ARE STORED IN ITEMP1 AND ITEMP2.  THE CURRENT RR
#       ANTENNAE MODE (RADMODES BIT 12) IS CHECKED WHICH IS = 0 FOR
#       MODE 1 AND =1 FOR MODE 2.
#
#       MODE 1 -- THE TRUNNION ANGLE IS CHECKED AT MAGSUB TO SEE IF IT IS
#       BETWEEN -55 AND +55 DEGREES.  IF NOT, RETURN TO L +2.  IF WITHIN LIMITS,
#       THE SHAFT ANGLE IS CHECKED TO SEE IF IT IS BETWEEN -70 AND +59 DEGREES.
#       IF NOT, RETURN TO L +2.  IF IN LIMITS, RETURN TO L +3.
#
#       MODE 2 -- THE SHAFT ANGLE IS CHECKED AT MAGSUB TO SEE IF IT IS
#       BETWEEN -139 AND -25 DEGREES.  IF NOT, RETURN TO L +2.  IF WITHIN
#       LIMITS, THE TRUNNION ANGLE IS CHECKED TO SEE IF IT IS BETWEEN +125
#       AND -125 (+235) DEGREES.  IF NOT, RETURN TO L +2.  IF IN LIMITS, RETURN
#       TO L +3.
#
# CALLING SEQUENCE:
#
#       L   TC   RLIMCHK (WITH INTERRUPT INHIBITED)
#       L   +1   ADRES   T,S  (DESIRED TRUNNION ANGLE ADDRESS)
#
# ERASABLE INITIALIZATION REQUIRED:
#
#       RADMODES, MODEA, MODEB (OR DESIRED TRUNNION AND SHAFT
#       ANGLES ELSEWHERE IN CONSECUTIVE LOCATIONS -- UNSWITCHED ERASABLE OR
#       CURRENT EBANK).
#
# SUBROUTINES CALLED:   MAGSUB
#
# JOBS OR TASKS INITIATED:  NONE
#
# ALARMS:  NONE
#
# EXIT:         L + 2   (EITHER OR BOTH ANGLES NOT WITHIN LIMITS OF CURRENT MODE)
#               L + 3   (BOTH ANGLES WITHIN LIMITS OF CURRENT MODE)

RRLIMCHK        EXTEND
                INDEX   Q
                INDEX   0
                DCA     0
                INCR    Q
                DXCH    ITEMP1
                LXCH    Q               # L(CALLER +2) TO L.

                CAF     ANTENBIT        # SEE WHICH MODE RR IS IN.
                MASK    RADMODES
                CCS     A
                TCF     MODE2CHK

                CA      ITEMP1          # MODE 1 IS DEFINED AS
```

```
                TC      MAGSUB          #       1. ABS(T) L 55 DEGS.
                DEC     -.30555         #       2. ABS(S + 5.5 DEGS) L 64.5 DEGS
                TC      L               #               (SHAFT LIMITS AT +59, -70 DEGS)

                CAF     5.5DEGS
                AD      ITEMP2
                TC      MAGSUB
                DEC     -.35833         # 64.5 DEGS
                TC      L
                TC      RRLIMOK         # IN LIMITS.

MODE2CHK        CAF     82DEGS          # MODE 2 IS DEFINED AS
                AD      ITEMP2          #       1. ABS(T) G 125 DEGS.
                TC      MAGSUB          #       2. ABS(S + 82 DEGS) L 57 DEGS
                DEC     -.31667         #               (SHAFT LIMITS AT -25, -139 DEGS)
                TC      L

                CA      ITEMP1
                TC      MAGSUB
                DEC     -.69444         # 125 DEGS

RRLIMOK         INDEX   L
                TC      L               # ( = TC 1 )

5.5DEGS         DEC     .03056
82DEGS          DEC     .45556
```

```
# PROGRAM NAME: SETTRKF
#
# FUNCTIONAL DESCRIPTION:
#
#       SETTRKF UPDATES THE TRACKER FAIL LAMP ON THE DSKY.
#       INITIALLY THE LAMP TEST FLAG (IMODES33 BIT 1) IS CHECKED.
#       IF A LAMP TEST IS IN PROGRESS, THE PROGRAM EXITS TO L +1.
#       IF NO LAMP TEST THE FOLLOWING IS CHECKED SEQUENTIALLY:
#               1) RR CDU'S BEING ZEROED, RR CDU OK, AND RR NOT IN
#                  AUTO MODE (RADMODES BITS 13, 7, 2).
#               2) LR VEL DATA FAIL AND NO LR POS DATA (RADMODES BITS
#                  8,5)
#               3) NO RR DATA (RADMODES BIT 4)
#       THE ABSENCE OF ALL THREE SIMULTANEOUSLY IN (1), THE PRESENCE OF BOTH
#       IN (2), AND THE PRESENCE OF (3) RESULTS IN EITHER THE TRACKER FAIL
#       LAMP (DSPTAB +11D BIT 8) BEING TURNED OFF OR IS LEFT OFF.  THEREFORE, THE
#       TRACKER FAIL LAMP IS TURN ON IF:
#               A) RR CDU FAILED WITH RR IN AUTO MODE AND RR CDU'S NOT BEING ZEROED
#               B) N SAMPLES OF LR DATA COULD NOT BE TAKEN IN 2N TRIES WITH
#                  EITHER THE ALT OR VEL INFORMATION
#               C) N SAMPLES OF RR DATA COULD NOT BE OBTAINED FROM 2N TRIES
#                  WITH EITHER THE AL
#
# CALLING SEQUENCE:
#
#       L       TC      SETTRKF
#
# ERASABLE INITIALIZATION REQUIRED:  IMODES33, RADMODES, DSPTAB +11D
#
# SUBROUTINES CALLED:  NONE
#
# JOBS OR TASKS INITIATED:  NONE
#
# ALARMS:  TRACKER FAIL LAMP
#
# EXIT:  L +1 (ALWAYS)

SETTRKF         CAF     BIT1            # NO ACTION IF DURING LAMP TEST
                MASK    IMODES33
                CCS     A
                TC      Q

RRTRKF          CA      BIT8
                TS      L

                CAF     13,7,2          # SEE IF CDU FAILED.
                MASK    RADMODES
                EXTEND
                BZF     TRKFLON         # CONDITION 3 ABOVE.

RRCHECK         CAF     RRDATABT        # SEE IF RR DATA FAILED.
                MASK    RADMODES
```

```
                CCS     A
TRKFLON         CA      L
                AD      DSPTAB +11D     # HALF ADD DESIRED AND PRESENT STATES.
                MASK    L
                EXTEND
                BZF     TCQ             # NO CHANGE.

FLIP            CA      DSPTAB +11D     # CAN'T USE LXCH DSPTAB +11D (RESTART PROB)
                EXTEND
                RXOR    LCHAN
                MASK    POSMAX
                AD      BIT15
                TS      DSPTAB +11D
                TC      Q

13,7,2          OCT     10102
ENDRMODF        EQUALS
```

```
# PROGRAM NAME:  RRTURNON
#
# FUNCTIONAL DESCRIPTION:
#
#       RRTURNON IS THE TURN-ON SEQUENCE WHICH, ALONG WTIH
#       RRZEROSB, ZEROES THE CDU'S AND DETERMINES THE RR MODE.
#       INITIALLY, CONTROL IS TRANSFERRED TO RRZEROSB FOR THE
#       ACTUAL TURN-ON SEQUENCE.  UPON RETURN THE PROGRAM
#       WAITS 1 SECOND BEFORE REMOVING THE TURN-ON FLAG
#       (RADMODES BIT1) SO THE REPOSITION ROUTINE WON'T
#       INITIATE PROGRAM ALARM 00501.  A CHECK IS THEN MADE
#       TO SEE IF A PROGRAM IS USING THE RR (STATE BIT 7).  IF
#       SO, THE PROGRAM EXITS TO ENDRADAR SO THAT THE RR CDU
#       FAIL FLAG (RADMODES BIT 7) CAN BE CHECKED BEFORE
#       RETURNING TO THE WAITING PROGRAM.  IF NOT, THE PROGRAM EXITS
#       TO TASKOVER.
#
# CALLING SEQUENCE:  WAITLIST TASK FROM RRAUTCHK IF THE RR POWER-ON AUTO
# BIT (CHAN 33 BIT 2) CHANGES TO 0 AND NO PROGRAM WAS USING
# THE RR (STATE BIT 7).
#
# ERASABLE INITIALIZATION REQUIRED:
#
#       RADMODES, STATE
#
# SUBROUTINES CALLED:  RRZEROSB, FIXDELAY, TASKOVER, ENDRADAR
#
# JOBS OR TASKS INITIATED:
#
#       NONE
#
# ALARMS:  NONE (SEE RRZEROSB)
#
# EXIT:  TASKOVER, ENDRADAR (WAITING PROGRAM)

                BANK    24
                SETLOC  P20S1
                BANK

                EBANK=  LOSCOUNT
                COUNT*  $$/RSUB
RRTURNON        TC      RRZEROSB
                TC      FIXDELAY        # WAIT 1 SEC BEFORE REMOVING TURN ON FLAG
                DEC     100             # SO A MONITOR REPOSITION WON'T ALARM.
                CS      TURNONBT
                MASK    RADMODES
                TS      RADMODES
                TCF     TASKOVER
```

```

# PROGRAM NAME:  RRZEROSB
#
# FUNCTIONAL DESCRIPTION:
#
#       RRZEROSB IS A CLOSED SUBROUTINE TO ZERO THE RR CDU'S,
#       DETERMINE THE RR MODE, AND TURN ON THE TRACKER FAIL
#       LAMP IF REQUIRED.  INITIALLY THE RR CDU ZERO BIT (CHAN 12
#       BIT 1) IS SET.  FOLLOWING A 20 MILLISECOND WAIT, THE LGC
#       RR CDU COUNTERS (OPTY, OPTX) ARE SET = 0 AFTER
#       WHICH THE RR CDU ZERO DISCRETE (CHAN 12 BIT 1) IS
#       REMOVED.  A 4 SECOND WAIT IS SET TO ALL THE RR CDU'S
#       TO REPEAT THE ACTUAL TRUNNION AND SHAFT ANGLES.  THE
#       RR CDU ZERO FLAG (RADMODES BIT 13) IS REMOVED.  THE
#       CONTENTS OF OPTY IS THEN CHECKED TO SEE IF THE TRUNNION
#       ANGLE IS LESS THAN 90 DEGREES.  IF NOT, BIT 12 OF
#       RADMODES IS SET = 1 TO INDICATE RR ANTENNA MODE 2.
#       IF LESS THAN 90 DEGREES, BIT 12 OF RADMODES IS SET = 0 TO
#       INDICATE RR ANTENNA MODE 1.  SETTRKF IS THEN CALLED TO
#       SEE IF THE TRACKER FAIL LAMP SHOULD BE TURNED ON.
#
# CALLING SEQUENCE:  L   TC   RRZEROSB   (FROM RRTURNON AND RRZERO)
#
# ERASABLE INITIALIZATION REQUIRED:
#
#       RADMODES (BIT 13 SET), DSPTAB +11D
#
# SUBROUTINES CALLED:  FIXDELAY, MAGSUB, SETTRKF
#
# JOBS OR TASKS INITIATED:
#
#       NONE
#
# ALARMS:  TRAKCER FAIL
#
# EXIT:  L +1 (ALWAYS)

RRZEROSB        EXTEND
                QXCH    RRRET
                CAF     BIT1            # BIT 13 OF RADMODES MUST BE SET BEFORE
                EXTEND                  # COMING HERE.
                WOR     CHAN12          # TURN ON ZERO RR CDU
                TC      FIXDELAY
                DEC     2

                CAF     ZERO
                TS      CDUT
                TS      CDUS
                CS      ONE             # REMOVE ZEROING BIT.
                EXTEND
                WAND    CHAN12
                TC      FIXDELAY
                DEC     1000            # RESET FAIL INHIBIT IN 10 SECS. -- D.281

                CS      RCDUOBIT        # REMOVE ZEROING IN PROCESS BIT
```

```
                MASK    RADMODES
                TS      RADMODES

                CA      CDUT
                TC      MAGSUB
                DEC     -.5
                TCF     +3              # IF MODE 2.

                CAF     ZERO
                TCF     +2
                CAF     ANTENBIT
                XCH     RADMODES
                MASK    -BIT12
                ADS     RADMODES

                TC      SETTRKF         # TRACKER LAMP MIGHT GO ON NOW.

                TC      RRRET           # DONE.

-BIT12          EQUALS  -1/8            # IN SPROOT
```

```
# PROGRAM NAME:  DORREPOS
#
# FUNCTIONAL DESCRIPTION:
#
#       DORREPOS IS A SEQUENCE OF TASKS TO DRIVE THE RENDEZVOUS RADAR
#       TO A SAFE POSITION.  INIITALLY SETRRECR IS CALLED WHERE THE RR
#       ERROR COUNTERS (CHAN 12 BIT 2) ARE ENABLED AND LASTYCMD
#       AND LASTXCMD SET = 0 TO INDICATE THE DIFFERENCE BETWEEN THE
#       DESIRED STATE AND PRESENT STATE OF THE COMMANDS.  THE RR
#       TURN-ON FLAG (RADMODES BIT 1) IS CHECKED AND IF NOT PRESENT,
#       PROGRAM ALARM 00501 IS REQUESTED BEFORE CONTINUING.  IN EITHER
#       CASE, FOLLOWING A 20 MILLISECOND WAIT THE PROGRAM CHECKS THE CURRENT
#       RR ANTENNA MODE (RADMODES BIT 12).  RRTONLY IS THEN CALLED
#       TO DRIVE THE TRUNNION ANGLE TO 0 DEGREES IF IN MODE 1 AND TO 180
#       DEGREES IF IN MODE 2.  UPON RETURN, THE CURRENT RR ANTENNA
#       MODE (RADMODES BIT 12) IS AGAIN CHECKED.  RRSONLY IS THEN
#       CALLED TO DRIVE THE SHAFT ANGLE TO 0 DEGREES IF IN MODE 1 AND TO
#       -90 DEGREES IF IN MODE 2.  IF DURING RRTONLY OR RRSONLY A
#       REMODE HAS BEEN REQUESTED (RADMODES BIT 14), AND ALWAYS
#       FOLLOWING COMPLETION OF RRSONLY, CONTROL IS TRANFERRED TO
#       REPOSRPT.  HERE THE REPOSITION FLAG (RADMODES BIT 11) IS
#       REMOVED.  A CHECK IS THEN MADE ON THE DESIGNATE FLAG (RADMODES
#       BIT 10).  IF PRESENT, CONTROL IS TRANSFERRED TO BEGDES.  IF NOT PRESENT
#       INDICATING NO FURTHER ANTENNA CONTROL REQUIRED, THE RR ERROR
#       COUNTER BIT (CHAN 12 BIT 2) IS REMOVED AND THE ROUTINE EXITS TO
#       TASKOVER.
#
# CALLING SEQUENCE:
#
#       WAITLIST CALL FROM RRGIMON IF TRUNNION AND SHAFT CDU ANGLES
#       NOT WITHIN LIMITS OF CURRENT MODE.
#
# ERASABLE INITIALIZATION REQUIRED:
#
#       RADMODES
#
# SUBROUTINES CALLED
#
#       RRTONLY, RRSONLY, BEGDES (EXIT)
#
# JOBS OR TASKS INITIATED:
#
#       NONE
#
# ALARMS:  NONE
#
# EXIT:  TASKOVER, BEGDES

DORREPOS        TC      SETRRECR        # SET UP RR CDU ERROR COUNTERS.

# ALARM 501 DELETED IN DANCE 279 PER PCR 97.

                TC      FIXDELAY
                DEC     2

                CAF     ANTENBIT        # MANEUVER TRUNNION ANGLE TO NOMINAL POS.
```

```
                MASK RADMODES
                CCS     A
                CAF     BIT15           # 0 FOR MODE 1 AND 180 FOR MODE 2.
                TC      RRTONLY

                CAF     ANTENBIT        # NOT PUT SHAFT IN RIGHT POSITION
                MASK    RADMODES
                CCS     A
                CS      HALF            # -90 FOR MODE 2.
                TC      RRSONLY

REPOSRPT        CS      REPOSBIT        # RETURNS HERE FROM RR1AXIS IN REMODE
                                        # REQUESTED DURING REPOSITION.
                MASK    RADMODES        # REMOVE REPOSITION BIT.
                TS      RADMODES
                MASK    DESIGBIT        # SEE IF SOMEONE IS WAITING TO DESIGNATE.
                CCS     A
                TCF     BEGDES
                CS      BIT2            # IF NO FURTHER ANTENNA CONTROL REQUIRED,
                EXTEND                  # REMOVE ERROR COUNTER ENABLE.
                WAND    CHAN12
                TCF     TASKOVER

SETRRECR        CAF     BIT2            # SET UP RR ERROR COUNTERS
                EXTEND
                RAND    CHAN12
                CCS     A               # DO NOT CLEAR LAST COMMAND IF
                TC      Q               # ERROR COUNTERS ARE ENABLED

                TS      LASTYCMD
                TS      LASTXCMD
                CAF     BIT2
                EXTEND
                WOR     CHAN12          # ENABLE RR CDU ERROR COUNTERS.
                TC      Q
```

```
# PROGRAM NAME:  REMODE
#
# FUNCTIONAL DESCRIPTION
#
#       REMODE IS THE GENERAL REMODING SUBROUTINE.  IT DRIVES THE
#       TRUNNION ANGLE TO 0 DEGREES IF THE CURRENT MODE IS MODE 1,
#       180 DEGREES FOR MODE 2, THEN DRIVES THE SHAFT ANGLE TO -45
#       DEGREES, AND FINALLY DRIVES THE TRUNNION ANGLE TO -130 DEGREES,
#       TO PLACE THE RR IN MODE 2, -50 DEGREES FOR MODE 1, BEFORE
#       INITIATING 2-AXIS CONTROL.  ALL REMODING IS DONE WITH SINGLE
#       AXIS ROTATIONS (RR1AXIS).  INITIALLY THE RR ANTENNA MODE FLAG
#       (RADMODES BIT 12) IS CHECKED.  CONTROL IS THEN TRANSFERRED TO
#       RRTONLY TO DRIVE THR TRUNNION ANGLE TO 0 DEGREES IF IN MODE 1
#       OR 180 DEGREES IF IN MODE 2.  RRSONLY IS THEN CALLED TO DRIVE
#       THE SHAFT ANGLE TO -45 DEGREES. THE RR ANTENNA MODE FLAG
#       (RADMODES BIT 12) IS CHECKED AGAIN.  CONTROL IS AGAIN
#       TRANSFERRED TO RRTONLY TO DRIVE THE TRUNNION ANGLE TO -130
#       DEGREES TO PLACE THE RR IN MODE 2 IF CURRENTLY IN MODE 1 OR TO
#       -50 DEGREES IF IN MODE 2 TO PLACE THE RR IN MODE 1.  RMODINV
#       IS THEN CALLED TO SET RADMODES BIT 12 TO INDICATE THE NEW
#       RR ANTENNA MODE.  THE REMODE FLAG (RADMODES BIT 14)
#       IS REMOVED TO INDICATE THAT REMODING IS COMPLETE.  THE PROGRAM
#       THEN EXITS TO STDESIG TO BEGIN 2-AXIS CONTROL.
#
# CALLIN SEQUENCE:
#
#       FROM BEGDES WHEN REMODE FLAG (RADMODES BIT 14) IS SET.
#       THIS FLAG MAY BE SET IN RRDESSM AND RRDESNB IF RRLIMCHK
#       DETERMINES THAT THE DESIRED ANGLES ARE WITHIN THE LIMITS OF THE
#       OTHER MODE.
#
# ERASABLE INIITIALIZATION REQUIRED:
#
#       RADMODES
#
# SUBROUTINES CALLED:
#
#       RRTONLY, RRSONL, RMODINV (ACTUALLY PART OF)
#
# JOBS OR TASKS INITIATED:
#
#       NONE
#
# ALARMS:  NONE
#
# EXIT:  STDESIG

REMODE          CAF     ANTENBIT        # DRIVE TRUNNION TO 0 (180)
                MASK    RADMODES        # (ERROR COUNTER ALREADY ENABLED)
                CCS     A
                CAF     BIT15
                TC      RRTONLY

                CAF     -45DEGSR
                TC      RRSONLY
```

```
                CS      RADMODES
                MASK    ANTENBIT
                CCS     A
                CAF     -80DEGSR        # GO TO T = -130 (-50).
                AD      -50DEGSR
                TC      RRTONLY

                CS      RADMODES
                MASK    ANTENBIT
                CCS     A
                CAF     BIT15           # GO TO T = -180 (+0).
                TC      RRTONLY

                CS      RADMODES        # GO TO S = -90 (+0).
                MASK    ANTENBIT
                CCS     A
                CS      HALF
                TC      RRSONLY

                TC      RMODINV

                CS      REMODBIT        # END OF REMODE.
                MASK    RADMODES
                TS      RADMODES

                CAF     DESIGBIT        # WAS REMODE CALLED DURING DESIGNATE?
                MASK    RADMODES        # (BIT10 RADMODES = 1)
                EXTEND
                BZF     RGOODEND        # NO -- RETURN TO CALLER WAITING IN RADSTALL
                TC      STDESIG         # YES -- RETURN TO DESIGNATE
-45DEGSR        =       13,14,15
-50DEGSR        DEC     -.27778
-80DEGSR        DEC     -.44444

RMODINV         LXCH    RADMODES        # INVERT THE MODE STATUS.
                CAF     ANTENBIT
                EXTEND
                RXOR    LCHAN
                TS      RADMODES
                TC      Q
```

```
# PROGRAM NAMES:        RRTONLY, RRSONLY
#
# FUNCTIONAL DESCRIPTION:
#
#       RRTONLY AND RRSONLY ARE SUBROUTINES FOR DOING SINGLE AXIS
#       RR MANEUVERS FOR REMODE AND REPOSITION.  IT DRIVES TO
#       WITHIN 1 DEGREE.  INITIALLY, AT RR1AX2, THE REMODE AND REPOSITION
#       FLAGS (RADMODES BITS 14, 11) ARE CHECKED.  IF BOTH EXIST,
#       THE PROGRAM EXITS TO REPOSRPT (SEE DORREPOS).  THIS INDICATES
#       THAT SOMEONE POSSIBLY REQUESTED A DESIGNATE (RADMODES BIT 10)
#       WHICH REQUIRES A REMODE (RADMODES BIT 14) AND THAT A
#       REPOSITION IS IN PROGRESS (RADMODES BIT 11).  IF NONE
#       OR ONLY ONE OF THE FLAGS EXIST, REMODE OR REPOSITION, MAGSUB
#       IS CALLED TO SEE IF THE APPROPRIATE ANGLE IS WITHIN 1 DEGREE.  IF YES,
#       CONTROL RETURNS TO THE CALLING ROUTINE.  IF NOT, CONTROL IS
#       TRANSFERRED TO RROUT FOR SINGLE AXIS MANEUVERS WITH THE OTHER
#       ANGLE SET = 0.  FOLLOWING A .5 SECOND WAIT, THE ABOVE PROCEDURE IS
#       REPEATED.
#
# CALLING SEQUENCE:     L-1     CAF     *ANGLE*         (DESIRED ANGLE SCALED PI)
#                       L       TC      RRTONLY         (TRUNNION ONLY)
#                       RRSONLY                         (SHAFT ONLY)
#                       RRTONLY IS CALLED BY PREPOS29;
#                       RRTONLY AND RRSONLY ARE CALLED BY DORREPOS AND REMODE
#
# ERASABLE INITIALIZATION REQUIRED:
#
#       C(A) = DESIRED ANGLE, RADMODES
#
# SUBROUTINES CALLED:
#
#       FIXDELAY, REPOSRPT, MAGSUB, RROUT
#
# JOBS OR TASKS INITIATED:
#
#       NONE
#
# ALARMS:  NONE
#
# EXIT: REPOSRPT (REMODE AND REPOSITION FLAGS PRESENT -- RADMODES
#       BITS 14, 11)
#       L+1 (ANGLE WITHIN ONE DEGREE OR RR OUT OF AUTO MODE)

RRTONLY         TS      RDES            # DESIRED TRUNNION ANGLE.
                CAF     ZERO
                TCF     RR1AXIS

RRSONLY         TS      RDES            # SHAFT COMMANDS ARE UNRESOLVED SINCE THIS
                CAF     ONE             # ROUTINE ENTERED ONLY WHEN T = 0 OR 180.

RR1AXIS         TS      RRINDEX
                EXTEND
                QXCH    RRRET
                TCF     RR1AX2
```

```
NXTRR1AX        TC      FIXDELAY
                DEC     50              # 2 SAMPLES PER SECOND.

RR1AX2          CS      RADMODES        # IF SOMEONE REQUESTS A DESIGNATE WHICH
                MASK    PRIO22          # REQUIRES A REMODE AND A REPOSITION IS IN
                EXTEND                  # PROGRESS, INTERRUPT IT AND START THE
                BZF     REPOSRPT        # REMODE IMMEDIATELY.

                CA      RDES
                EXTEND
                INDEX   RRINDEX
                MSU     CDUT
                TS      ITEMP1          # SAVE ERROR SIGNAL.
                EXTEND
                MP      RRSPGAIN        # TRIES TO NULL .7 OF ERROR OVER NEXT .5
                TS      L
                CA      RADMODES
                MASK    AUTOMBIT
                XCH     ITEMP1          # STORE RR-OUT-OF-AUTO-MODE BIT.
                TC      MAGSUB          # SEE IF WITHIN ONE DEGREE.
                DEC     -.00555         # SCALED IN HALF-REVS.

                CCS     ITEMP1          # NO.  IF RR OUT OF AUTO MODE, EXIT.
                TC      RRRET           # RETURN TO CALLER.

                CCS     RRINDEX         # COMMAND FOR OTHER AXIS IS ZERO.
                TCF     +2              # SETTING A TO 0.
                XCH     L
                DXCH    TANG

                TC      RROUT

                TCF     NXTRR1AX        # COME BACK IN .5 SECONDS.

RRSPGAIN        DEC     .59062          # NULL .7 ERROR IN .5 SEC.
```

```
# PROGRAM NAME:  RROUT
#
# FUNCTIONAL DESCRIPTION:
#
#       RROUT RECEIVES RR GYRO COMMANDS IN TANG, TANG +1 IN RR
#       ERROR COUNTER SCALING.  RROUT THEN LIMITS THEM AND
#       GENERATES COMMANDS TO THE CDU TO ADJUST THE ERROR COUNTERS
#       TO THE DESIRED VALUES.  INITIALLY MAGSUB CHECKS THE MAGNITUDE OF
#       THE COMMAND (SHAFT ON 1ST PASS) TO SEE IF IT IS GREATER THAN
#       384 PULSES.  IF NOT, CONTROL IS TRANFERRED TO RROUTLIM TO
#       LIMIT THE COMMAND TO +384 OR -384 PULSES.  THE DIFFERENCE IS
#       THEN CALCULATED BETWEEN THE DESIRED STATE AND TEH PRESENT STATE OF
#       THE ERROR COUNTER AS RECORDED IN LASTYCMD AND LASTXCMD.
#       THE RESULT IS STORED IN OPTXCMD (1ST PASS) AND OPTYCMD (2ND
#       PASS).  FOLLOWING THE SECOND PASS, FOR THE TRUNNION COMMAND, THE
#       OCDUT AND OCDUS ERROR COUNTER DRIVE BITS (CHAN 14 BITS 12, 11)
#       ARE SET.  THIS PROGRAM THEN EXITS TO THE CALLING PROGRAM.
#
# CALLING SEQUENCE:
#
#       L TC RROUT (WITH RUPT INHIBITED) RROUT IS CALLED BY
#       RRTONLY, RRSONLY, AND DODES
#
# ERASABLE INITIALIZATION REQURIED:
#
#       TANG, TANG +1 (DESIRED COMMANDS), LASTYCMD, LASTXCMD
#       (1ST PASS = 0), RR ERROR COUNTER ENAGLE SET (CHAN 12 BIT 2).
#
# SUBROUTINES CALLED:
#
#       MAGSUB
#
# JOBS OR TASKS INITIATED:
#
#       NONE
#
# ALARMS:  NONE
#
# EXIT:  L+1 (ALWAYS)

RROUT           LXCH    Q               # SAVE RETURN
                CAF     ONE             # LOOP TWICE.
RROUT2          TS      ITEMP2
                INDEX   A
                CA      TANG
                TS      ITEMP1          # SAVE SIGN COMMAND FOR LIMITING.

                TC      MAGSUB          # SEE IF WITHIN LIMITS.
-RRLIMIT        DEC     -384
                TCF     RROUTLIM        # LIMIT COMMAND TO MAG OF 384.

SETRRCTR        CA      ITEMP1          # COUNT OUT DIFFERENCE BETWEEN DESIRED
                INDEX   ITEMP2          # STATE AND PRESENT STATE AS RECORDED IN
                XCH     LASTYCMD        # LASTYCMD AND LASTXCMD
                COM
```

```
                AD      ITEMP1
                AD      NEGO            # PREVENT +0 IN OUTCOUNTER
                INDEX   ITEMP2
                TS      CDUTCMD

                CCS     ITEMP2          # PROCESS BOTH INPUTS.
                TCF     RROUT2

                CAF     PRIO6           # ENABLE COUNTERS.
                EXTEND
                WOR     CHAN14          # PUT ON CDU DRIVES S AND T
                TC      L               # RETURN.

RROUTLIM        CCS     ITEMP1          # LIMIT COMMAND TO ABS VAL OF 384.
                CS      -RRLIMIT
                TCF     +2
                CA      -RRLIMIT
                TS      ITEMP1
                TCF     SETRRCTR +1
```

```
# ROUTINE TO ZERO THE RR CDUS AND DETERMINE THE ANTENNA MODE.

RRZERO          CAF     BIT11+1         # SEE IF MONITOR REPOSITION OR NOT IN AUTO
                MASK    RADMODES        # IF SO, DON'T RE-ZERO CDUS.
                CCS     A
                TCF     RADNOOP         # (IMMEDIATE TASK TO RGOODEND).

                INHINT
                CS      RCDUOBIT        # SET FLAG TO SHOW ZEROING IN PROGRESS.
                MASK    RADMODES
                AD      RCDUOBIT
                TS      RADMODES

                CAF     ONE
                TC      WAITLIST
                EBANK=  LOSCOUNT
                2CADR   RRZ2

                CS      RADMODES        # SEE IF IN AUTO MODE.
                MASK    AUTOMBIT
                CCS     A
                TCF     ROADBACK
                TC      ALARM           # AUTO DISCRETE NOT PRESENT -- TRYING
                OCT     510
ROADBACK        RELINT
                TCF     SWRETURN

RRZ2            TC      RRZEROSB        # COMMON TO TURNON AND RRZERO.
                TCF     ENDRADAR

BIT11+1         OCT     02001
```

```
# PROGRAM NAME:  RRDESSM
#
# FUNCTIONAL DESCRIPTION:
#
#       THIS INTERPRETIVE ROUTINE WILL DESIGNATE, IF DESIRED ANGLES ARE
#       WITHIN THE LIMITS OF EITHER MODE, TO A LINE-OF-SIGHT (LOS) VECTOR
#       (HALF-UNIT) KNOWN WITH RESPECT TO THE STABLE MEMBER PRESENT
#       ORIENTATION.  INITIALLY THE IMU CDU'S ARE READ AND CONTROL
#       TRANSFERRED TO SMNB TO TRANSFORM THE LOS VECTOR FROM STABLE
#       MEMBER TO NAVIGATION BASE COORDINATES (SEE STG MEMO 699)
#       RRANGLES IS THEN CALLED TO CALCULATE THE RR GIMBAL ANGLES,
#       TRUNNION AND SHAFT, FOR BOT THE PRESENT AND ALTERNATE MODE.
#       RRLIMCHK IS CALLED TO SEE IF THE ANGLES CALCULATED FOR THE
#       PRESENT MODE ARE WITHIN LIMITS.  IF WITHIN LIMITS, THE RETUREN
#       LOCATION IS INCREMENTED, INASMUCH AS NO VEHICLE MANEUVER IS
#       REQUIRED, BEFORE EXITING TO STARTDES.  IF NOT WITHIN THE LIMITS OF THE
#       CURRENT MODE, TRYSWS IS CALLED.  FOLLOWING INVERTING OF THE RR
#       ANTENNA MODE FLAG (RADMODES BIT 12), RRLIMCHK IS CALLED
#       TO SEE IF THE ANGLES CALCULATED FOR THE ALTERNATE MODE ARE WITHIN
#       LIMITS.  IF YES, THE RR ANTENNA MODE FLAG IS AGAIN INVERTED,
#       THE REMODE FLAG (RADMODES BIT 14) SET, AND THE RETURN LOCATION
#       INCREMENTED, TO INDICATE NO VEHICLE MANEUVER IS REQUIRED, BEFORE
#       EXITING TO STARTDES.  IF THESE ANGLES ARE NOT WITHIN LIMITS
#       OF THE ALTERNATE MODE, THE RR ANTENNA MODE FLAG (RADMODES
#       BIT 12) IS INVERTED BEFORE RETURNING DIRECTLY TO THE CALLING PROGRAM
#       TO INDICATE THAT A VEHICLE MANEUVER IS REQUIRED.
#
# CALLING SEQUENCE:
#
#       L       STCALL  RRTARGET        (LOS HALF-UNIT VECTOR IN SM COORDINATES)
#       L+1     RRDESM
#       L+2     BASIC                   (VEHICLE MANEUVER REQUIRED)
#       L+3     BASIC                   (NO VEHICLE MANEUVER REQUIRED)
#
# ERASABLE INITIALIZATION REQUIRED:
#
#       RRTARGET, RADMODES
#
# SUBROUTINES CALLED:
#
#       READCDUS, SMNB, RRANGLES, RRLIMCHK, TRYSWS (ACTUALLY
#       PART OF), RMODINV
#
# JOBS OR TASKS INITIATED:
#
#       NONE
#
# ALARMS:  NONE
#
# EXIT:  L+2 (NEITHER SET OF ANGLES ARE WITHIN LIMITS OF RELATED MODE)
# STARTDES (DESIGNATE POSSIBLE AT PRESENT VEHICLES ATTITUDE -- RETURNS
# TO L+3 FROM STARTDES)

RRDESSM         STQ     CLEAR
                        DESRET
```

```
                                RRNBSW
                        CALL                    # COMPUTES SINES AND COSINES, ORDER Y Z X
                                CDUTRIG
                        VLOAD   CALL            # LOAD VECTOR AND CALL TRANSFORMATION
                                RRTARGET
                                *SMNB*

                        CALL                    # GET RR GIMBAL ANGLES IN PRESENT AND
                                RRANGLES        # ALTERNATE MODE.
                        EXIT

                        INHINT
                        TC      RRLIMCHK
                        ADRES   MODEA           # CONFIGURATION FOR CURRENT MODE.
                        TC      +3              # NOT IN CURRENT MODE
        OKDESSM         INCR    DESRET          # INCREMENT SAYS NO VEHICLE MANEUVER REQ.
                        TC      STARTDES        # SHOW DESIGNATE REQUIRED
                        CS      FLAGWRD8
                        MASK    SURFFBIT        # CHECK IF ON LUNAR SURFACE (SURFFLAG=P22F)
                        EXTEND
                        BZF     NORDSTAL        # BRANCH -- YES -- CANNOT DESIGNATE IN MODE 2
                        TC      TRYSWS

        LUNDESCH        CS      FLAGWRD8        # OVERFLOW RETURN FROM RRANGLES
                        MASK    SURFFBIT        # CHECK IF ON LUNAR SURFACE
                        EXTEND
                        BZF     NORDSTAL        # BRANCH -- YES -- RETURN TO CALLER -- ALARM 527
                        CA      STATE
                        MASK    RNDVZBIT
                        CCS     A               # TEST RNDVZFLG
                        TC      NODESSM         # NOT ON MOON -- CALL FOR ATTITUDE MANEUVER
                        TCF     ENDOFJOB        # ... BUT NOT IN R29.
```

```
# PROGRAM NAME:  STARTDES
#
# FUNCTIONAL DESCRIPTION:
#
#       STARTDES IS ENTERED WHEN WE ARE READY TO BEGIN DESIGNATION.
#       BIT 14 OF RADMODES IS ALREADY SET IF A REMODE IS REQUIRED.
#       AT THIS TIME, THE RR ANTENNA MAY BE IN A REPOSITON
#       OPERATION.  IN THIS CASE, IF A REMODE IS REQUIRED IT MAY HAVE
#       ALREADY BEGUN BUT IN ANY CASE THE REPOSITION WILL BE INTERRUPTED.
#       OTHERWISE, THE REPOSITION WILL BE COMPLETED BEFORE 2-AXIS
#       DESIGNATION BEGINS.  INITIALLY DESCOUNT IS SET = 60 TO INDICATE
#       THAT 30 SECONDS WILL BE ALLOWED FOR THE RR DATA GOOD INBIT
#       (CHAN 33 BIT 4) IF LOCK-ON IS DESIRED (STATE BIT 5).  BIT 10
#       OF RADMODES IS SET TO SHOW THAT A DESIGNATE IS REQUIRED.
#       THE REPOSITON FLAG (RADMODES BIT 11) IS CHECKED.  IF SET,
#       THE PROGRAM EXITS TO L+3 OF THE CALLING PROGRAM (SEE RRDESSM
#       AND RRDESNB).  THE PROGRAM WILL BEGIN DESIGNATING TO THE DESIRED
#       ANGLES FOLLOWING THE REPOSITON OR REMODE IF ONE WAS
#       REQUESTED.  IF THE REPOSITON FLAG IS NOT SET, SETRRECR IS CALLED
#       WITH SETS THE RR ERROR COUNTER ENABLE BIT (CHAN 12 BIT 2)
#       AND SETS LASTYCMD AND LASTXCMD = 0 TO INDICATE THE
#       DIFFERENCE BETWEEN THE PRESENT AND DESIRED STATE OF THE ERROR
#       COUNTERS.  A 20 MILLISECOND WAITLIST CALL IS SET FOR BEGDES
#       AFTER WHICH THE PROGRAM EXITS TO L+3 OF TEH CALLING PROGRAM.
#
# CALLING SEQUENCE:
#
#       FROM RRDESSM AND RRDESNB WHEN ANGLES WITHIN LIMITS.
#
# ERASABLE INITIALIZATION REQUIRED:
#
#       RADMODES, (SEE DODES)
#
# SUBROUTINES CALLED
#
#       SETRRECR, WAITLIST
#
# JOBS OR TASKS INITIATED:
#
#       BEGDES
#
# ALARMS:  NONE
#
# EXIT: L+3 OF CALLING PROGRAM (SEE RRDESSM)
#       L+2 OF CALLING PROGRAM (SEE RRDESNB)

STARTDES        INCR    DESRET
                CS      RADMODES
                MASK    DESIGBIT
                ADS     RADMODES
                MASK    REPOSBIT        # SEE IF REPOSITIONING IN PROGRESS.
                CCS     A
                TCF     DESRETRN        # ECTR ALREADY SET UP.

                TC      SETRRECR        # SET UP ERROR COUNTERS.
```

```
                  CAF      TWO
                  TC       WAITLIST
                  EBANK=   LOSCOUNT
                  2CADR    BEGDES

DESRETRN          CA       RADCADR          # FIRST PASS THRU DESIGNATE
                  EXTEND
                  BZF      DESRTRN          # YES   SET EXIT
                  TC       ENDOFJOB         # NO
DESRTRN           RELINT
                  INCR     DESRET
                  CA       DESRET
                  TCF      BANKJUMP

NORDSTAL          CAF      ZERO             # ZERO RADCADR TO WIPE OUT ANYONE
                  TS       RADCADR          # WAITING IN RADSTALL SINCE WE ARE NOW
                  TCF      DESRTRN          # RETURNING TO P20 AND MAY DO NEW RADSTALL
```

```
# SEE IF RRDESSM CAN BE ACCOMPLISHED AFTER A REMODE.

TRYSWS          TC      RMODINV         # (NOTE RUPT INHIBIT)
                TC      RRLIMCHK        # TRY DIFFERENT MODE.
                ADRES   MODEB
                TCF     NODESSM         # VEHICLE MANEUVER REQUIRED

                TC      RMODINV         # RESET BIT12
                CAF     REMODBIT        # SET FLAG FOR REMODE.
                ADS     RADMODES

                TCF     OKDESSM

NODESSM         TC      RMODINV         # RE-INVERT MODE AND RETURN
                INCR    DESRET          # TO CALLER +2
                TCF     NORDSTAL

MAXTRYS         DEC     60
```

```
# DESIGNATE TO SPECIFIC RR GIMBAL ANGLES (INDEPENDENT OF VEHICLE MOTION).  ENTER WITH DESIRED ANGLES IN
# TANG AND TANG +1.

RRDESNB         TC      MAKECADR
                TS      DESRET

                TC      DOWNFLAG        # RESET FLAG TO PREVENT DODES FROM GOING
                ADRES   LOSCMFLG        # BACK TO R21
                CA      MAXTRYS         # SET TIME LIMIT COUNTER
                TS      DESCOUNT        # FOR DESIGNATE
                INHINT                  # SEE IF CURRENT MODE OK.
                TC      RRLIMNB         # DO SPECIAL V41 LIMIT CHECK
                ADRES   TANG
                TCF     TRYSWN          # SEE IF IN OTHER MODE.

OKDESNB         RELINT
                EXTEND
                DCA     TANG
                DXCH    TANGNB
                TC      INTPRET

                CALL                    # GET LOS IN NB COORDS.
                        RRNB
                STORE   RRTARGET

                SET     EXIT
                        RRNBSW

                INHINT
                TCF     STARTDES +1
TRYSWN          TC      RMODINV         # SEE IF OTHER MODE WILL DO.
                TC      RRLIMNB         # DO SPECIAL V41 LIMIT CHECK
                ADRES   TANG
                TCF     NODESNB         # NOT POSSIBLE.

                TC      RMODINV
                CAF     REMODBIT        # CALL FOR REMODE.
                ADS     RADMODES
                TCF     OKDESNB

NODESNB         TC      RMODINV         # REINVERT MODE BIT.
                TC      ALARM           # BAD INPUT ANGLES.
                OCT     502
                TC      CLRADMOD
                TC      ENDOFJOB        # AVOID 503 ALARM.

RRLIMNB         INDEX   Q               # THIS ROUTINE IS IDENTICAL TO RRLIMCHK
                CAF     0               # EXCEPT THAT THE MODE 1 SHAFT LOWER
                INCR    Q               # LIMIT IS -85 INSTEAD OF -70 DEGREES
                EXTEND
```

```
                INDEX   A               # READ GIMBAL ANGLES INTO ITEMP STORAGE
                DCA     0
                DXCH    ITEMP1
                LXCH    Q               # L(CALLER +2) TO L

                CAF     ANTENBIT        # SEE WHICH MODE RR IS IN
                MASK    RADMODES
                CCS     A
                TCF     MODE2CHK        # MODE 2 CAN USE RRLIMCHK CODING
                CA      ITEMP1
                TC      MAGSUB          # MODE 1 IS DEFINED AS
                DEC     -.30555         # 1     ABS(T) L 55 DEGS
                TC      L               # 2     SHAFT LIMITS AT +59, -85 DEGS

                CA      ITEMP2          # LOAD SHAFT ANGLE
                EXTEND
                BZMF    NEGSHAFT        # IF NEGATIVE SHAFT ANGLE, ADD 20.5 DEGS
                AD      5.5DEGS
SHAFTLIM        TC      MAGSUB
                DEC     -.35833         # 64.5 DEGREES
                TC      L               # NOT IN LIMITS
                TC      RRLIMOK         # IN LIMITS
NEGSHAFT        AD      20.5DEGS        # MAKE NEGATIVE SHAFT LIMIT -85 DEGREES
                TCF     SHAFTLIM

20.5DEGS        DEC     .11389
```

```
# PROGRAM NAME:  BEGDES
#
# FUNCTIONAL DESCRIPTION:
#
#       BEGDES CHECKS VARIOUS DESIGNATE REQUESTS AND REQUESTS THE
#       ACTUAL RR DESIGNATION.  INITIALLY A CHECK IS MADE TO SEE IF A
#       REMODE (RADMODES BIT 14) IS REQUESTED OR IN PROGRESS.  IF SO,
#       CONTROL IS TRANFERRED TO STDESIG AFTER ROUTINE REMODE IS
#       EXECUTED.  IF NO REMODE, STDESIG IS IMMEDIATELY CALLED WHERE
#       FIRST THE REPOSITION FLAG (RADMODES BIT 11) IS CHECKED.  IF
#       PRESENT, THE DESIGNATE FLAG (RADMODES BIT 10) IS REMOVED
#       AFTER WHICH THE PROGRAM EXITS TO RDBADEND.  IF THE REPOSITION
#       FLAG IS NOT PRESET, THE CONTINUOUS DESIGNATE FLAG (RADMODES
#       BIT 15) IS CHECKED.  IF PRESENT, AN EXECUTIVE CALL IS IMMEDIATELY
#       MADE FOR DODES AFTER WHICH A .5 SECOND WAIT IS INITIATED BEFORE
#       REPEATING AT STDESIG.  IF THE RR SEARCH ROUTINE (LRS24.1) IS DESIGNATING
#       TO A NEW POINT (NEWPTFLG SET) THE CURRENT DESIGNATE TASK IS TERMINATED.
#       IF CONTINUOUS DESIGNATE IS NOT WANTED, THE DESIGNATE FLAG (RADMODES
#       BIT 10) IS CHECKED.  IF NOT PRESENT, THE PROGRAM EXITS TO ENDRADAR TO
#       CHECK RR CDU FAIL BEFORE RETURNING TO THE CALLING PROGRAM.  IF DESIGNATE
#       IS STILL REQUIRED, DESCOUNT IS CHECKED TO SEE IF THE 30 SECONDS HAS
#       EXPIRED BEFORE RECEIVING THE RR DATA GOOD (CHAN 33 BIT 4)
#       SIGNAL.  IF OUT OF TIME, PROGRAM ALARM 00503 IS REQUESTED, THE
#       RR AUTO TRACKER ENABLE AND RR ERROR COUNTER ENABLE
#       (CHAN 12 BITS 14,2) BITS REMOVED, AND THE DESIGNATE FLAG
#       (RADMODES BIT 10) REMOVED BEFORE EEXITING TO RDBADEND.  IF
#       TIME HAS NOT EXPIRED, DESCOUNT IS DECREMENTED, THE
#       EXECUTIVE CALL MADE FOR DODES, AND A .5 SECOND WAIT INITIATED
#       BEFORE REPEATING THIS PROCEDURE AT STDESIG.
#
# CALLING SEQUENCE:
#
#       WAITLIST CALL FROM STARTDES
#       TCF BEGDES FROM DORREPOS
#       TC STDESIG RETURNING, FROM REMODE
#
# ERASABLE INITIALIZATION REQUIRED:
#
#       DESCOUNT, FINDVAC
#
# JOBS OR TASKS INITIATED:  DODES
#
# ALARMS:  PROGRAM ALARM 00503 (30 SECONDS HAVE EXPIRED) WITH NO RR DATA
# GOOD (CHAN 33 BIT 4) RECEIVED WHEN LOCK-ON (STATE BIT 5) WAS REQUESTED.
#
# EXIT:         TASKOVER (SEARCH PATTERN DESIGNATING TO NEW POINT)
#               ENDRADAR (NO DESIGNATE -- RADMODES BIT 10)
#               RDBADEND (REPOSITION OR 30 SECONDS EXPIRED)

BEGDES          CS      RADMODES
```

```
                MASK    REMODBIT
                CCS     A
                TC      STDESIG
                TC      REMODE
DESLOOP         TC      FIXDELAY        # 2 SAMPLES PER SECOND.
                DEC     50

STDESIG         CAF     REPOSBIT
                MASK    RADMODES        # SEE IF GIMBAL LIMIT MONITOR HAS FOUND US
                CCS     A               # OUT OF BOUNDS.  IF SO, THIS BIT SHOWS A
                TCF     BADDES          # REPOSITION TO BE IN PROGRESS.

                CCS     RADMODES        # SEE IF CONTINUOUS DESIGNATE WANTED.
                TCF     +3              # IF SO, DON'T CHECK BIT 10 TO SEE IF IN
                TCF     +2              # LIMITS BUT GO RIGHT TO FINDVAC ENTRY.
                TCF     MOREDES +1

                CS      RADMODES        # IF NON-CONTINUOUS, SEE IF END OF
                MASK    DESIGBIT        # PROBLEM (DATA GOOD IF LOCK-ON WANTED OR
                CCS     A               # WITHIN LIMITS IF NOT).  IF SO, EXIT AFTER
                TCF     ENDRADAR        # CHECKING RR CDU FAIL.

STDESIG1        CCS     DESCOUNT        # SEE IF THE TIME LIMIT HAS EXPIRED
                TCF     MOREDES

                CS      B14+B2          # IF OUT OF TIME, REMOVE ECR ENABLE + TRKR
                EXTEND
                WAND    CHAN12
BADDES          CS      DESIGBIT        # REMOVE DESIGNATE FLAG
                MASK    RADMODES
                TS      RADMODES
                TCF     RDBADEND

MOREDES         TS      DESCOUNT
                CAF     PRIO26          # UPDATE GYRO TORQUE COMMANDS.
                TC      FINDVAC
                EBANK=  LOSCOUNT
                2CADR   DODES

                TCF     DESLOOP

B14+B2          OCT     20002
```

```
# PROGRAM NAME:  DODES
#
# FUNCTIONAL DESCRIPTION:
#
#       DODES CALCULATES AND REQUESTS ISSUANCE OF RR GYRO TORQUE
#       COMMANDS.  INITIALLY THE CURRENT RR CDU ANGLES ARE STORED AND
#       THE LOS HALF-UNIT VECTOR TRANSFORMED FROM STABLE MEMBER TO
#       NAVIGATION BASE COORDINATES VIA SMNB IF NECESSARY.  THE
#       SHAFT AND TRUNNION COMMANDS ARE THEN CALCULATED AS FOLLOWS:
#               + SHAFT = LOS . (COS(S), 0, -SIN(S))  (DOT PRODUCT)
#               - TRUNNION = LOS . (SIN(T)SIN(S), COS(T), SIN(T)COS(S))
#       THE SIGN OF THE SHAFT COMMAND IS THEN REVERSED IF IN MODE 2
#       (RADMODES BIT 12) BECAUSE A RELAY IN THE RR REVERSES THE
#       POLARITY OF THE COMMAND.  AT RRSCALUP EACH COMMAND IS
#       SCALED AND IF EITHER, OR BOTH, OF THE COMMANDS IS GREATER THAN
#       .5 DEGREES, MPAC +1 IS SET POSITIVE.  IF A CONTINUOUS DESIGNATE
#       (RADMODES BIT 15) IS DESIRED AND THE SEARCH ROUTINE IS NOT OPERATING,
#       THE RR AUTO TRACKER ENABLE BIT (CHAN 12 BIT 14) IS CLEARED AND RROUT
#       CALLED TO PUT OUT THE COMMANDS PROVIDED NO REPOSITION (RADMODES BIT 11)
#       IS IN PROGRESS.  IF A CONTINUOUS DESIGNATE AND THE SEARCH ROUTINE IS
#       OPERATING (SRCHOPT FLAT SET) THE TRACK ENABLE IS NOT CLEARED.  IF NO
#       CONTINUOUS DESIGNATE AND BOTH COMMANDS ARE NOT LESS THAN .5 DEGREES AS
#       INDICATED BY MPAC +1, THE RR AUTO TRACKER ENABLE BIT (CHAN 12 BIT 14) IS
#       CLEARED AND RROUT CALLED TO PUT OUT THE COMMANDS PROVIDED NO REPOSITON
#       (RADMODES BIT 11) IS IN PROGRESS.  IF BOTH COMMANDS ARE LESS THAN .5
#       DEGREES AS INDICATED BY MPAC+1, THE RR AUTO TRACKER ENABLE BIT
#       (CHAN 12 BIT 14) IS CLEARED AND RROUT CALLED TO PUT OUT THE
#       COMMANDS PROVIDED NO REPOSITION (RADMODES BIT 11) IS IN
#       PROGRESS.  IF BOTH COMMANDS ARE LESS THAN .5 DEGREES, THE
#       LOCK-ON FLAG (STATE BIT 5) IS CHECKED.  IF NOT PRESETN, THE
#       DESIGNATE FLAG (RADMODES BIT 10) IS CLEARED, AND ENDOFJOB
#       CALLED.  IF LOCK-ON IS DESIRED, THE RR AUTO TRACKER (CHAN 12
#       BIT 14) IS ENABLED FOLLOWED BY A CHECK OF THE RECEIPT OF THE
#       RR DATA GOOD (CHAN 33 BIT 4) SIGNAL.  IF RR DATA GOOD
#       PRESENT, THE DESIGNATE FLAG (RADMODES BIT 10) IS CLEARED,
#       THE RR ERROR COUNTER ENABLE BIT (CHAN 12 BIT 2) IS CLEARED,
#       AND ENDOFJOB CALLED.  IF RR DATA GOOD IS NOT PRESENT, RROUT
#       IS CALLED TO PUT OUT THE COMMANDS PROVIDED NO REPOSITION
#       (RADMODES BIT 11) IS IN PROGRESS AFTER WHICH THE JOB IS TERMINATED
#       VIA ENDOFJOB.
#
# CALLING SEQUENCE:
#
#       EXECUTIVE CALL EVERY .5 SECONDS FROM BEGDES.
#
# ERASABLE INITIALIZATION REQUIRED:
#
#       RRTARGET (HALF-UNIT LOS VECTOR IN EITHER SM OR NB COORDINATES),
#       LOKONSW (STATE BIT 5), RRNBSW (STATE BIT 6), RADMODES
#
# SUBROUTINES CALLED:
#
#       READCDUS, SMNB, CDULOGIC, MAGSUB, RROUT
```

```
#
# JOBS OR TASKS INITIATED:
#
#       NONE
#
# ALARMS:  NONE
#
# EXIT:  ENDOFJOB (ALWAYS)

DODES           EXTEND
                DCA     CDUT
                DXCH    TANG

                TC      INTPRET

                SETPD   VLOAD
                        0
                        RRTARGET
                BON     VXSC
                        RRNBSW
                        DONBRD          # TARGET IN NAV-BASE COORDINATES
                        MLOSV           # MULTIPLY UNIT LOS BY MAGNITUDE
                VSL1    PDVL
                        LOSVEL
                VXSC    VAD             # ADD ONE SECOND RELATIVE VELOCITY TO LOS
                        MCTOMS
                UNIT    CALL
                        CDUTRIG
                CALL
                        *SMNB*

DONBRD          STODL   32D
                        TANG +1
                RTB     PUSH            # SHAFT COMMAND = V(32D).(COS(S), 0,
                        CDULOGIC        #       -SIN(S)).
                SIN     PDDL            # SIN(S) TO 0 AND COS(S) TO 2.
                COS     PUSH
                DMP     PDDL
                        32D
                        36D
                DMP     BDSU
                        0
                STADR
                STORE   TANG +1         # SHAFT COMMAND

                SLOAD   RTB
                        TANG
                        CDULOGIC
                PUSH    COS             # COS(T) TO 4.
                PDDL    SIN
                PUSH    DMP             # SIN(T) TO 6.
                        2
```

```
          SL1       PDDL            # DEFINE VECTOR U =     [SIN(T)SIN(S)]
                    4               #                       [   COS(T)   ]
          PDDL      DMP             #                       [SIN(T)COS(S)]
                    6
                    0
          SL1       VDEF
          DOT       EXIT            # DOT U WITH LOS TO GET TRUNNION COMMAND.
                    32D
```

```
# AT THIS POINT WE HAVE A ROTATION VECTOR IN DISH AXES LYING IN THE TS PLANE.  CONVERT THIS TO A
# COMMANDED RATE AND ENABLE THE TRACKER IF WE ARE WITHIN .5 DEGREES OF THE TARGET.

                CS      MPAC            # DOT WAS NEGATIVE OF DESREG ANGLE.
                EXTEND
                MP      RDESGAIN        # SCALING ON INPUT ANGLE WAS 4 RADIANS.
                TS      TANG            # TRUNNION COMMAND.
                CS      RADMODES        # A RELAY IN THE RR REVERSES POLARITY OF
                MASK    BIT12           # THE SHAFT COMMANDS IN MODE 2 SO THAT A
                EXTEND                  # POSITIVE TORQUE APPLIED TO THE SHAFT
                BZF     +3              # GYRO CAUSES A POSITIVE CHANGE IN THE
                CA      TANG +1         # SHAFT ANGLE.  COMPENSATE FOR THIS SWITCH
                TCF     +2              # BY CHANGING THE POLARITY OF OUR COMMAND.
 +3             CS      TANG +1
                EXTEND
                MP      RDESGAIN        # SCALING ON INPUT ANGLE WAS 4 RADIANS.
                TS      TANG +1         # SHAFT COMMAND FOR RROUT
                TC      INTPRET

                DLOAD   DMP
                        2               # COS(S).
                        4               # COS(T).
                SL1     PDDL            # Z COMPONENT OF URR.
                DCOMP   PDDL            # Y COMPONENT = -SIN(T)
                        0               # SIN(S).
                DMP     SL1
                        4               # COS(T).
                VDEF    BON             # FORM URR IN NB AXES.
                        RRNBSW          # BYPASS NBSM CONVERSION IN VERB 41
                        +3
                CALL
                        *NBSM*          # GET URR IN SM AXES.
                DOT     EXIT
                        RRTARGET        # GET COSIN OF ANGLE BETWEEN RR AND LOS

                EXTEND
                DCS     COS1/2DG
                DAS     MPAC            # DIFFERENCE OF COSINES, SCALED B-2.
                CCS     MPAC
                CA      ZERO            # IF COS ERROR BIGGER, ERROR IS SMALLER
                TCF     +2
                CA      ONE
                TS      MPAC +1         # ZERO IF RR IS POINTED OK, ONE IF NOT.
```

```
# SEE IF TRACKER SHOULD BE ENABLED OR DISABLED.

                CCS     RADMODES        # IF CONTINUOUS DESIGNATE WANTED, PUT OUT
                TCF     SIGNLCHK        # COMMANDS WITHOUT CHECKING MAGNITUDE OF
                TCF     SIGNLCHK        # ERROR SIGNALS
                TCF     DORROUT
SIGNLCHK        CCS     MPAC +1         # SEE IF BOTH AXES WERE WITHIN .5 DEGS.
                TCF     DGOODCHK
                CS      STATE           # IF WITHIN LIMITS AND NO LOCK-ON WANTED,
                MASK    LOKONBIT        # PROBLEM IS FINISHED.
                CCS     A
                TCF     RRDESDUN

                CAF     BIT14           # ENABLE THE TRACKER
                EXTEND
                WOR     CHAN12

DGOODCHK        CAF     BIT4            # SEE IF DATA GOOD RECEIVED YET
                EXTEND
                RAND    CHAN33
                CCS     A
                TCF     DORROUT

RRDESDUN        CS      BIT10           # WHEN PROBLEM DONE, REMOVE BIT 10 SO NEXT
                MASK    RADMODES        # WAITLIST TASK WE WILL GO TO RGOODEND.
                INHINT
                TS      RADMODES

                TC      DOWNFLAG        # RESET LOSCMFLG TO PREENT A
                ADRES   LOSCMFLG        # RECOMPUTATION OF LOS AFTER DATA GOOD
                CS      BIT2            # TURN OFF ENABLE RR ERROR COUNTER
                EXTEND
                WAND    CHAN12
                TCF     ENDOFJOB        # WITH ECTR DISABLED.

DORROUT         CA      FLAGWRD2        # IF BOTH LOSCMFLAG AND SEARCH FLAG ARE
                MASK    BIT12,14        # ZERO, BYPASS VELOCITY ADJUSTMENT TO LOS
                EXTEND
                BZF     NOTP20
                TC      INTPRET
                VLOAD   VXSC            # MULTIPLY UNIT LOS BY MAGNITUDE
                        RRTARGET
                        MLOSV
                VSL1    PUSH
                VLOAD   VXSC            # ADD .5 SEC. OF VELOCITY
                        LOSVEL          # TO LOS VECTOR
                        MCTOMS
                VSR1    VAD
                UNIT
                STODL   RRTARGET        # STORE VELOCITY-CORRECTED LOS (UNIT)
```

```
                        36D
            STORE       MLOSV           # AND STORE MAGNITUDE
            EXIT
NOTP20      INHINT
            CS          RADMODES        # PUT OUT COMMAND UNLESS MONITOR
            MASK        REPOSBIT        # REPOSITION HAS TAKEN OVER
            CCS         A
            TC          RROUT

            CA          FLAGWRD2
            MASK        LOSCMBIT        # IF LOSCMFLG NOT SET, DON'T TEST
            EXTEND                      # LOS COUNTER
            BZF         ENDOFJOB
            CCS         LOSCOUNT        # TEST LOS COUNTER TO SEE IF TIME TO GET
            TC          DODESEND        # A NEW LOS
            INHINT
            TC          KILLTASK        # YES -- KILL TASK WHICH SCHEDULES DODES
            CADR        DESLOOP +2
            RELINT
            CCS         NEWJOB
            TC          CHANG1
            TC          BANKCALL
            CADR        R21LEM2

DODESEND    TS          LOSCOUNT
            TC          ENDOFJOB

RDESGAIN    DEC         .53624          # TRIES TO NULL .5 ERROR IN .5 SEC.
BIT12,14    EQUALS      PRIO24          # OCT 24000
COS1/2DG    2DEC        .999961923 B-2  # COSINE OF 0.5 DEGREES.
MCTOMS      2DEC        100 B-13
```

```
# RADAR READ INITIALIZATION
#
# RADAR DATA READ BY A BANKCALL FOR THE APPROPRIATE LEAD-IN BELOW.

LRALT           TC      INITREAD -1     # ONE SAMPLE PER READING.
ALLREAD         OCT     17

LRVELZ          TC      INITREAD
                OCT     16

LRVELY          TC      INITREAD
                OCT     15

LRVELX          TC      INITREAD
                OCT     14

RRRDOT          TC      INITREAD -1
                OCT     12

RRRANGE         TC      INITREAD -1
                OCT     11

# LRVEL IS THE ENTRY TO THE LR VELOCITY READ ROUTINE WHEN 5 SAMPLES ARE
# WANTED.  ENTER WITH C(A)= 0,2,4 FOR LRVELZ,LRVELY,LRVELX RESP.

LRVEL           TS      TIMEHOLD        # STORE VBEAM INDEX HERE MOMENTARILY
                CAF     FIVE            # SPECIFY FIVE SAMPLES
                INDEX   TIMEHOLD
                TCF     LRVELZ
```

```
 -1             CAF     ONE             # ENTRY TO TAKE ONLY 1 SAMPLE
INITREAD        INHINT

                TS      TIMEHOLD        # GET DT OF MIDPOINT OF NOMINAL SAMPLING
                EXTEND                  # INTERVAL (ASSUMES NO BAD SAMPLES WILL BE
                MP      BIT3            # ENCOUNTERED).
                DXCH    TIMEHOLD

                CCS     A
                TS      NSAMP
                AD      ONE
# INSERT FOLLOWING INSTRUCTION TO GET 2N TRIES FOR N SAMPLES.
#               DOUBLE
                TS      SAMPLIM

                CAF     DGBITS          # READ CURRENT VALUE OF DATA GOOD BITS.
                EXTEND
                RAND    CHAN33
                TS      OLDATAGD

                CS      ALLREAD
                EXTEND
                WAND    CHAN13          # REMOVE ALL RADAR BITS

                INDEX   Q
                CAF     0
                EXTEND
                WOR     CHAN13          # SET NEW RADAR BITS

                EXTEND
                DCA     TIME2
                DAS     TIMEHOLD        # TIME OF NOMINAL MIDPOINT

                CAF     ZERO
                TS      L
                DXCH    SAMPLSUM
                TCF     ROADBACK

DGBITS          OCT     230
```

```
# RADAR RUPT READER
#
# THIS ROUTINE STARTS FROM A RADARUPT.  IT READS THE DATA & LOTS MORE.

                SETLOC  RADARUPT
                BANK

                COUNT*  $$/RRUPT
RADAREAD        EXTEND                          # MUST SAVE SBANK BECAUSE OF RUPT EXITS
                ROR     SUPERBNK                # VIA TASKOVER (BADEND OR GOODEND).
                TS      BANKRUPT
                EXTEND
                QXCH    QRUPT

                CAF     SEVEN
                EXTEND
                RAND    CHAN13
                TS      DNINDEX
                EXTEND                          # IF RADAR SELECT BITS ZERO, DO NOT STORE
                BZF     TRYCOUNT                # DATA FOR DOWNLIST (ERASABLE PROBLEMS)
                CA      RNRAD
                INDEX   DNINDEX
                TS      DNRRANGE -1
TRYCOUNT        CCS     SAMPLIM
                TCF     PLENTY
                TCF     NOMORE
                TC      ALARM
                OCT     520
                TC      RESUME

NOMORE          CA      FLGWRD11                # IS LRBYPASS SET?
                MASK    LRBYBIT
                EXTEND
                BZF     BADRAD                  # NO.  R12 IS ON -- BYPASS 521 ALARM.

                CS      FLAGWRD3                # CHECK R04FLAG.
                MASK    R04FLBIT                # IF 1, R04 IS RUNNING.  DO NOT ALARM
                EXTEND
                BZF     BADRAD

                TC      ALARM                   # P20 WANTS THE ALARM.
                OCT     521
BADRAD          CS      ONE
                TS      SAMPLIM
                TC      RDBADEND -2
PLENTY          TS      SAMPLIM
                CAF     BIT3
                EXTEND
                RAND    CHAN13                  # TO FIND OUT WHICH RADAR
                EXTEND
```

```
                BZF     RENDRAD

                TC      R77CHECK        # R77 QUITS HERE.
LRPOSCHK        CA      RADMODES        # SEE IF LR IN DESIRED POSITION
                EXTEND
                RXOR    CHAN33
                MASK    BIT6
                EXTEND
                BZF     VELCHK

                TC      ALARM
                OCT     522
                TC      BADRAD

VELCHK          CAF     BIN3            # = 00003 OCT
                EXTEND
                RXOR    CHAN13          # RESET ACTIVITY BIT
                MASK    BIN3
                EXTEND
                BZF     LRHEIGHT        # TAKE A LR RANGE READING

                CAF     POSMAX
                MASK    RNRAD
                AD      LVELBIAS
                TS      L
                CAE     RNRAD
                DOUBLE
                MASK    BIT1
                DXCH    ITEMP3

                CAF     BIT8            # DATA GOOD ISN'T CHECKED UNTIL AFTER READ-
                TC      DGCHECK         # ING DATA SO SOME RADAR TESTS WILL WORK
                                        # INDEPENDENT OF DATA GOOD.

                CCS     NSAMP
                TC      NOEND
GOODRAD         CS      ONE
                TS      SAMPLIM
                CS      ITEMP1          # WHEN ENOUGH GOOD DATA HAS BEEN GATHERED,
                MASK    RADMODES        # RESET DATA FAIL FLAGS FOR SETTRKF.
                TS      RADMODES
                TC      RADLITES        # LAMPS MAY GO OFF IF DATA JUST GOOD.
                TC      RGOODEND -2

NOEND           TS      NSAMP
RESAMPLE        CCS     SAMPLIM         # SEE IF ANY MORE TRIES SHOULD BE MADE.
                TCF     +2
                TCF     DATAFAIL        # N SAMPLES NOT AVAILABLE.
                CAF     BIT4            # RESET ACTIVITY BIT.
                EXTEND
```

```
                WOR     CHAN13          # RESET ACTIVITY BIT
                TC      RESUME

LRHEIGHT        CAF     BIT5
                TS      ITEMP1          # (POSITION OF DATA GOOD BIT IN CHAN 33)

                CAF     BIT9
                TC      SCALECHK -1

RENDRAD         CAF     REPOSBIT        # MAKE SURE ANTENNA HAS NOT GONE OUT OF
                MASK    RADMODES        # LIMITS.
                CCS     A
                TCF     BADRAD

                CS      RADMODES        # BE SURE RR CDU HASN'T FAILED.
                MASK    RCDUFBIT
                CCS     A
                TCF     BADRAD

                CAF     BIT4            # SEE IF DATA HAS BEEN GOOD.
                TS      ITEMP1          # (POSITION OF DATA GOOD BIT IN CHAN 33)

                CAF     BIT1            # SEE IF RR RDOT.
                EXTEND
                RAND    CHAN13
                TS      Q               # FOR LATER TESTING.
                CCS     A
                TCF     +2
                TCF     RADIN           # NO SCALE CHECK FOR RR RDOT.
                CAF     BIT3
                TS      L

SCALECHK        EXTEND
                RAND    CHAN33          # SCALE STATUS NOW
                XCH     L
                MASK    RADMODES        # SCALE STATUS BEFORE
                EXTEND
                RXOR    LCHAN           # SEE IF THEY DIFFER
                CCS     A
                TC      SCALCHNG        # THEY DIFFER.

RADIN           CAF     POSMAX
                MASK    RNRAD
                TS      ITEMP4

                CAE     RNRAD
                DOUBLE
                MASK    BIT1
                TS      ITEMP3
```

```
                CCS     Q               # SEE IF RR RDOT.
                TCF     SCALADJ         # NO, BUT SCALE CHANGING MAY BE NEEDED.

                EXTEND                  # IF RR RANGE RATE, THROW OUT BIAS
                DCS     RDOTBIAS
DASAMPL         DAS     ITEMP3
DGCHECK2        CA      ITEMP1          # SEE THAT DATA HAS BEEN GOOD BEFORE AND
                TC      DGCHECK +1      # AFTER TAKING SAMPLE.
                TC      GOODRAD

SCALCHNG        LXCH    RADMODES
                AD      BIT1
                EXTEND
                RXOR    LCHAN
                TS      RADMODES
                CAF     DGBITS          # UPDATE LAST VALUE OF DATA GOOD BITS.
                EXTEND
                RAND    CHAN33
                TS      OLDATAGD
                TC      UPFLAG          # SET RNGSCFLG
                ADRES   RNGSCFLG        # FOR LRS24.1
                TCF     BADRAD

# R77 MUST IGNORE DATA FAILS SO AS NOT TO DISTURB THE ASTRONAUT.

R77CHECK        CS      FLAGWRD5
                MASK    R77FLBIT
                CCS     A
                TC      Q               # NOT R77
                CS      BITS5,8         # UPDATE LR DATA GOOD BITS IN RADMODES
                MASK    RADMODES
                TS      L
                CA      BITS5,8
                EXTEND
                RAND    CHAN33
                AD      L
                TS      RADMODES
                TC      RGOODEND -2
BITS5,8         OCT     220
```

```
# THE FOLLOWING ROUTINE INCORPORATES RR RANGE AND LR ALT SCALE INFORMATION AND LEAVES DATA AT LO SCALE.

SCALADJ         CCS     L                   # L HAS SCALE INBIT FOR THIS RADAR.
                TCF     +2                  # ON HIGH SCALE.
                TCF     DGCHECK2

                CA      DNINDEX
                MASK    BIT3
                CCS     A
                TCF     LRSCK

                DXCH    ITEMP3
                DDOUBL
                DDOUBL
                DDOUBL
                DXCH    ITEMP3

                TCF     DGCHECK2

LRSCK           CCS     ITEMP3
                TCF     +11
                CS      ITEMP4
                AD      HISCALIM
                EXTEND
                BZMF    +5

                CS      FLGWRD11
                MASK    SCABBIT
                ADS     FLGWRD11
                TCF     +4

                CS      SCABBIT
                MASK    FLGWRD11
                TS      FLGWRD11

                EXTEND
                DCA     ITEMP3
                DDOUBL
                DDOUBL
                TCF     DASAMPL

HISCALIM        DEC     460                 # 2481.7 FT ******************************
```

```
DGCHECK         TS      ITEMP1          # UPDATE DATA GOOD BIT IN OLDATAGD AND
                EXTEND                  # MAKE SURE IT WAS ON BEFORE AND AFTER THE
                RAND    CHAN33          # SAMPLE WAS TAKEN BEFORE RETURNING.  IF
                TS      L               # NOT, GOES TO RESAMPLE TO TRY AGAIN.  IF
                CS      ITEMP1          # MAX NUMBER OF TRIES HAS BEEN REACHED,
                MASK    OLDATAGD        # THE BIT CORRESPONDING TO THE DATA GOOD
                AD      L               # WHICH FAILED TO APPEAR IS IN ITEMP1 AND
                XCH     OLDATAGD        # CAN BE USED TO SET RADMODES WHICH VIA
                MASK    ITEMP1          # SETTRKF SETS THE TRACKER FAIL LAMP.
                AD      L
                CCS     A               # SHOULD BOTH BE ZERO.
                TC      RESAMPLE
                DXCH    ITEMP3          # IF DATA GOOD BEFORE AND AFTER, ADD TO
                DAS     SAMPLSUM        # ACCUMULATION.
                TC      Q

DATAFAIL        CS      ITEMP1          # IN THE ABOVE CASE, SET RADMODES BIT
                MASK    RADMODES        # SHOWING SOME RADAR DATA FAILED.
                AD      ITEMP1
                TS      RADMODES

                DXCH    ITEMP3          # IF WE HAVE BEEN UNABLE TO GATHER N
                DXCH    SAMPLSUM        # SAMPLES, USE LAST ONE ONLY.
                TC      RADLITES
                TCF     NOMORE
```

```
# THIS ROUTINE CHANGES THE LR POSITION, AND CHECKS THAT IT GOT THERE.

                SETLOC  P20S1
                BANK

                COUNT*  $$/RSUB
LRPOS2          INHINT

                CS      RADMODES
                MASK    LRPOSBIT        # SHOW DESIRED LR POSITION IS 2
                ADS     RADMODES

                CAF     BIT7
                EXTEND
                RAND    CHAN33          # SEE IF ALREADY THERE.
                EXTEND
                BZF     RADNOOP

                CAF     BIT13
                EXTEND
                WOR     CHAN12          # COMMAND TO POSITION 2
                CAF     6SECS           # START SCANNING FOR INBIT AFTER 7 SECS.
                TC      WAITLIST
                EBANK=  LOSCOUNT
                2CADR   LRPOSCAN

                TC      ROADBACK

LRPOSNXT        TS      SAMPLIM
                TC      FIXDELAY        # SCAN ONCE PER SECOND 15 TIMES MAX AFTER
                DEC     100             # INITIAL DELAY OF 7 SECONDS.

                CAF     BIT7            # SEE IF LR POS2 IS ON
                EXTEND
                RAND    CHAN33
                EXTEND
                BZF     LASTLRDT        # IF THERE, WAIT FINAL SECOND FOR BOUNCE.

                CCS     SAMPLIM         # SEE IF MAX TIME UP.
                TCF     LRPOSNXT

                CS      BIT13           # IF TIME UP, DISABLE COMMAND AND ALARM.
                EXTEND
                WAND    CHAN12
                TCF     RDBADEND

RADNOOP         CAF     ONE             # NO FURTHER ACTION REQUESTED.
                TC      WAITLIST
                EBANK=  LOSCOUNT
                2CADR   RGOODEND
```

```
                TC      ROADBACK

LASTLRDT        CA      2SECS           # WAIT TWO SECONDS AFTER RECEIPT OF INBIT
                TC      VARDELAY        # TO WAIT FOR ANTENNA BOUNCE TO DIE OUT.

                CS      BIT13           # REMOVE COMMAND
                EXTEND
                WAND    CHAN12
                TCF     RGOODEND

LRPOSCAN        CAF     FOURTEEN        # SET UP FOR 15 SAMPLES.
                TCF     LRPOSNXT
6SECS           DEC     600
```

```
# SEQUENCES TO TERMINATE RR OPERATIONS.

ENDRADAR        CAF     RCDUFBIT        # PROLOG TO CHECK RR CDU FAIL BEFORE END.
                MASK    RADMODES
                CCS     A
                TCF     RGOODEND
                TCF     RDBADEND
 -2             CS      ZERO            # RGOODEND WHEN NOT UNDER WAITLIST CONTROL
                TS      RUPTAGN

RGOODEND        CAF     TWO
                TC      POSTJUMP
                CADR    GOODEND

 -2             CS      ZERO            # RDBADEND WHEN NOT UNDER WIATLIST.
                TS      RUPTAGN
RDBADEND        CAF     TWO
                TC      POSTJUMP
                CADR    BADEND

BIN3            EQUALS  THREE
```

```
# PROGRAM NAME:  LPS20.1 VECTOR EXTRAPOLATION AND LOS COMPUTATION
# MOD. NO. 2     BY J.D. COYNE    SDC      DATE 12-7-66
#
# FUNCTIONAL DESCRPIPTION:
#       1)      EXTRAPOLATE THE LEM AND CSM VECTORS IN ACCORDANCE WITH THE TIME REFERRED TO IN CALLER + 1.
#       2)      COMPUTES THE LOS VECTOR TO THE CSM, CONVERTS IT TO STABLE MEMBER COORDINATES AND STORES IT IN RRTARGET.
#       3)      COMPUTES THE MAGNITUDE OF TEH LOS VECTOR AND STORES IT IN MLOSV
#
# CALLING SEQUENCE:     CALL
#                               LPS20.1
#
# SUBROUTINES CALLED:
#       LEMPREC, CSMPREC
#
# NORMAL EXIT:  RETURN TO CALLER + 2.
#
# ERROR EXITS:  NONE
#
# ALARMS:  NONE
#
# OUTPUT:
#       LOS VECTOR (HALF UNIT) IN SM COORDINATES STORED IN RRTARGET
#       MAGNITUDE OF TEH LOS VECTOR (METERS SCALED B-29) STORED IN MSLOV
#       RRNBSW CLEARED.
#
# INITIALIZED ERASABLE
#       TDEC1 MUST CONTAIN THE TIME FOR EXTRAPOLATION
#       SEE ORBITAL INTEGRATION ROUTINE
#
# DEBRIS:
#       MPAC DESTROYED BY THE ROUTINE

                BANK    23
                SETLOC  P20S
                BANK
```

```
                COUNT*  $$/LPS20

LPS20.1         STQ     BOFF
                        LS21X
                        LOSCMFLG        # LOSCMFLG = 0 MEANS NOT CALLED BY R21
                        LMINT           # SO CALL LEMCONIC TO GET LM STATE
                BON                     # IF IN R21 AND ON LUNAR SURFACE
                        SURFFLAG        # DON'T CALL LEMCONIC
                        CSMINT
LMINT           CALL
                        LEMCONIC        # EXTRAPOLATE LEM
                VLOAD
                        RATT
                STOVL   LMPOS           # SAVE LM POSITION B-29
                        VATT
                STODL   LMVEL           # SAVE LM VELOCITY B-7
                        TAT
CSMINT          STCALL  TDEC1
                        CSMCONIC        # EXTRAPOLATE CSM
                VLOAD   VSU             # COMPUTE RELATIVE VELOCITY V(CSM) - V(LM)
                        VATT
                        LMVEL
                MXV     VSL1
                        REFSMMAT
                EXIT
                TC      KILLTASK        # KILL THE TASK WHICH CALLS DODES SINCE
                CADR    DESLOOP +2      # STORING INTO ERASEABLES DODES USES
                TC      INTPRET
                STOVL   LOSVEL
                        RATT
                VSU     BOFF
                        LMPOS
                        RNDVZFLG
                        NOTSHIFT
                BOVB
                        TCDANZIG
                VSL
                        9D
NOTSHIFT        UNIT    BOVB            # IF OVERFLOW, RANGE MUST BE GREATER
                        526ALARM        # THAN 400 N. M.
                MXV     VSL1
                        REFSMMAT        # CONVERT TO STABLE MEMBER
                STODL   RRTARGET
                        36D             # SAVE MAGNITUDE OF LOS VECTOR FOR
                STORE   MLOSV           # VELOCITY CORRECTION IN DESIGNATE
                CLRGO
                        RRNBSW
                        LS21X
```

```
# PROGRAM NAME:  LPS20.2  400 NM RANGE CHECK
# MOD. NO. 2    BY J.D. COYNE   SDC     DATE 12-7-66
#
# FUNCTIONAL DESCRIPTION:
#       COMPARES THE MAGNITUDE OF THE LOS VECTOR TO 400 NM.
#
# CALLING SEQUENCE:     CALL
#                               LPS20.2
#
# SUBROUTINES CALLED:  NONE
#
# NORMAL EXIT:  RETURN TO CALLER +1, MPAC EQ 0 (RANGE 400NM OR LESS.)
#
# ERROR EXITS:  RETURN TO CALLER +1, MPAC EQ 1 (RANGE GREATER THAN 400NM)
#
# ALARMS:  NONE
#
# OUTPUT:  NONE
#
# INITIALIZED ERASEABLE:
#       PDL 36D MUST CONTAIN THE MAGNITUDE OF THE VECTOR
#
# DEBRIS:
#       MPAC DESTROYED BY THIS ROUTINE

                SETLOC  P20S1
                BANK
                COUNT*  $$/LPS20

LPS20.2         DLOAD   DSU
                        MLOSV           # MAGNITUDE OF LOS
                        FHNM            # OVER 400NM
                BPL
                        TOFAR
                SLOAD   RVQ
                        ZERO/SP
TOFAR           SLOAD   RVQ
                        ONE/SP
ONE/SP          DEC     1
```

FHNM 2DEC 740800 B-20 # 400 NAUTICAL MILES IN METERS B-20

```
# PROGRAM NAME:  LRS22.1 (DATA READ SUBROUTINE 1)
# MOD. NO.: 1            BY: P. VOLANTE  SDC              DATE:  11-15-66
#
# FUNCTIONAL DESCRIPTION:
#       1)      READS RENDEZVOUS RADAR RANGE AND RANGE-RATE, TRUNNION AND SHAFT ANGLES, THREE CDU VALUES AND TIME.  CONVERTS THIS
#               DATA AND LEAVES IT FOR THE MEASUREMENT INCORPORATION ROUTINE (LSR22.3).  CHECKS FOR THE RR DATA GOOD DISCRETE, FOR
#               RR REPOSITION AND RR CDU FAIL
#       2)      COMPARES RADAR LOS WITH LOS COMPUTED FROM STATE VECTORS TO SEE IF THEY ARE WITHIN THREE DEGREES
#
# CALLING SEQUENCE:  BANKCALL FOR LRS22.1
#
# SUBROUTINES CALLED:
#       RRDOT           LPS20.1
#       RRRANGE         BANKCALL
#       RADSTALL        CDULOGIC
#       RRNB            SMNB
#
# NORMAL EXIT:  RETURN TO CALLER+1 WITH MPAC SET TO +0
#
# ERROR EXITS:  RETURN TO CALLER+1 WITH ERROR CODE STORED IN MPAC AS FOLLOWS:
#       00001 -- ERROR EXIT 1 -- RR DATA NO GOOD (NO RR DATA GOOD DISCRETE OR RR CDU FAIL OR RR REPOSITION)
#       00002 -- ERROR EXIT 2 -- RR LOS NOT WITHIN THREE DEGREES OF LOS COMPUTED FROM STATE VECTORS
#
# ALARMS:  521 -- COULD NOT READ RADAR DATA (RR DATA GOOD DISCRETE NOT PRESENT BEFORE AND AFTER READING THE RADAR)
#       (THIS ALARM IS ISSUED BY RADARREAD SUBROUTINE WHICH IS ENTERED FROM A RADARUPT)
#
# OUTPUT:  RRLOSVEC -- THE RR LINE-OF-SIGHT VECTOR (USED BY LRS22.2) -- A HALF-UNIT VECTOR
#       RM -- THE RR RANGE READING (TO THE CSM) DP, IN METERS SCALED BY B-29 (USED BY LRS22.2 AND LRS22.3)
#
#       ALL OF THE FOLLOWING OUTPUTS ARE USED BY LRS22.3:
#               RDOTM -- THE RR RANGE-RATE READING, DP, IN METERS PER CENTISECOND, SCALED BY B-7
#               RRTRUN -- THE RR TRUNNION ANGLE, DP, IN REVOLUTIONS, SCALED B0
#               RRSHAFT -- RR SHAFT ANGLE, DP, IN REVOLUTIONS, SCALED B0
#               AIG,AMG,AOG -- THE CDU ANGLES, THREE SP WORDS
#               MKTIME -- THE TIME OF THE RR READING, DP, IN CENTISECONDS
#
# ERASABLE INITIALIZATION REQUIRED:
#       RNRAD, THE RADAR READ COUNTER FROM WHICH IS OBTAINED:
```

```
#               1) RR RANGE SCALED 9.38 FT. PER BIT ON THE LOW SCALE AND 75.04 FT. PER BIT ON THE HIGH SCALE
#               2) RR RANGE RATE, SCALED .6278 FT./SEC. PER BIT
#       THE CDU ANGLES FROM CDUX, CDUY, CDUZ, AND TIME1 AND TIME2
#
# DEBRIS:  LRS22.1X, A, L, Q, PUSHLIST

                BANK    32
                SETLOC  LRS22
                BANK
                COUNT*  $$/LRS22

LRS22.1         TC      MAKECADR
                TS      LRS22.1X
                TC      DOWNFLAG
                ADRES   RNGSCFLG
                INHINT
                CAF     BIT3
                EXTEND                  # GET RR RANGE SCALE
                RAND    CHAN33          # FROM CHANNEL 33 BIT 3
                TS      L
                CS      RRRSBIT
                MASK    RADMODES
                AD      L
                TS      RADMODES
                RELINT
READRDOT        TC      BANKCALL
                CADR    RRRDOT          # READ RANGE-RATE (ONE SAMPLE)
                TC      BANKCALL
                CADR    RADSTALL        # WAIT FOR DATA READ COMPLETION
                TCF     EREXIT1         # COULD NOT READ RADAR-ERROR EXIT 1

                INHINT                  # NO INTERRUPTS WHILE READING TIME AND CDU
                DXCH    TIMEHOLD        # SET MARK TIME EQUAL TO THE MID-POINT
                DXCH    MPAC +5         # TEMP BUFFER FOR DOWNLINK
                DXCH    SAMPLSUM        # SAVE RANGE-RATE READING
                DXCH    RDOTMSAV
                EXTEND
                DCA     CDUY            # SAVE ICDU ANGLES
                DXCH    MPAC +3         # TEMP BUFFER FOR DOWNLINK
                CA      CDUX
                TS      MPAC +2         # TEMP BUFFER FOR DOWNLINK
                EXTEND
                DCA     TIME2           # SAVE TIME
                DXCH    MPAC            # SAVE TIME OF CDUY READINGS IN MPAC
                EXTEND
                DCA     CDUT            # SAVE TRUNNION AND SHAFT ANGLES FOR RRNB
                DXCH    TANG
```

```
                RELINT
                TC      BANKCALL
                CADR    RRRANGE         # READ RR RANGE (ONE SAMPLE)
                TC      BANKCALL
                CADR    RADSTALL        # WAIT FOR READ COMPLETE
                TC      CHEXERR         # CHECK FOR ERRORS DURING READ
                INHINT                  # COPY CYCLE FOR MARK DATA ON DOWNLINK
                DXCH    DNRRANGE        # RANGE, RANGE RATE (RAW DATA)
                DXCH    RANGRDOT
                DXCH    MPAC +5
                DXCH    MKTIME          # MARK TIME
                DXCH    MPAC +3
                DXCH    AIG             # CDUY, CDUZ
                EXTEND
                DCA     TANG            # PRESERVE TANG
                DXCH    TANGNB          # TRUNNION AND SHAFT ANGLES
                CA      MPAC +2
                TS      AOG             # CDUX
                TC      INTPRET
                STODL   20D             # SAVE TIME OF CDU READINGS IN 20D
                        RDOTMSAV        # CONVERT RDOT UNITS AND SCALING
                SL      DMPR            # START WITH READING SCALED B-28, -.6278
                        14D             # FT./SECOND PER BIT
                        RDOTCONV        # END WITH METERS/CENTISECOND, B-7
                STORE   RDOTM
                SLOAD   RTB
                        TANG            # GET TRUNNION ANGLE
                        CDULOGIC        # CONVERT TO DP ONES COMP. IN REVOLUTIOINS
                STORE   RRTRUN          # AND SAVE FOR TMI ROUTINE (LSR22.3)
                SLOAD   RTB
                        TANG +1         # DITTO FOR SHAFT ANGLE
                        CDULOGIC
                STODL   RRSHAFT
                        SAMPLSUM
                DMP     SL2R            # CONVERT UNITS AND SCALING DP RANGE
                        RANGCONV        # PER BIT, END WITH METERS, SCALED -29
                STCALL  RM
                        RRNB            # COMPUTE RADAR LOS USING RRNB
                STODL   RRBORSIT        # AND SAVE
                        20D
                STCALL  TDEC1           # GET STATE VECTOR LOS AT TIME OF CDU READ
                        LPS20.1
                EXIT
                CA      AIG             # STORE IMU CDU ANGLES AT MARKTIME
                TS      CDUSPOT         # IN CDUSPOT FOR TRG*SMNB
                CA      AMG
                TS      CDUSPOT +2
                CA      AOG
                TS      CDUSPOT +4
                TC      INTPRET
```

```
                VLOAD   CALL            # LOAD VECTOR AND CALL TRANSFORMATION
                        RRTARGET
                        TRG*SMNB        # ROTATE LOS AT MARKTIME FROM SM TO NB.
                DOT                     # DOT WITH RADAR LOS TO GET ANGLE
                        RRBORSIT
                SL1     ACOS            # BETWEEN THEM
                STORE   DSPTEM1         # STORE FOR POSSIBLE DISPLAY
                DSU     BMN             # IS IT LESS THAN 3 DEGREES
                        THREEDEG
                        NORMEXIT        # YES -- NORMAL EXIT

                EXIT                    # ERROR EXIT 2
                CAF     BIT2            # SET ERROR CODE
                TS      MPAC
                TCF     OUT22.1

NORMEXIT        EXIT                    # NORMAL EXIT -- SET MPAC EQUAL ZERO
                CAF     ZERO
                TS      MPAC
OUT22.1         CAE     LRS22.1X        # EXIT FROM LRS22.1
                TC      BANKJUMP
CHEXERR         CAE     FLAGWRD5
                MASK    RNGSCBIT
                CCS     A               # CHECK IF RANGE SCALE CHANGED
                TCF     READRDOT        # YES -- TAKE ANOTHER READING

EREXIT1         CA      BIT1            # SET ERROR CODE
                TS      MPAC
                TC      OUT22.1
THREEDEG        2DEC    .008333333      # THREE DEGREES, SCALED REVS, B0

RRLOSVEC        EQUALS  RRTARGET
```

```
# PROGRAM NAME -- LRS22.2 (DATA READ SUBROUTINE 2)
# MOD. NO.: 1            BY: P. VOLANTE  SDC              DATE: 4-11-67
#
# FUNCTIONAL DESCRIPTION:
#       (YES, I KNOW POINT #1 IS MISSING.  IT IS MISSING FROM THE PROGRAM LISTING -- RSB 2003)
#       2)  CHECKS IF THE RR LOS (I.E., THE RADAR BORESIGHT VECTOR) IS WITHIN 30 DEGREES OF THE LM +Z AXIS
#
# CALLING SEQUENCE:  BANKCALL FOR LRS22.2
#
# SUBROUTINES CALLED:  G+N, AUTO, SETMAXDB
#
# NORMAL EXIT:  RETURN TO CALLER WITH MPAC SET TO +0 (VIA SWRETURN)
#
# ERROR EXIT:  RETURN TO CALLER WITH MPAC SET TO 00001 -- RADAR LOS NOT WITHIN 30 DEGREES OF LM +Z AXIS.
#
# ALARMS:  NONE
#
# ERASABLE INITIALIZATION REQUIRED:
#       RRLOSVEC -- THE RR LINE-OF-SIGHT VECTOR -- A HALF UNIT VECTOR COMPUTED BY LRS22.1
#       RM -- RR RANGE, METERS B-29, FROM LRS22.1
#       BIT 14 CHANNEL 31 -- INDICATES AUTOPILOT IS IN AUTO MODE
#
# DEBRIS -- A,L,Q,MPAC -- PUSHLIST AND PUSHLOC ARE NOT CHANGED BY THIS ROUTINE

                SETLOC  P20S
                BANK
LRS22.2         TC      MAKECADR
                TS      LRS22.1X
                TC      INTPRET
                                        # CHECK IF RR LOS IS WITHIN 30 DEG OF
30DEGCHK        DLOAD   ACOS            # THE SPACECRAFT +Z AXIT
                        RRBORSIT +4     # BY TAKING ARCCOS OF Z-COMP. OF THE RR
                                        # LOS VECTOR, A HALF UNIT VECTOR
                                        # IN NAV BASE AXES)
                DSU     BMN
                        30DEG
                        OKEXIT          # NORMAL EXIT -- WITHIN 30 DEG.
                EXIT                    # ERROR EXIT -- NOT WITHIN 30 DEG.
                CAF     BIT1            # SETS ERROR CODE IN MPAC
                TS      MPAC
                TCF     OUT22.2
OKEXIT          EXIT                    # NORMAL EXIT -- SET MPAC = ZERO
```

```
                CAF     ZERO
                TS      MPAC
OUT22.2         CAE     LRS22.1X
                TC      BANKJUMP

30DEG           2DEC    .083333333      # THIRTY DEGREES, SCALED REVS, B0
```

```
# PROGRAM NAME -- LSR22.3                                      DATE -- 29 MAY 1967
# MOD. NO 3                                                    LOG SECTION -- P20-P25
# MOD. BY -- DANFORTH                                          ASSEMBLY LEMP20S REV 10
#
# FUNCTIONAL DESCRIPTION:
#       THIS ROUTINE COMPUTES THE B-VECTORS ADN DELTA Q FOR EACH OF THE QUANTITIES MEASURED BY THE RENDEZVOUS
#       RADAR.  (RANGE, RANGE RATE, SHAFT AND TRUNNION ANGLES).  THE ROUTINE CALLS THE INCORP1 AND INCORP2 ROUTINES
#       WHICH COMPUTE THE DEVIATIONS AND CORRECT THE STATE VECTOR.
#
# CALLING SEQUENCE:
#       THIS ROUTINE IS PART OF P20 RENDEZVOUS NAVIGATION FOR THE LM COMPUTER ONLY.  THE ROUTINE IS ENTERED FROM
#       R22 LEM ONLY AND RETURNS DIRECTLY TO R22LEM FOLLOWING SUCCESSFUL INCORPORATION OF MEASURED DATA.  IF THE
#       COMPUTED STATE VECTOR DEVATIONS EXCEED THE MAXIMUM PERMITTED.  THE ROUTINE RETURNS TO R22LEM TO DISPLAY
#       THE DEVIATIONS.  IF THE ASTRONAUT ACCEPTS THE DATA R22LEM RETURNS TO LSR22.3 TO INCORPORATE THE
#       DEVIATIONS INTO THE STATE VECTOR.  IF THE ASTRONAUT REJECTS THE DEVIATIONS, NO MORE MEASUREMENTS ARE
#       PROCESSED FOR THIS MARK, I.E.,  R22LEM GETS THE NEXT MARK.
#
# SUBROUTINES CALLED:
#       WLINIT          LGCUPDTE        INTEGRV         INCORP1         ARCTAN
#       GETULC          RADARANG        INCORP2         NBSM            INTSTALL
#
# OUTPUT:
#       CORRECTED LM OR CSM STATE VECTOR (PERMANENT)
#       NUMBER OF MARKS INCORPORATED IN MARKCTR
#       MAGNITUDE OF POSITION DEVIATION (FOR DISPLAY) IN R22DISP METERS B-29
#       MAGNITUDE OF VELOCITY DEVIATION (FOR DISPLAY) IN R22DISP +2 M/CSEC B-7
#       UPDATED W-MATRIX
#
# ERASABLE INITIALIZATION REQUIRED:
#       LM AND CSM STATE VECTORS
#       W-MATRIX
#       MARK TIME IN MKTIME
#       RADAR RANGE IN RM METERS B-29
#               RANGE RATE IN RDOTM METERS/CSES B-7
#               SHAFT ANGLE IN RRSHAFT REVS. BO
#               TRUNNION ANGLE IN RRTRUN REVS. BO
#       GIMBAL ANGLES   INNER IN AIG
#                       MIDDLE IN AMG
#                       OUTER IN ACG
#       REFSMMAT
#       RENDWFLG
#       NOANGFLG
#       VEHUPFLG
#
# DEBRIS:
#       PUSHLIST -- ALL
#       MX, MY, MZ (VECTORS)
```

```
#       ULC, RXZ, SINTHETA, LGRET, RDRET, BVECTOR, W.IND, X78T

                BANK    13
                SETLOC  P20S3
                BANK

                EBANK=  LOSCOUNT
                COUNT*  $$/LSR22
LSR22.3         CALL
                        GRP2PC
                BON     SET
                        SURFFLAG        # ARE WE ON LUNAR SURFACE
                        LSR22.4         # YES
                        DMENFLG
                BOFF    CALL
                        VEHUPFLG
                        DOLEM
                        INTSTALL
                CLEAR   CALL            # LM PRECISION INTEGRATION
                        VINTFLAG
                        SETIFLGS
                CALL
                        INTGRCAL
                CALL
                        GRP2PC
                CALL
                        INTSTALL
                CLEAR   BOFF
                        DIMOFLAG
                        RENDWFLG
                        NOTWCSM
                SET     SET             # CSM WITH W-MATRIX INTEGRATION
                        DIMOFLAG
                        D6OR9FLG
NOTWCSM         SET     CLEAR
                        VINTFLAG
                        INTYPFLG
                SET     CALL
                        STATEFLG
                        INTGRCAL
                GOTO
                        MARKTEST
DOLEM           CALL
                        INTSTALL
                SET     CALL
                        VINTFLAG
                        SETIFLGS
                CALL
                        INTGRCAL
```

```
                CALL
                        GRP2PC
                CALL
                        INTSTALL
                CLEAR   BOFF
                        DIMOFLAG
                        RENDWFLG
                        NOTWLEM
                SET     SET             # LM WITH W-MATRIX INTEGRATION
                        DIMOFLAG
                        D6OR9FLG
NOTWLEM         CLEAR   CLEAR
                        INTYPFLG
                        VINTFLAG
                SET     CALL
                        STATEFLG
                        INTGRCAL
MARKTEST        BON     CALL            # HAS W-MATRIX BEEN INVALIDATED
                        RENDWFLG        # HAS W-MATRIX BEEN INVALIDATED
                        RANGEBQ
                        WLINIT          # YES -- REINITIALIZE
RANGEBQ         BON     EXIT            # DON'T CALL R65 IF ON SURFACE
                        SURFFLAG
                        RANGEBQ1
                CA      ZERO
                TS      R65CNTR
                TC      BANKCALL
                CADR    R65LEM
                TC      INTPRET
RANGEBQ1        AXT,2   BON             # CLEAR X2
                        0
                        LMOONFLG        # IS MOON SPHERE OF INFLUENCE
                        SETX2           # YES.  STORE ZERO IN SCALSHFT REGISTER
                INCR,2
                        2
SETX2           SXA,2   CALL
                        SCALSHFT        # 0 -- MOON.  2 -- EARTH.
                        GRP2PC
                AXT,1   SXA,1           # STORE RANGE CODE (1) FOR R3 IN NOUN 49
                        1
                        WHCHREAD
                SLOAD   SR              # GET SINGLE PRECISION RVARMIN (B-12)
                        RVARMIN         # SHIFT TO TRIPLE PRECISION (B-40)
                        28D
                RTB
                        TPMODE          # AND SAVE IN 20D
                STORE   20D
                CALL                    # BEGIN COMPUTING THE B-VECTORS, DELTAQ
                        GETULC          # B-VECTORS FOR RANGE
                BON     VCOMP           # BO, COMP. IF LM BEING CORRECTED
```

```
                        VEHUPFLG
                        +1
                STOVL   BVECTOR
                        ZEROVECS
                STORE   BVECTOR +6      # B1
                STODL   BVECTOR +12D    # B2
                        36D
                SRR*    BDSU
                        2,2             # SHIFT FROM EARTH/MOON SPHERE TO B-29
                        RM              # RM - (MAGNITUDE RCSM-RLM)
                SLR*
                        2,2             # SHIFT TO EARTH/MOON SPHERE
                STODL   DELTAQ          # EARTH B-29.  MOON B-27
                        36D             # RLC B-29/B-27
                NORM    DSQ             # NORMALIZE AND SQUARE
                        X1
                DMP     SR*
                        RANGEVAR        # MULTIPLY BY RANGEVAR (B12) THEN
                        0 -2,1          # UNNORMALIZE
                SR*     SR*
                        0,1
                        0,2
                SR*     RTB
                        0,2
                        TPMODE
                STORE   VARIANCE        # B-40
                DCOMP   TAD
                        20D             # B-40
                BMN     TLOAD
                        QOK
                        20D             # B-40
                STORE   VARIANCE
QOK             CALL
                        LGCUPDTE

                SSP     CALL
                        WHCHREAD
                DEC     2               # STORE R-RATE CODE (2) FOR R3 IN NOUN 49
                        GRP2PC
                CALL                    # B-VECTOR, DELTAQ FOR RANGE RATE
                        GETULC
                PDDL    SR*             # GET RLC SCALED B-29/B-27
                        36D             # AND SHIFT TO B-23
                        0 -4,2
                STOVL   36D             # THEN STORE BACK IN 36D
                BON     VCOMP           # B1, COMP. IF LM BEING CORRECTED
                        VEHUPFLG
                        +1
                VXSC
                        36D             # B1 = RLC (B-24/B-22)
```

```
                STOVL   BVECTOR +6
                        NUVLEM
                VSR*    VAD
                        6,2             # SHIFT FOR EARTH/MOON SPHERE
                        VCVLEM          # EARTH B-7. MOON B-5
                PDVL    VSR*            # VL TO PD6
                        NUVCSM
                        6,2             # SHIFT FOR EARTH/MOON SPHERE
                VAD     VSU
                        VCVCSM
                PDVL    DOT             # VC - VL = VLC TO PD6
                        0
                        6
                PUSH    SRR*            # RDOT B-8/B-6 TO PD12
                        2,2             # SHIFT FROM EARTH/MOON SPHERE TO B-8
                DSQ     DMPR            # RDOT**2 B-16 X RATEVAR B12
                        RATEVAR
                STORE   VARIANCE
                SLOAD   SR
                        VVARMIN         # GET SINGLE PRECISION VVARMIN (B+12)
                        16D             # SHIFT TO DP (B-4)
                STORE   24D             # AND SAVE IN 24D
                DSU     BMN             # IS MIN. VARIANCE > COMPUTED VARIANCE
                        VARIANCE
                        VOK             # BRANCH -- NO
                DLOAD                   # YES -- USE MINIMUM VARIANCE
                        24D
                STORE   VARIANCE
VOK             DLOAD   SR2             # RDOT (PD12) FROM B-8/B-6
                PDDL    SLR*            # TO B-10/B-8
                        RDOTM           # SHIFT TO EARTH/MOON SPHERE
                        0 -1,2          # B-7 TO B-10/B-8
                DSU
                DMPR
                        36D
                STOVL   DELTAQ          # B-33
                        0               # NOW GET BO
                VXV     VXV             # (ULC X VLC) X ULC
                BON     VCOMP           # BO, COMP. IF LM BEING CORRECTED
                        VEHUPFLG
                        +1
                VSR*
                        0 -2,2          # SCALED B-5
                STOVL   BVECTOR
                        ZEROVECS
                STORE   20D             # ZERO OUT 20 TO 25 IN PUSHLIST
                STOVL   BVECTOR +12D
                        BVECTOR
                ABVAL   NORM            # LOAD BO, GET MAGNITUDE AND NORMALIZE
                        20D             # SHIFT COUNT IN 20D
```

```
                VLOAD   ABVAL
                        BVECTOR +6D     # LOAD B1, GET MAGNITUDE AND NORMALIZE
                NORM    DLOAD
                        22D             # SHIFT COUNT IN 22D
                        22D             # FIND WHICH SHIFT IS SMALLER
                DSU     BMN             # BRANCH -- BO HAS A SMALLER SHIFT COUNT
                        20D
                        VOK1
                LXA,1   GOTO
                        22D             # LOAD X2 WITH THE SMALLER SHIFT COUNT
                        VOK2
VOK1            LXA,1
                        20D
VOK2            VLOAD   VSL*            # THEN ADJUST BO, B1, DELTAQ AND VARIANCE
                        BVECTOR         # WITH THI SSHIFT COUNT
                        0,1
                STOVL   BVECTOR
                        BVECTOR +6
                VSL*
                        0,1
                STODL   BVECTOR +6
                        DELTAQ
                SL*
                        0,1
                STORE   DELTAQ
                DLOAD   SL*             # GET RLC AND ADJUST FOR SCALE SHIFT
                        36D
                        0 -1,1
                DSQ     DMP             # MULTIPLY RLC**2 BY VARIANCE
                        VARIANCE
                SL4     RTB             # SHIFT TO CONFORM TO BVECTORS AND DELTAQ
                        TPMODE
                STCALL  VARIANCE        # AND STORE TP VARIANCE
                        LGCUPDTE

                CALL
                        GRP2PC
                BON     EXIT            # ARE ANGLES TO BE DONE
                        SURFFLAG
                        RENDEND         # NO
                EBANK=  AIG
MXMYMZ          CAF     AIGBANK
                TS      BBANK
                CA      AIG             # YES, COMPUTE  MX, MY, MZ
                TS      CDUSPOT
                CA      AMG
                TS      CDUSPOT +2
                CA      AOG
                TS      CDUSPOT +4      # GIMBAL ANGLES NOW IN CDUSPOT FOR TRG*NBSM
                TC      INTPRET
```

```
                VLOAD   CALL
                        UNITX
                        TRG*NBSM
                VXM     VSL1
                        REFSMMAT
                STOVL   MX
                        UNITY
                CALL
                        *NBSM*
                VXM     VSL1
                        REFSMMAT
                STOVL   MY
                        UNITZ
                CALL
                        *NBSM*
                VXM     VSL1
                        REFSMMAT
SHAFTBQ         STCALL  MZ
                        RADARANG
                SSP     CALL            # STORE SHAFT CODE (3) FOR R3 IN NOUN 49
                        WHCHREAD
                DEC     3
                        GRP2PC
                VLOAD   DOT             # COMPUTE DELTAQ,B VECTORS FOR SHAFT ANG.
                        ULC
                        MX
                SL1
                STOVL   SINTH           # 18D
                        ULC
                DOT     SL1
                        MZ
                STCALL  COSTH           # 16D
                        ARCTAN
                BDSU    DMP
                        RRSHAFT
                        2PI/8
                SL3R    PUSH
                DLOAD   SL3
                        X789
                SRR*    BDSU            # SHIFT FROM -5/-3 TO BO
                        0,2
                DMP     SRR*
                        RXZ
                        0,1             # SHIFT TO EARTH/MOON SPHERE
                STOVL   DELTAQ          # EARTH B-29.  MOON B-27
                        ULC
                VXV     VSL1
                        MY
                UNIT
                BOFF    VCOMP           # BO, COMP. IF CSM BEING CORRECTED
```

```
                                VEHUPFLG
                                +1
                        STOVL   BVECTOR
                                ZEROVECS
                        STORE   BVECTOR +6
                        STODL   BVECTOR +12D
                                RXZ
                        SR*     SRR*            # SHIFT FROM EARTH/MOON SPHERE TO B-25
                                0 -2,1
                                0,2
                        STORE   BVECTOR +12D
                        SLOAD
                                SHAFTVAR
                        DAD     DMP
                                IMUVAR          # RAD**2 B12
                                RXZ
                        SRR*    DMP
                                0,1             # SHIFT TO EARTH/MOON SPHERE
                                RXZ
                        SR*     SR*
                                0 -2,1
                                0,2
                        SR*     RTB
                                0,2
                                TPMODE          # STORE VARIANCE TRIPLE PRECISION
                        STCALL  VARIANCE        # B-40
                                LGCUPDTE

                        CALL
                                GRP2PC
        TRUNBQ          CALL
                                RADARANG
                        SSP     CALL            # STORE TRUNNION CODE (4) FOR R3 IN N49
                                WHCHREAD
                        DEC     4
                                GRP2PC
                        VLOAD   VXV
                                ULC
                                MY
                        VSL1    VXV
                                ULC
                        VSL1                    # (ULC X MY) X ULC
                        BOFF    VCOMP           # BO, COMP. IF CSM BEING CORRECTED
                                VEHUPFLG
                                +1
                        STOVL   BVECTOR
                                ZEROVECS
                        STORE   BVECTOR +6
                        STODL   BVECTOR +12D
                                RXZ
```

```
                SR*     SRR*            # SHIFT FROM EARTH/MOON SPHERE TO B-25
                        0 -2,1
                        0,2
                STORE   BVECTOR +14D
                SLOAD
                        TRUNVAR
                DAD     DMP
                        IMUVAR
                        RXZ
                SRR*    DMP
                        0,1             # SHIFT TO EARTH/MOON SPHERE
                        RXZ
                SR*     SR*
                        0 -2,1
                        0,2
                SR*     RTB
                        0,2
                        TPMODE          # STORE VARIANCE TRIPLE PRECISION
                STODL   VARIANCE
                        SINTHETA
                ASIN    BDSU            # SIN THETA IN PD6
                        RRTRUN
                DMP     SL3R
                        2PI/8
                PDDL    SL3
                        X789 +2
                SRR*    BDSU            # SHIFT FROM -5/-3 TO B0
                        0,2
                DMP     SRR*
                        RXZ
                        0,1
                STCALL  DELTAQ          # EARTH B-29.  MOON B-27
                        LGCUPDTE
                CALL
                        GRP2PC
RENDEND         GOTO
                        R22LEM93

# FUNCTIONAL DESCRIPTION:
#       LSR22.4 IS THE ENTRY TO PERFORM LUNAR SURFACE NAVIGATION FOR THE LM
#       COMPUTER ONLY.  THIS ROUTINE COMPUTES THE BE-VECTORS AND DELTA Q FOR RANGE
#       AND RANGE RATE MEASURED BY THE RENDEZVOUS RADAR
#
# SUBROUTINES CALLED:
#       INTSTALL        LGCUPDTE        INCORP1         RP-TO-R
#       INTEGRV         GETULC          INCORP2
#
# OUTPUT
#       CORRECTED CSM STATE VECTOR (PERMANENT)
#       NUMBER OF MARKS INCORPORATED IN MARKCTR
```

```
#       MAGNITUDE OF POSITION DEVIATION (FOR DISPLAY) IN R22 DISP METERS B-29
#       MAGNITUDE OF VELOCITY DEVIATION (FOR DISPLAY) IN R22DISP +2 M/CSEC B-7
#       UPDATED W-MATRIX

# ERASABLE INITIALIZATION REQUIRED
#       LM AND CSM STATE VECTORS
#       W-MATRIX
#       MARK TIME IN MKTIME
#       RADAR RANGE IN RM METERS B-29
#       RANGE RATE IN RDOTM METERS/CSEC B-7
#       VEHUPFLG

LSR22.4         CALL
                        INTSTALL
                SET     CLEAR
                        STATEFLG
                        VINTFLAG        # CALL TO GET LM POS + VEL IN REF COORD.
                CALL
                        INTGRCAL
                CALL
                        GRP2PC
                CLEAR   CALL
                        DMENFLG         # SET MATRIX SIZE TO 6X6 FOR INCORP
                        INTSTALL
                DLOAD   BHIZ            # IS THIS FIRST TIME THROUGH
                        MARKCTR
                        INITWMX6        # YES, INITIALIZE 6X6 W-MATRIX
                CLEAR   SET
                        D6OR9FLG
                        DIMOFLAG
                SET     CLEAR
                        VINTFLAG
                        INTYPFLG
                CALL
                        INTGRCAL
                GOTO
                        RANGEBQ

INITWMX6        CALL
                        WLINIT          # INITIALIZE W-MATRIX
                SET     CALL
                        VINTFLAG
                        SETIFLGS
                CALL
                        INTGRCAL
                GOTO
                        RANGEBQ

# THIS ROUTINE CLEARS RFINAL (DP) AND CALLS INTEGRV
```

```
INTGRCAL        STQ     DLOAD
                        IGRET
                        MKTIME
                STCALL  TDEC1
                        INTEGRV
                GOTO
                        IGRET

# THIS ROUTINE INITIALIZES THE W-MATRIX BY ZEROING ALL W THEN SETTING
# DIAGONAL ELEMENTS TO INITIAL STORED VALUES.

                EBANK=  W
WLINIT          EXIT
                CAF     WBANK
                TS      BBANK
                CAF     WSIZE
                TS      W.IND
                CAF     ZERO
                INDEX   W.IND
                TS      W
                CCS     W.IND
                TC      -5
                CAF     AIGBANK         # RESTORE EBANK 7
                TS      BBANK
                TC      INTPRET
                BON     SLOAD           # IF ON LUNAR SURFACE, INITIALIZE WITH
                        SURFFLAG        # WSURFPOS AND WSURFVEL INSTEAD OF
                        WLSRFPOS        # WRENDPOS AND WRENDVEL
                        WRENDPOS
                GOTO
                        WPOSTORE
WLSRFPOS        SLOAD
                        WSURFPOS
WPOSTORE        SR                      # SHIFT TO B-19 SCALE
                        5
                STORE   W
                STORE   W +8D
                STORE   W +16D
                BON     SLOAD
                        SURFFLAG
                        WLSRFVEL
                        WRENDVEL
                GOTO
                        WVELSTOR
WLSRFVEL        SLOAD
                        WSURFVEL
WVELSTOR        STORE   W +72D
                STORE   W +80D
                STORE   W +88D
                SLOAD
```

```
                                WSHAFT
                        STORE   W +144D
                        SLOAD
                                WTRUN
                        STORE   W +152D
                        SET     SSP             # SET RENDWFLG -- W-MATRIX VALID
                                RENDWFLG
                                MARKCTR         # SET MARK COUNTER EQUAL ZERO
                                0
                        RVQ

                        EBANK=  W
WBANK                   BBCON   WLINIT
                        EBANK=  AIG
AIGBANK                 BBCON   LSR22.3

# GETULC
#
# THIS SUBROUTINE COMPUTES THE RELATIVE POSITION VECTOR BETWEEN THE CSM
# AND THE LM, LEAVING THE UNIT VECTOR IN THE PUSHLIST AND MPAC AND THE
# MAGNITUDE IN 36D.

GETULC                  SETPD   VLOAD
                                0
                                DELTALEM
                        LXA,2
                                SCALSHFT        # LOAD X2 WITH SCALE SHIFT
                        VSR*    VAD
                                9D,2            # SHIFT FOR EARTH/MOON SPHERE
                                RCVLEM
                        PDVL    VSR*
                                DELTACSM
                                9D,2            # SHIFT FOR EARTH/MOON SPHERE
                        VAD     VSU
                                RCVCSM
                        RTB     PUSH            # USE NORMUNIT TO PRESERVE ACCURACY
                                NORMUNX1
                        STODL   ULC
                                36D
                        SL*                     # ADJUST MAGNITUDE FROM NORMUNIT
                                0,1
                        STOVL   36D             # ULC IN PD0 AND MPAC,RLC IN 36D
                                ULC
                        RVQ

# RADARANG
#
# THIS SUBROUTINE COMPUTS SINTHETA = -ULC DOT MY
# RXZ = (SQRT (1-SINTHETA**2))RLC
# OUTPUT
#       ULC IN ULC, PD0
```

```
#       RLC IN PD36D
#       SIN THETA IN SINTHETA AND PD6
#       RXZ NORM IN RXZ (N IN X1)

RADARANG        STQ     CALL
                        RDRET
                        GETULC
                VCOMP   DOT
                        MY
                SL1R    PUSH            # SIN THETA TO PD6
                STORE   SINTHETA
                DSQ     BDSU
                        DP1/4TH         # 1-(SIN THETA)**2
                SQRT    DMP
                        36D
                SL1     NORM
                        X1              # SET SHIFT COUNTER IN X1
                STORE   RXZ
                GOTO                    # EXIT
                        RDRET
LGCUPDTE        STQ     CALL
                        LGRET
                        INCORP1
                VLOAD   ABVAL
                        DELTAX +6
                LXA,2   SRR*
                        SCALSHFT        # 0 -- MOON.  2 -- EARTH
                        2,2             # SET VEL DISPLAY TO B-7
                STOVL   R22DISP +2
                        DELTAX
                ABVAL   SRR*
                        2,2             # SET POS DISPLAY TO B-29
                STORE   R22DISP
                SLOAD   SR
                        RMAX
                        10D
                DSU     BMN
                        R22DISP
                        R22LEM96        # GO DISPLAY
                SLOAD   DSU
                        VMAX
                        R22DISP +2      # VMAX MINUS VEL. DEVIATION
                BMN
                        R22LEM96        # GO DISPLAY
ASTOK           CALL
                        INCORP2
                GOTO
                        LGRET
IMUVAR          2DEC    E-6 B12         # RAD**2

WSIZE           DEC     161
```

```
2PI/8           2DEC    3.141592653 B-2

                EBANK=  LOSCOUNT
```

```
# PROGRAM NAME LRS24.1          RR SEARCH ROUTINE
# MCD NO. 0                     BY P. VOLANTE, SDC              DATE 1-15-67
#
# FUNCTIONAL DESCRIPTION
#
# DRIVES THE RENDEZVOUS RADAR IN A HEXAGONAL SEARCH PATTERN ABOUT THE LOS TO THE CSM (COMPUTED FROM THE CSM AND LM
# STATE VECTORS) CHECKING FOR THE DATA GOOD DISCRETE AND MONITORING THE ANGLE BETWEEN THE RADAR BORESIGHT AND THE
# LM +Z AXIS.  IF THIS ANGLE EXCEEDS 30 DEGREES THE PREFERRED TRACKING ATTITUDE ROUTINE IS CALLED TO PERFORM AN
# ATTITUDE MANEUVER.
#
# CALLING SEQUENCE -- BANKCALL FOR LRS24.1
#
# SUBROUTINES CALLED
#
#       LEMCONIC        R61LEM
#       CSMCONIC        RRDESSM
#       JOBDELAY        FLAGDOWN
#       WAITLIST        FLAGUP
#       RRNB            BANKCALL
#
# EXIT -- TO ENDOFJOB WHEN THE SEARCH FLAG (SRCHOPT) IS NOT SET
#
# OUTPUT
#
#       DATAGOOD (SP) -- FOR DISPLAY IN R1 --   00000 INDICATES NO LOCKON
#                                               11111 INDICATES LOCKON ACHIEVED
#       OMEGAD (SP)   -- FOR DISPLAY IN R2 --   ANGLE BETWEEN RR BORESIGNT VECTOR AND THE SPACECRAFT +Z AXIS
#
# ERASABLE INITIALIZATION REQUIRED:
#
#       SEARCH FLAG MUST BE SET
#       LM AND CSM STATE VECTORS AND REFSMMAT MATRIX
#
# DEBRIS
#
#       RLMSRCH         UXVECT
#       VXRLM           UYVECT
#       LOSDESRD        NSRCHPNT
#       DATAGOOD        OMEGAD
#       MPAC            PUSHLIST

                COUNT*  $$/LRS24
LRS24.1         CAF     ZERO
                TS      NSRCHPNT        # SET SEARCH PATTERN POINT COUNTER TO ZERO
CHKSRCH         CAF     BIT14           # ISSUE AUTO TRACK ENABLE TO RADAR
                EXTEND
```

```
                WOR     CHAN12
                CAF     SRCHOBIT        # CHECK IF SEARCH STILL REQUESTED
                MASK    FLAGWRD2        # (SRCHOPT FLAG SET)
                EXTEND
                BZF     ENDOFJOB        # NO-TERMINATE JOB

                CAF     6SECONDS        # SCHEDULE TASK TO DRIVE RADAR TO NEXT PT.
                INHINT
                TC      WAITLIST        # IN 6 SECONDS
                EBANK=  LOSCOUNT
                2CADR   CALLDGCH

                RELINT
                CS      RADMODES        # IS REMODE IN PROGRESS
                MASK    REMODBIT
                EXTEND
                BZF     ENDOFJOB        # YES -- WAIT SIX SECONDS
                TC      INTPRET

                RTB     DAD             # COMPUTE LOS AT PRESENT TIME + 1.5 SEC.
                        LOADTIME
                        1.5SECS
LRS24.11        STCALL  TDEC1
                        LEMCONIC        # EXTRAPOLATE LM STATE VECTOR
                VLOAD
                        RATT
                STOVL   RLMSRCH         # SAVE LEM POSITION
                        VATT
                STODL   SAVLEMV         # SAVE LEM VELOCITY
                        TAT
                STCALL  TDEC1           # EXTRAPOLATE CSM STATE VECTOR
                        CSMCONIC        # EXTRAPOLATE CSM STATE VECTOR
                VLOAD   VSU             # LOS VECTOR = R(CSM) - R(LM)
                        RATT
                        RLMSRCH
                UNIT
                STOVL   LOSDESRD        # STORE DESIRED LOS
                        VATT            # COMPUTE UNIT(V(CM) CROSS R(CM))
                UNIT    VXV
                        RATT
                UNIT
                STORE   VXRCM
                VLOAD   VSU
                        VATT
                        SAVLEMV
                MXV     VSL1            # CONVERT FROM REFERENCE TO STABLE MEMBER
                        REFSMMAT
                STORE   SAVLEMV         # VLC = V(CSM) - V(LM)
                SLOAD   BZE             # CHECK IF N=0
```

```
                NSRCHPNT
                DESGLOS         # YES -- DESIGNATE ALONG LOS
        DSU     BZE             # IS N=1
                ONEOCT          # YES -- CALCULATE X AND Y AXES OF
                CALCXY          # SEARCH PATTERN COORDINATE SYSTEM
        VLOAD                   # NO -- ROTATE X-Y AXES TO NEXT SEARCH POINT
                UXVECT
        STOVL   UXVECTPR        # SAVE ORIGINAL X AND Y VECTORS
                UYVECT          # UXPRIME = ORIGINAL UX
        STORE   UYVECTPR        # UYPRIME = ORIGINAL UY
        VXSC
                SIN60DEG        # UX = (COS 60)UXPR + (SIN 60)UYPR
        STOVL   UXVECT
                UXVECTPR
        VXSC    VAD
                COS60DEG
                UXVECT
        UNIT
        STOVL   UXVECT
                UXVECTPR        # UY = (-SIN 60)UXPR + (COS 60)UYPR
        VXSC
                SIN60DEG
        STOVL   UYVECT
                UYVECTPR
        VXSC    VSU
                COS60DEG
                UYVECT
        UNIT
        STORE   UYVECT
OFFCALC VXSC    VAD             # OFFSET VECTOR = K(UY)
                OFFSTFAC        # LOS VECTOR + OFFSET VECTOR DEFINES
                LOSDESRD        # DESIRED POINT IN SEARCH PATTERN
        UNIT    MXV
                REFSMMAT        # CONVERT TO STABLE MEMBER COORDINATES
        VSL1
CONTDESG STOVL  RRTARGET
                SAVLEMV
        STORE   LOSVEL
        EXIT
        INHINT
        TC      KILLTASK        # KILL ANY PRESENTLY WAITLISTED TASK
        CADR    DESLOOP +2      # WHICH WOULD DESIGNATE TO THE LAST
                                # POINT IN THE PATTERN
CONTDES2 CS     CDESBIT
        MASK    RADMODES        # SET BIT 15 OF RADMODES TO INDICATE
        AD      CDESBIT         # A CONTINUOUS DESIGNATE WANTED.
        TS      RADMODES
        TC      INTPRET

        CALL
```

```
                        RRDESSM          # DESIGNATE RADAR TO RRTARGET VECTOR
                EXIT
                TC      LIMALARM         # LOS NOT IN MODE 2 COVERAGE (P22)
                TC      LIMALARM         # VEHICLE MANEUVER REQUIRED (P20)

                                         # COMPUTE OMEGA,ANGLE BETWEEN RR LOS AND
                                         # SPACECRAFT +Z AXIS
OMEGCALC        EXTEND
                DCA     CDUT
                DXCH    TANGNB
                TC      INTPRET
                CALL
                        RRNB
                DLOAD   ACOS             # OMEGA IS ARCCOSINE OF Z-COMPONENT OF
                        36D              # VECTOR COMPUTED BY RRNB (LEFT AT 32D)
                STORE   OMEGDISP         # STORE FOR DISPLAY IN R2
                EXIT
                TC      ENDOFJOB
```

```
# CALCULATE X AND Y VECTORS FOR SEARCH PATTERN COORDINATE SYSTEM.

CALCXY          VLOAD   VXV
                        VXRCM
                        LOSDESRD
                UNIT
                STOVL   UXVECT          # UX = (VLM X RLM) X LOS
                        LOSDESRD
                VXV     UNIT
                        UXVECT
                STORE   UYVECT          # UY = LOS X UX
                GOTO
                        OFFCALC

DESGLOS         VLOAD   MXV             # WHEN N= 0,DESIGNATE ALONG LOS
                        LOSDESRD
                        REFSMMAT        # CONVERT LOS FROM REFERENCE TO SM COORDS
                VSL1    GOTO
                        CONTDESG

CALLDGCH        CAE     FLAGWRD0        # IS RENDEZVOUS FLAG SET
                MASK    RNDVZBIT
                EXTEND
                BZF     TASKOVER        # NO -- EXIT R24
                CAF     PRIO25          # YES -- SCHEDULE JOB TO DRIVE RADAR TO NEXT
                TC      FINDVAC         # PONT IN SEARCH PATTERN
                EBANK=  RLMSRCH
                2CADR   DATGDCHK

                TC      TASKOVER

DATGDCHK        CAF     BIT4
                EXTEND                  # CHECK IF DATA GOOD DISCRETE PRESENT
                RAND    CHAN33
                EXTEND
                BZF     STORE1S         # YES -- GO TO STORE 11111 FOR DISPLAY IN R1
                CS      SIX
                AD      NSRCHPNT        # IS N GREATER THAN 6
                EXTEND
                BZF     LRS24.1         # YES -- RESET N = 0 AND START AROUND AGAIN
                INCR    NSRCHPNT        # NO -- SET N = N+1 AND GO TO
                TCF     CHKSRCH         # NEXT POINT IN PATTERN

STORE1S         CAF     ALL1S           # STORE 11111 FOR DISPLAY IN R1
                TS      DATAGOOD
```

```
                INHINT
                TC      KILLTASK        # DELETE DESIGNATE TASK FROM
                CADR    DESLOOP +2      # WAITLIST USING KILLTASK
                TC      ENDOFJOB

LIMALARM        TC      ALARM           # ISSUE ALARM 527 -- LOS NOT IN MODE2
                OCT     527             # COVERAGE IN P22 OR VEHICLE MANEUVER
                INHINT                  # REQUIRED IN P20
                TC      KILLTASK        # KILL WAITLIST CALL FOR NEXT
                CADR    CALLDGCH        # POINT IN SEARCH PATTERN
                TC      ENDOFJOB

ALL1S           DEC     11111
SIN60DEG        2DEC    .86603
COS60DEG        =       DPHALF          # (2DEC .50)
UXVECTPR        EQUALS  12D             # PREVIOUS
UYVECTPR        EQUALS  18D
RLMUNIT         EQUALS  12D
OFFSTFAC        2DEC    0.05678         # TANGENT OF 3.25 DEGREES

ONEOCT          OCT     00001           # **** NOTE -- THESE TWO CONSTANTS MUST ****
3SECONDS        2DEC    300             # **** BE IN THIS ORDER BECAUSE         ****

                                        # **** ONEOCT NEEDS A LOWER ORDER       ****
                                        # **** WORD OF ZEROES                   ****
6SECONDS        DEC     600
1.5SECS         2DEC    150

ZERO/SP         EQUALS  HI6ZEROS

                BLOCK   02
                SETLOC  FFTAG5
                BANK
                COUNT*  $$/P20
GOTOV56         EXTEND                  # P20 TERMINATES BY GOTOV56 INSTEAD OF
                DCA     VB56CADR        # GOTOPOOH
                TCF     SUPDXCHZ
                EBANK=  WHOCARES
VB56CADR        2CADR   TRMTRACK
```

```
# PROGRAM NAME: R29     (RENDEZVOUS RADAR DESIGNATE DURING POWERED FLIGHT)
# MOD NO. 2     BY H. BLAIR-SMITH       JULY 2, 1968
#
# FUNCTIONAL DESCRIPTION:
#
#       DESIGNATES THE RENDEZVOUS RADAR TOWARD THE COMPUTES LOS TO THE CSM, WITH THE CHIEF OBJECTIVE OF OBTAINING RANGE
#       AND RANGE RATE DATA AT 2-SECOND INTERVALS FOR TRANSMISSION TO THE GROUND.  WHEN THE RR IS WITHIN .5 DEGREE OF
#       THE COMPUTED LOS, TRACKING IS ENABLED, AND DESIGNATION CONTINUES UNTIL THE DATA-GOOD DISCRETE IS RECEIVED.  AT
#       THAT POINT, DESIGNATION CEASES AND A RADAR-READING ROUTINE TAKES OVER, PREPARING A CONSISTENT SET OF DATA FOR
#       DOWN TELEMETRY.  THE SET INCLUDES RANGE, RANGE RATE, MARK TIME, TWO RR CDU ANGLES, THREE IMUCDU ANGLES, AND AN
#       INDICATOR WHICH IS 1 WHEN THE SET IS CONSISTENT AND 0 OTHERWISE.  THE INDICATOR IS IN TRKMKCNT.
#
# CALLING SEQUENCE:  BEGUN EVERY 2 SECONDS AS AN INTEGRAL PART OF SERVICER
#
# SUBROUTINES CALLED:
#
#       REMODE          RRPONLY
#       UNIT            MPACVBUF
#       QUICTRIG        AX*SR*T
#       SPSIN           SPCOS
#       SETRRECR        RROUT
#       RRRDOT          RRRANGE
#
# EXIT:  TO NOR29NOW, IN SERVICER.
#
# OUTPUT:  (ALL FOR DOWNLINK)
#
#       RM              RDOTM           (RAW)
#       AIG             AMG
#       AOG             TRKMKCNT        TRKMKCNT = 00001 IF SET IS CONSISTENT,
#       TANGNB          TANGNB +1       OTHERWISE TRKMKCNT = 00000.
#       MKTIME
```

```
#
# ERASABLE INITIALIZATION REQUIRED:
#
#       NOR29FLG        READRFLG                (TO 1 AND 0 BY FRESH START) (RESET NOR29FLG TO LET SERVICER RUN R29)
#       PIPTIME         RADMODES (BIT 10)       (BIT SET TO 0 BY FRESH START)
#       R(CSM)          V(CSM)
#       R               V                       (PIPTIME THRU V BY AVE G IN SERVICER)
#
# DEBRIS:
#
#       RADMODES (BIT 10)
#       LOSSM           LOSVDT/4                (= RRTARGET & LOSVEL)
#       SAVECDUT        OLDESFLG                (SAVECDUT = MLOSV)
#       LOSCMFLG        READRFLG
#
# ALARMS:  NONE.
#
# COMPONENT JOBS AND TASKS:
#
#       INITIALIZING, IF RR IS FOUND TO BE IN MODE 1:  JOB R29REMOJ AND TASK REMODE:  ALWAYS: TASK PREPOS29.
#       DESIGNATING:  TASK BEGDES29 & JOB R29DODES.
#       RADAR READING:  TASK R29READ AND JOB R29RDJOB.  ALL JOBS ARE NOVAC TYPE.

                BANK    33
                SETLOC  R29/SERV
                BANK

                COUNT*  $$/R29

NR29&RDR        EQUALS  EBANK5
```

```
# SERVICER COMES TO R29 FROM "R29?" IF NOR29FLG, READRFLG, RRREMODE, RRCDUZRO, RRREPOS, AND DISPLAY-INERTIAL-DATA
# ARE ALL RESET, AND THE RR IS IN LGC MODE (OFTEN CONFUSINGLY CALLED AUTO MODE).

R29             CS      RADMODES
                MASK    DESIGBIT
                EXTEND
                BZF     R29.LOS         # BRANCH IF DESIGNATION IS ALREADY ON.

                INHINT
                ADS     RADMODES        # SHOW THAT DESIGNATION IS NOW ON.
                CS      BIT14
                EXTEND
                WAND    CHAN12          # REMOVE RR TRACK ENABLE DISCRETE.
                CS      LOSCMBIT
                MASK    FLAGWRD2
                TS      FLAGWRD2        # CLEAR LOSCMFLG TO SHOW DES. LOOP IS OFF.
                CS      OLDESBIT
                MASK    STATE
                TS      STATE           # SHOW THAT DES. LOOP IS NOT REQUESTED.
                TC      BANKCALL
                CADR    SETRRECR        # ENABLE RR ERROR COUNTERS.
                CA      ANTENBIT
                MASK    RADMODES
                CCS     A               # TEST RR MODE BIT.
                TCF     SETPRPOS        # MODE 2.

                CA      PRIO21          # MODE 1:  MUST REMODE.
                TC      NOVAC
                EBANK=  LOSCOUNT
                2CADR   R29REMOJ        # NEEDS OWN JOB TO RADSTALL IN.

                CS      DESIGBIT
                MASK    RADMODES        # CLEAR DESIGNATE FLAG IN RADMODES
                TS      RADMODES        # BEFORE CALLING REMODE
                CA      REMODBIT
                ADS     RADMODES        # SHOW THAT REMODING IS ON.
                TCF     NOR29NOW        # CONTINUE SERVICER FUNCTIONS.

SETPRPOS        CA      ONE
                TC      WAITLIST
                EBANK=  LOSCOUNT
                2CADR   PREPOS29        # TASK TO SET TRUNNION ANGLE TO 180 DEG.

                CA      REPOSBIT
                ADS     RADMODES        # SHOW THAT REPOSITIONING IS ON.
                TCF     NOR29NOW
```

```
# FORCE RENDEZVOUS RADAR INTO MODE 2.

R29REMOJ        CA      ONE
                TC      WAITLIST
                EBANK=  LOSCOUNT
                2CADR   REMODE          # REMODE MUST RUN AS A TASK.

                TC      BANKCALL        # WAIT FOR END OF REMODING
                CADR    RADSTALL

                TCF     ENDOFJOB        # BAD EXIT CAN'T HAPPEN.
                TCF     ENDOFJOB

# TASK TO PREPOSITION THE RR TRUNNION ANGLE TO -180 DEG.

                SETLOC  R29S1
                BANK

PREPOS29        CA      NEGMAX          # -180 DEG.
                TC      RRTONLY         # DRIVE TRUNNION CDU.
                CS      REPOSBIT        # SHOW THAT REPOSITIONING IS OFF.
                MASK    RADMODES
                TS      RADMODES
                TCF     TASKOVER

# COMPUTE THE LINE-OF-SIGHT AND LOS VELOCITY, AND PASS THEM TO THE R29DODES LOOP.

                SETLOC  R29
                BANK

R29.LOS         EXTEND
                DCS     PIPTIME
                DXCH    MPAC
                EXTEND
                DCA     TIME2
                DAS     MPAC            # (MPAC) = T-PIPTIME, SCALED B-28.
                TS      MODE            # SET MODE TO DOUBLE PRECISION.
                CA      MPAC +1
                EXTEND
                MP      BIT12
                DXCH    MPAC            # T-PIPTIME NOW SCALED B-17.
                TC      INTPRET
```

```
# LOSCMFLG = 0 MEANS THAT THE DESIGNATION IS READY FOR NEW DATA.  SETTING LOSCMFLG MAKES IT GO AWAY SO SETUP29D CAN
# START IT UP WHEN THE DATA IS IN PLACE.

                PDVL    VSU             # PUSH DOWN T-PIPTIME
                        V(CSM)
                        V               # LOSVEL = V(CSM) - V
                PDDL    VXSC            # SWAP LOSVEL FOR T-PIPTIME, MULTIPLY THEM
                VAD     VSU             #       AND ADD THE RESULT TO R(CSM) - R TO GET
                        R(CSM)          #       AN UP-TO-DATE LOS VECTOR IN SM AXES.
                        R
                BOFSET  EXIT            # (BOFSET DOES ITS THING INHINTED.)
                        LOSCMFLG        # IF DESIGNATE LOOP IS OFF, CHANGE LOSCM-
                        SETUP29D        # FLG TO ON AND GO TO SET UP NEW DATA.
                TCF     NOR29NOW        # IF DES. LOOP IS ON, LET IT USE OLD DATA.

SETUP29D        STOVL   LOSSM           # LINE-OF-SIGHT VECTOR, STABLE MEMBER AXES
                        0
                VXSC
                        .5SECB17
                STORE   LOSVDT/4        # 1/2 SECOND'S WORTH OF LOS VELOCITY.
                CLEAR   EXIT
                        LOSCMFLG        # LET R29DLOOP USE NEW DATA.

                CS      STATE
                MASK    OLDESBIT
                EXTEND
                BZF     NOR29NOW        # BRANCH IF R29 DES. LOOP IS REQUESTED.
                INHINT
                ADS     STATE           # OTHERWISE REQUEST IT NOW.

                CCS     PIPCTR          # SEE IF TASK SHOULD BE OFFSET ONE SECOND.
                CS      SUPER110        # -96D +100D = 4.
                AD      1SEC            # 0 +100D = 100D.
                TC      WAITLIST
                EBANK=  LOSCOUNT
                2CADR   BEGDES29        # START BEGDES29 TASK ASAP.

                TCF     NOR29NOW        # RELINT AND CONTINUE SERVICER FUNCTIONS.

.5SECB17        2DEC    50 B-17
```

```
# R29 DESIGNATE JOB AND TASK MACHINERY.  TASK RECURS EVERY .5 SEC UNTIL DESIGNATE IS CALLED OFF; IT MAY WAIT FOR A
# CENTISECOND OR TWO IF IT COMES UP WHILE SETUP29D IS SUPPLYING NEW DATA.

                BANK    24
                SETLOC  P20S
                BANK

                COUNT*  $$/R29

BEGDES29        CAF     PRIO21
                TC      NOVAC
                EBANK=  LOSVDT/4
                2CADR   R29DODES        # START R29DODES JOB TWICE A SECOND.

R29DLOOP        CAF     .5SEC
                TC      VARDELAY

                CS      RADMODES
                MASK    DESIGBIT
                CCS     A
                TCF     TASKOVER        # QUIT IF DESIGNATION IS CALLED OFF.

                CS      FLAGWRD2
                MASK    LOSCMBIT
                EXTEND
                BZF     +3              # BRANCH IF SETUP29D'S SUPPLYING NEW DATA.
                ADS     FLAGWRD2        # SET LOSCMFLG:  SHOW THAT DES. LOOP IS ON.
                TCF     BEGDES29

                CA      ONE
                TCF     R29DLOOP +1     # WAIT A CENTISECOND FOR NEW DATA.
```

```
# R29DODES:  RR DESIGNATION LOOP FOR R29
#
# THIS ROUTINE DOES MUCH THE SAME THING AS DODES, BUT A GREAT DEAL FASTER.  IT TAKES THE NON-UNITIZED LOS VECTOR
# IN STABLE MEMBER COORDINATES (LOSSM) AND A DELTA-LOS IN SM AXES (LOSVDT/4) WHICH IS 1/2 SEC TIMES LOS VELOCITY,
# AND DEVELOPS THE SHAFT AND TRUNNION COMMANDS USING SINGLE PRECISION AS MUCH AS POSSIBLE, AND INTERPRETIVE NOT AT
# ALL.  THE UNIT(LOSM + LOSVEL * 1 SEC) IS COMPUTED IN DP AND TRANSFORMED TO NAV BASE COORDINATES IN DOUBLE PRE-
# CISION (USING SP SINES AND COSINES OF CDU ANGLES), AND THE REST IS DONE IN SP.
#
# THE FUNCTIONAL DIFFERENCE IS THAT R29DODES ALWAYS CLEARS LOSCMFLG WHEN IT ENDS, AND IT STARTS UP THE R29READ
# TASK WHEN LOCK-ON IS ACHIEVED.

                BANK    32
                SETLOC  F2DPS*32
                BANK

                COUNT*  $$/R29
                EBANK=  LOSVDT/4

R29DODES        CA      ONE
                TS      TANG            # INDICATE 1ST PASS THRU VECTOR LOOP.
                CA      FIVE

R29DVBEG        CCS     A               # COUNT DOWN BY TWO'S IN VECTOR LOOP.
                TS      Q
                CCS     TANG
                TCF     R29DPAS1        # DO THIS ON 1ST PASS THRU LOOP.

                EXTEND                  # (A "PASS" HERE MEANS 3 TIMES AROUND).
                INDEX   Q
                DCA     LOSVDT/4
                INDEX   Q
                DAS     LOSSM           # ADVANCE LOS VECTOR 1/2 SECOND.

R29DPAS1        EXTEND
                INDEX   Q
                DCA     LOSSM
                INDEX   Q               # MOVE CURRENT LOS (1ST PASS) OR LOS PRO-
                DXCH    MPAC +1         # JECTED 1/2 SEC AHEAD (2ND PASS).
                CCS     TANG
                TCF     R29DVEND        # BUG OUT HERE IN 1ST PASS.

                EXTEND
                INDEX   Q
                DCA     LOSVDT/4
                INDEX   Q
                DAS     MPAC +1         # PROJECT LOS 1 SECOND AHEAD (2ND PASS).

R29DVEND        CCS     Q
                TCF     R29DVBEG        # BRANCH TO CONTINUE VECTOR LOOP.
```

```
# UNITIZE AND TRANSFORM TO NAV BASE AXES THE PRESENT LOS (1ST PASS) OR THE 1-SEC PROJECTED LOS (2ND PASS).

                DXCH    MPAC +1
                DXCH    MPAC
                CA      R29FXLOC        # = ADRES INTB15 + -34D
                TS      FIXLOC
                TC      USPRCADR        # WITH FIXLOC ARMED FOR LENGTH AND LENGTH
                CADR    UNIT            # SQUARED, BORROW UNITIZING ROUTINE.
                TC      MPACVBUF        # MOVE UNIT(LOS) TO AX*SR*T ARG AREA.

                CCS     TANG
                TCF     +2
                TCF     GOTANGLS        # GET CDU ANGLES ONLY AFTER 1ST PASS.
                INHINT                  # ENSURE CONSISTENT CDU READINGS.
                EXTEND
                DCA     CDUT
                DXCH    SAVECDUT        # TRUNNION AND SHAFT ANGLES.
                CA      CDUY
                TS      CDUSPOT
                CA      CDUZ
                TS      CDUSPOT +2
                CA      CDUX
                TS      CDUSPOT +4      # CDU ANGLES IN FUNNY ORDER FOR AX*SR*T.
                TC      BANKCALL
                CADR    QUICTRIG        # GET SINES AND COSINES OF CDU ANGLES.

GOTANGLS        CS      THREE
                TC      BANKCALL
                CADR    AX*SR*T         # TRANSFORM UNIT LOS TO NB AXES (ULOSNB).

                CCS     TANG
                TCF     +2
                TCF     R29DPAS2        # GO TO RR COMMAND COMP. AFTER 2ND PASS.
```

```
# COMPUTE COSINE OF THE ANGLE BETWEEN THE PRESENT LOS AND THE RR BORESIGHT VECTOR, AND SET THE SELFTRACK ENABLE IF
# THE COSINE IS APPROXIMATELY COS(.5 DEG) OR GREATER (I.E., SMALLER ANGLE).

                INHINT
                TS      TANG            # INDICATE 2ND PASS THRU VECTOR LOOP.
                CA      SAVECDUT
                TC      SPCOS
                TS      PUSHLOC         # PUSHLOC = COS T.
                CS      SAVECDUT
                TC      SPSIN
                TS      MODE            # MODE = -SIN T.
                EXTEND
                MP      VBUF +2         # FORM - SIN T ULOSNBY.
                DXCH    MPAC
                CA      SAVECDUT +1
                TC      SPSIN
                TS      SAVECDUT        # SAVECDUT NOW = SIN S.
                EXTEND
                MP      PUSHLOC
                EXTEND
                MP      VBUF            # FORM SIN S COS T ULOSNBX.
                DAS     MPAC
                CA      SAVECDUT +1
                TC      SPCOS
                TS      SAVECDUT +1     # SAVECDUT +1 NOW = COS S.
                EXTEND
                MP      PUSHLOC
                EXTEND
                MP      VBUF +4         # FORM COS S COS T ULOSNBZ.
                DAS     MPAC            # COS(ERROR) = ULOSNB . (SIN S COS T,
                EXTEND                  # -SIN T, COS S COS T).
                DCA     MPAC
TESTCOS         DAS     MPAC            # (ULOSNB IN VBUF WAS A HALF-UNIT VECTOR).
                CCS     A               # TEST FOR + OVERFLOW, NONE, OR MINUS.
                CA      BIT14
                NOOP
                EXTEND
                WOR     CHAN12          # IF PLUS OVERFLOW, SET SELFTRACK ENABLE.
                RELINT
                TCF     R29DVBEG -1     # MAKE 2ND PASS THRU VECTOR LOOP.
```

```
# COMPUTE SHAFT AND TRUNNION COMMANDS TO NULL HAVE THE ERROR IN HALF A SECOND.

R29DPAS2        CA      SAVECDUT +1
                EXTEND
                MP      VBUF            # FORM COS S ULOSNB'X.
                DXCH    TANG
                CS      SAVECDUT
                EXTEND
                MP      VBUF +4         # FORM - SIN S ULOSNB'Z.
                DAS     TANG            # RAW SHAFT CMD = ULOSNB' . (COS S, 0,
                CS      MODE            # - SIN S)
                EXTEND
                MP      SAVECDUT
                EXTEND
                MP      VBUF            # FORM SIN T SIN S ULOSNB'X.
                DXCH    MPAC
                CA      PUSHLOC
                EXTEND
                MP      VBUF +2         # FORM COS T ULOSNB'Y.
                DAS     MPAC
                CS      MODE
                EXTEND
                MP      SAVECDUT +1
                EXTEND
                MP      VBUF +4         # FORM SIN T COS S ULOSNB'Z.
                DAS     MPAC            # RAW TRUNNION CMD = ULOSNB'.
                CA      MPAC            # (SIN S SIN T, COS T, SIN S COS T).
                EXTEND
                MP      RR29GAIN
                XCH     TANG            # STORE REFINED T CMD, GET RAW S CMD.
                EXTEND
                MP      RR29GAIN
                TS      TANG +1         # STORE REFINED S CMD.
```

```
# WHETHER OR NOT TRACKING WAS ENABLED THIS TIME, CHECK ON RR DATA-GOOD.  IF PRESENT, STOP DESIGNATING AND START
# READING DATA FROM THE RENDEZVOUS RADAR.

DGOOD?          CAF     BIT4
                EXTEND
                RAND    CHAN33          # GET RR DATA-GOOD BIT.
                INHINT                  # (MAINLY FOR RROUT).
                EXTEND
                BZF     R29LOKON        # BRANCH IF DATA-GOOD IS PRESENT.

                TC      BANKCALL
                CADR    RROUT           # DATA-GOOD IS ABSENT, SO SEND COMMANDS.
                TCF     END29DOD

R29LOKON        CS      DESIGBIT
                MASK    RADMODES
                TS      RADMODES        # SHOW THAT DESIGNATION IS OVER.
                CS      BIT2
                EXTEND
                WAND    CHAN12          # DISABLE RR ERROR COUNTERS.
                CA      READRBIT
                ADS     FLAGWRD3        # SHOW THAT READING HAS BEEN REQUESTED.
                CCS     PIPCTR          # SEE IF TASK SHOULD BE OFFSET 1 SEC.
                CS      SUPER110        # -96D + 100D = 4.
                AD      1SEC            # 0 + 100D = 100D.
                TC      WAITLIST
                EBANK=  LOSCOUNT
                2CADR   R29READ         # START READING TASK AND JOB.

END29DOD        CS      LOSCMBIT
                MASK    FLAGWRD2
                TS      FLAGWRD2        # ALWAYS CLEAR LOSCMFLG.
                TCF     ENDOFJOB

R29FXLOC        ADRES   INTB15+ -34D
RR29GAIN        DEC     -.53624
LOSVDT/4        EQUALS  LOSVEL
LOSSM           EQUALS  RRTARGET
SAVECDUT        EQUALS  MLOSV
```

```
# RR READING IS SET UP BY R29DODES WHEN IT DETECTS RR LOCK-ON

                BANK    24
                SETLOC  P20S
                BANK

                COUNT*  $$/R29

                EBANK=  LOSCOUNT

R29READ         CAF     PRIO26          # CALLED BY WAITLIST
                TC      NOVAC
                EBANK=  LOSCOUNT
                2CADR   R29RDJOB        # START JOB TO READ AND DOWNLINK FOR R29.

                CA      2SECS
                TC      VARDELAY

                CA      FLAGWRD3        # 2 SECONDS LATER, SEE IF READING IS STILL
                MASK    READRBIT        # ALLOWED (NO TRACKER FAIL ETC.)
                CCS     A
                TCF     R29READ         # IT'S OK:  CALL IT AGAIN.
                TCF     TASKOVER        # IT AIN'T:  WAIT FOR REDESIGNATE.

R29RDJOB        CA      FLAGWRD3        # CALLED VIA NOVAC.
                MASK    NR29FBIT
                CCS     A               # TEST "NOR29FLG".
                TCF     ENDRRD29        # R29 IS NOW OVER, STOP AT ONCE.

                CA      RADMODES
                MASK    AUTOMBIT
                CCS     A               # TEST RR-NOT-IN-AUTO-MODE BIT.
                TCF     ENDRRD29        # ASTRO TOOK RR OUT OF AUTO MODE.

                TC      BANKCALL
                CADR    RRRDOT          # INITIATE READING OF RANGE RATE.
                TC      BANKCALL
                CADR    RADSTALL        # GOT TO SLEEP UNTIL IT'S READY.
                TCF     ENDRRD29        # BAD READ; REDESIGNATE.
```

```
# R29 RADAR READING CONTINUED.

                DXCH    TIMEHOLD
                DXCH    MPAC            # TIME OF RR READING, FOR DOWNLINK.
                INHINT                  # BE SURE OF 5 CONSISTENT CDU ANGLES.
                EXTEND
                DCA     CDUT
                DXCH    MPAC +2         # RRCDU ANGLES AT RR READ, FOR DOWNLINK.
                EXTEND
                DCA     CDUY
                DXCH    MPAC +4         # MPAC'S 7 WORDS ARE BUFFERED FOR COPYCYCLE.
                CA      CDUX
                TS      MPAC +6         # IMUCDU ANGLES AT RR READ, FOR DOWNLINK.

R29RANGE        TC      BANKCALL
                CADR    RRRANGE         # INITIATE READING OF RR RANGE.
                TC      BANKCALL
                CADR    RADSTALL        # GO TO SLEEP UNTIL IT'S READY.
                TCF     R29RRR?         # BAD READ OR SCALE CHANGE ... WHICH?

                INHINT
                DXCH    DNRRANGE        # COPYCYCLE TO LAY OUT NEW R29 DOWNLINK.
                DXCH    RM
                DXCH    MPAC
                DXCH    MKTIME
                DXCH    MPAC +2
                DXCH    TANGNB
                DXCH    MPAC +4
                DXCH    AIG
                CA      MPAC +6
                TS      AOG
                CA      ONE
                TS      TRKMKCNT        # SHOW THAT DOWNLINK DATA IS CONSISTENT.
                TCF     ENDOFJOB

R29RRR?         CS      FLAGWRD5
                MASK    BIT10
                CCS     A               # WAS IT A SCALE CHANGE (REAL OR PHONY)?
                TCF     ENDRRD29        # NO, A BAD READ; REDESIGNATE.
                TC      DOWNFLAG
                ADRES   RNGSCFLG
                TCF     R29RANGE        # YES; CLEAR FLAG AND READ AGAIN.

ENDRRD29        CA      ZERO            # TROUBLE MADE US COME HERE TO LEAVE THE
                TS      TRKMKCNT        # RR-READING MODE.  DISCREDIT DOWNTEL.
                TC      DOWNFLAG
                ADRES   READRFLG
                CS      BIT14
                EXTEND
```

```
          WAND    CHAN12          # REMOVE TRACK-ENABLE DISCRETE.
          TCF     ENDOFJOB
```

```
# W-MATRIX MONITOR

                BANK    31
                SETLOC  VB67
                BANK
                COUNT*  $$/EXTVB

                EBANK=  WWPOS

V67CALL         TC      INTPRET
                CALL
                        V67WW
                EXIT
                EXTEND                  # SAVE THE PRESENT N99 VALUES FOR
                DCA     WWPOS           # COMPARISON AFTER THE DISPLAY
                DXCH    WWBIAS +2
                EXTEND
                DCA     WWVEL
                DXCH    WWBIAS +4
                EXTEND
                DCA     WWBIAS
                DXCH    WWBIAS +6
V06N99DS        CAF     V06N99
                TC      BANKCALL
                CADR    GOXDSPF
                TCF     ENDEXT
                TCF     V6N99PRO
                TCF     V06N99DS
V6N99PRO        ZL
                CA      FIVE
N99LOOP         TS      Q
                INDEX   Q
                CS      WWPOS
                INDEX   Q
                AD      WWPOS +6
                ADS     L
                CCS     Q               # THE SUM OF ALL DIFFERENCES MUST BE ZERO.
                TCF     N99LOOP
                LXCH    A
                EXTEND
                BZF     V06N9933
                TC      UPFLAG
                ADRES   V67FLAG

V06N9933        TC      INTPRET
                BON     EXIT
                        V67FLAG
                        +2
                TCF     ENDEXT
                DLOAD
```

```
                          WWPOS
                SL4       SL1
                STODL     0D
                          WWVEL
                STODL     2D
                          WWBIAS
                SL                          # SHIFT FROM NOUN SCALING (B-5) TO
                          10D               # INTERNAL SCALING (B+5)
                STORE     4D
                BON       LXA,1
                          SURFFLAG
                          V67SURF
                          0D
                SXA,1     LXA,1
                          WRENDPOS
                          2D
                SXA,1     GOTO
                          WRENDVEL
                          V67CLRF
V67SURF         LXA,1     SXA,1
                          0D
                          WSURFPOS
                LXA,1     SXA,1
                          2D
                          WSURFVEL
V67CLRF         LXA,1     SXA,1
                          4D
                          WTRUN
                SXA,1
                          WSHAFT
                CLEAR     EXIT
                          RENDWFLG
                TCF       ENDEXT
V67WW           STQ       BOV
                          S2
                          +1
                CLEAR     CALL
                          V67FLAG
                          INTSTALL
                SSP       DLOAD
                          S1
                DEC       6
                          ZEROVECS
                STORE     WWPOS
                STORE     WWVEL
                STORE     WWBIAS
                AXT,1
                DEC       54
NXPOSVEL        VLOAD*    VSQ
                          W +54D,1
```

```
                        GOTO
                                ADDPOS
V06N99                  VN      0699

                        SETLOC  VB67A
                        BANK
                        COUNT*  $$/EXTVB

ADDPOS                  DAD
                                WWPOS
                        STORE   WWPOS
                        VLOAD*  VSQ
                                W +108D,1
                        DAD
                                WWVEL
                        STORE   WWVEL
                        VLOAD*  VSQ
                                W +162D,1
                        DAD
                                WWBIAS
                        STORE   WWBIAS
                        TIX,1   SQRT
                                NXPOSVEL
                        SR                      # SHIFT FROM INTERNAL SCALING (B+5) TO
                                10D             # NOUN SCALING (B-5)
                        STODL   WWBIAS
                                WWVEL
                        SQRT
                        STODL   WWVEL
                                WWPOS
                        SQRT
                        STORE   WWPOS
                        BOV     GOTO
                                +2
                                V67XXX
                        DLOAD
                                DPPOSMAX
                        STORE   WWPOS
                        STORE   WWVEL
                        STORE   WWBIAS
V67XXX                  DLOAD   DSU
                                WWPOS
                                FT99999
                        BMN     DLOAD
                                +3
                                FT99999
                        STORE   WWPOS
                        LXA,1   SXA,1
                                S2
                                QPRET
```

```
                EXIT
                TC          POSTJUMP
                CADR        INTWAKE

FT99999         2DEC        30479 B-19
```

```
                BANK    25
                SETLOC  RADARUPT
                BANK
                COUNT*  $$/RRUPT

                EBANK=  LOSCOUNT

R12LITES        CA      ONE
                MASK    IMODES33
                CCS     A
                TCF     ISWRETRN

                TC      HLIGHT
                TC      HLIGHT -3
                TCF     ISWRETRN

RADLITES        CA      BIT1
                MASK    IMODES33
                CCS     A
                TC      Q

                CS      BIT5
                AD      ITEMP1
                CCS     A
                CS      ONE
                TCF     VLIGHT

                TCF     RRTRKF

HLIGHT          TS      ITEMP5          # ZERO ITEMP5 FOR H INDEX

                CA      HLITE
                TS      L

                CA      FLGWRD11
                MASK    SCABBIT
                CCS     A
                TCF     ONLITES
                CA      LRALTBIT
BOTHLITS        MASK    RADMODES
                CCS     A
                TCF     ONLITES

                CA      FLGWRD11
                INDEX   ITEMP5
                MASK    HFLSHBIT
                CCS     A
                TCF     RRTRKF
```

```
LITIT           EXTEND
                QXCH    ITEMP6
                TC      TRKFLON +1

                EXTEND
                QXCH    ITEMP6
                TCF     RRTRKF

ONLITES         INDEX   ITEMP5
                CS      HFLSHBIT
                MASK    FLGWRD11
                TS      FLGWRD11

                CA      L
                TCF     LITIT
VLIGHT          TS      ITEMP5
                CA      VLITE
                TS      L
                CA      BIT8
                TCF     BOTHLITS

HLITE           EQUALS  BIT5
VLITE           EQUALS  BIT3

# *** END OF LEMP20S .127 ***
```

```
# PROGRAM DESCRIPTION P30       DATE 3-6-67
#
# MOD.1 BY RAMA AIYAWAR
# FUNCTIONAL DESCRIPTION
#       ACCEPT ASTRONAUT INPUTS OF TIG.DELV(LV)
#       CALL IMU STATUS CHECK ROUTINE (R02)
#       DISPLAY TIME TO GO, APOGEE, PERIGEE, DELV(MAG), MGA AT IGN
#       REQUEST BURN PROGRAM
#
# CALLING SEQUENCE VIA JOB FROM V37
#
# EXIT VIA V37 CALL OR TO GOTOPOOH (V34E)
#
# SUBROUTINE CALLS-FLAGUP, PHASCHNG, BANKCALL, ENDOFJOB, GOFLASH, GOFLASHR
#                   GOPERF3R, INTPRET, BLANKET, GOTOPOOH, R02BOTH, S30.1,
#                   TIG/N35, MIDGIM, DISPMGA
#
# ERASABLE INITIALIZATION- STATE VECTOR
#
# OUTPUT-RINIT, VINIT, +MGA, VTIG, RTIG, DELVSIN, DELVSAB, DELVSLV, HAPO,
#         HPER, TTOGO
#
# DEBRIS- A,L, MPAC, PUSHLIST

                BANK    32
                SETLOC  P30S
                BANK
                EBANK=  +MGA
                COUNT*  $$/P30
P30             TC      UPFLAG          # SET UPDATE FLAG
                ADRES   UPDATFLG
                TC      UPFLAG          # SET TRACK FLAG
                ADRES   TRACKFLG

P30N33          CAF     V06N33          # T OF IGN
                TC      VNPOOH          # RETURNS ON PROCEED, POOH ON TERMINATE

                CAF     V06N81          # DISPLAY DELTA V (LV)
                TC      VNPOOH          #     REDISPLAY ON RECYCLE

                TC      DOWNFLAG        # RESET UPDATE FLAG
                ADRES   UPDATFLG
                TC      INTPRET
                CALL
                        S30.1
                SET     EXIT
                        UPDATFLG
PARAM30         CAF     V06N42          # DISPLAY APOGEE,PERIGEE ,DELTA V
                TC      VNPOOH
```

```
                TC      INTPRET
                SETGO
                        XDELVFLG        # FOR P40'S: EXTERNAL DELTA-V GUIDANCE.
                        REVN1645        # TRKMKCNT, T60, +MGA  DISPLAY

V06N33          VN      0633
V06N42          VN      0642
```

```
# PROGRAM DESCRIPTION S30.1     DATE 9NOV66
# MOD NO 1                      LOG SECTION P30,P37
# MOD BY RAMA AIYAWAR **
# FUNCTIONAL DESCRIPTION
#       BASED ON STORED TARGET PARAMETERS(R OF IGNITION(RTIG),V OF
#       IGNITION(VTIG),TIME OF IGNITION (TIG)),COMPUTE PERIGEE ALTITUDE
#       APOGEE ALTITUDE AND DELTAV REQUIRED(DELVSIN).
# CALLING SEQUENCE
#       L       CALL
#       L+1             S30.1
# NORMAL EXIT MODE
#       AT L+2 OR CALLING SEQUENCE (GOTO L+2)
# SUBROUTINES CALLED
#       LEMPREC
#       PERIAPO
# ALARM OR ABORT EXIT MODES
#       NONE
# ERASABLE INITIALIZATION REQUIRED
#       TIG             TIME OF IGNITION        DP B28CS
#       DELVSLV         SPECIFIED DELTA-V IN LOCAL VERT.
#                       COORDS. OF ACTIVE VEHICLE AT
#                       TIME OF IGNITION        VECTOR  B+7 METERS/CS
#
# OUTPUT
#       RTIG            POSITION AT TIG         VECTOR  B+29 METERS
#       VTIG            VELOCITY AT TIG         VECTOR  B+29 METERS/CS
#       PDL 4D          APOGEE ALTITUDE         DP      B+29 M ,  B+27 METERS.
#       HAPO            APOGEE ALTITUDE         DP      B+29 METERS
#       PDL 8D          PERIGEE ALTITUDE        DP      B+29 M ,  B+27 METERS.
#       HPER            PERIGEE ALTITUDE        DP      B+29 METERS
#       DELVSIN         SPECIFIED DELTA-V IN INTERTIAL
#                       COORD. OF ACTIVE VEHICLE AT
#                       TIME OF IGNITION        VECTOR  B+7 METERS/CS
#       DELVSAB         MAG. OF DELVSIN         VECTOR  B+7 METERS/CS
#
# DEBRIS        QTEMP   TEMP. ERASABLE
#               QPRET,MPAC
#               PUSHLIST

                SETLOC  P30S1
                BANK

                COUNT*  $$/S30S

S30.1           STQ     DLOAD
                        QTEMP
                        TIG             # TIME IGNITION SCALED AT 2(+28)CS
                STCALL  TDEC1
                        LEMPREC         # ENCKE ROUTINE FOR LEM

                VLOAD   SXA,2
```

```
                RATT
                RTX2
        STORE   RTIG            # RADIUS VECTOR AT IGNITION TIME
        UNIT    VCOMP
        STOVL   DELVSIN         # ZRF/LV IN DELVSIN SCALED AT 2
                VATT            # VELOCITY VECTOR AT TIG, SCALED 2(7) M/CS
        STORE   VTIG
        VXV     UNIT
                RTIG
        SETPD   SXA,1
                0
                RTX1
        PUSH    VXV             # YRF/LV PDL 0  SCALED AT 2
                DELVSIN
        VSL1    PDVL
        PDVL    PDVL            # YRF/LV PDL 6 SCALED AT 2
                DELVSIN         # ZRF/LV PDL 12D SCALED AT 2
                DELVSLV
        VXM     VSL1
                0
        STORE   DELVSIN         # DELTAV IN INERT. COOR. SCALED TO B+7M/CS
        ABVAL
        STOVL   DELVSAB         # DELTA V MAG.
                RTIG            # (FOR PERIAPO)
        PDVL    VAD             # VREQUIRED = VTIG + DELVSIN (FOR PERIAPO)
                VTIG
                DELVSIN
        CALL
                PERIAPO1
        CALL
                SHIFTR1         # RESCALE IF NEEDED
        CALL                    # LIMIT DISPLAY TO 9999.9 N. MI.
                MAXCHK
        STODL   HPER            # PERIGEE ALT 2(29) METERS, FOR DISPLAY
                4D
        CALL
                SHIFTR1         # RESCALE IF NEEDED
        CALL                    # LIMIT DISPLAY TO 9999.9 N. MI.
                MAXCHK
        STCALL  HAPO            # APOGEE  ALT 2(29) METERS, FOR DISPLAY
                QTEMP
```

```
# COELLIPTIC SEQUENCE INITIATION (CSI) PROGRAMS (P32 AND P72)
#
# MOD NO -1       LOG SECTION -- P32-P35, P72-P75
# MOD BY WHITE.P  DATE 1JUNE67
#
# PURPOSE

#       (1)     TO CALCULATE PARAMETERS ASSOCIATED WITH THE TIME FOLLOWING
#               CONCENTRIC FLIGHT PLAN MANEUVERS -- THE CO-ELLIPTIC SEQUENCE
#               INITIATION (CSI) MANEUVER AND THE CONSTANT DELTA ALTITUDE
#               (CDH) MANEUVER.

#       (2)     TO CALCULATE THESE PARAMETERS BASED UPON MANEUVER DATA
#               APPROVED AND KEYED INTO THE DSKY BY THE ASTRONAUT.

#       (3)     TO DISPLAY TO THE ASTRONAUT AND THE GROUND DEPENDENT VARIABLES
#               ASSOCIATED WITH THE CONCENTRIC FLIGHT PLAN MANEUVERS FOR
#               APPROVAL BY THE ASTRONAUT/GROUND.

#       (4)     TO STORE THE CSI TARGET PARAMETERS FOR USE BY THE DESIRED
#               THRUSTING PROGRAM.
#
# ASSUMPTIONS

#       (1)     AT A SELECTED TPI TIME THE LINE OF SIGHT BETWEEN THE ACTIVE
#               AND PASSIVE VEHICLES IS SELECTED TO BE A PRESCRIBED ANGLE (E)
#               FROM THE HORIZONTAL PLANE DEFINED BY THE ACTIVE VEHICLE
#               POSITION.

#       (2)     THE TIME BETWEEN CSI IGNITION AND CDH IGNITION MUST BE
#               COMPUTED TO BE GREATER THAN 10 MINUTES FOR SUCCESSFUL
#               COMPLETION OF THE PROGRAM.

#       (3)     THE TIME BETWEEN CDH IGNITION AND TPI IGNITION MUST BE
#               COMPUTED TO BE GREATER THAN 10 MINUTES FOR SUCCESSFUL
#               COMPLETION OF THE PROGRAM.

#       (4)     CDH DELTA V IS SELECTED TO MINIMIZE THE VARIATION OF THE
#               ALTITUDE DIFFERENCE BETWEEN THE ORBITS.

#       (5)     CSI BURN IS DEFINED SUCH THAT THE IMPULSIVE DELTA V IS IN THE
#               HORIZONTAL PLANE DEFINED BY THE ACTIVE VEHICLE POSITION AT CSI
#               IGNITION.

#       (6)     THE PERICENTER ALTITUDE OF THE ORBIT FOLLOWING CSI AND CDH
#               MUST BE GREATER THAN 35,000 FT (LUNAR ORBIT) OR 85 NM (EARTH
#               ORBIT) FOR SUCCESSFUL COMPLETION OF THIS PROGRAM.

#       (7)     THE CSI AND CDH MANEUVERS ARE ORIGINALLY ASSUMED TO BE
#               PARALLEL TO THE PLANE OF THE CSM ORBIT.  HOWEVER, CREW
```

```
#               MODIFICATION OF DELTA V (LV) COMPONENTS MAY RESULT IN AN
#               OUT-OF-PLANE CSI MANEUVER

#       (8)     STATE VECTOR UPDATES BY P27 ARE DISALLOWED DURING AUTOMATIC
#               STATE VECTOR UPDATING INITIATED BY P20 (SEE ASSUMPTION 10).

#       (9)     COMPUTED VARIABLES MAY BE STORED FOR LATER VERIFICATION BY
#               THE GROUND.  THESE STORAGE CAPABILITIES ARE NORMALLY LIMITED
#               ONLY TO THE PARAMETERS FOR ONE THRUSTING MANEUVER AT A TIME
#               EXCEPT FOR CONCENTRIC FLIGHT PLAN MANEUVER SEQUENCES.

#       (10)    THE RENDEZVOUS RADAR MAY OR MAY NOT BE USED TO UPDATE THE LM
#               OR CSM STATE VECTORS FOR THIS PROGRAM.  IF RADAR USE IS
#               DESIRED THE RADAR WAS TURNED ON AND LOCKED BY THE CSM BY
#               PREVIOUS SELECTION OF P20.  RADAR SIGHTING MARKS WILL BE MADE
#               AUTOMATICALLY APPROXIMATELY ONCE A MINUTE WHEN ENABLED BY THE
#               TRACK AND UPDATE FLAGS (SEE P20).  THE RENDEZVOUS TRACKING
#               MARK COUNTER IS ZEROED BY THE SELECTION OF P20 AND AFTER EACH
#               THRUSTING MANEUVER.

#       (11)    THE ISS NEED NOT BE ON TO COMPLETE THIS PROGRAM.

#       (12)    THE OPERATION OF THE PROGRAM UTILIZES THE FOLLOWING FLAGS --
#
#                       ACTIVE VEHICLE FLAG -- DESIGNATES THE VEHICLE WHICH IS
#                       DOING RENDEZVOUS THRUSTING MANEUVERS TO THE PROGRAM WHICH
#                       CALCULATES THE MANEUVER PARAMETERS.  SET AT THE START OF
#                       EACH RENDEZVOUS PRE-THRUSTING PROGRAM.
#
#                       FINAL FLAG -- SELECTS FINAL PROGRAM DISPLAYS AFTER CREW HAS
#                       COMPLETED THE FINAL MANEUVER COMPUTATION AND DISPLAY
#                       CYCLE.
#
#                       EXTERNAL DELTA V STEERING FLAG -- DESIGNATES THE TYPE OF
#                       STEERING REQUIRED FOR EXECUTION OF THIS MANEUVER BY THE
#                       THRUSTING PROGRAM SELECTED AFTER COMPLETION OF THIS
#                       PROGRAM.
#
#       (13)    IT IS NORMALLY REQUIRED THAT THE ISS BE ON FOR 1 HOUR PRIOR TO
#               A THRUSTING MANEUVER.
#
#       (14)    THIS PROGRAM IS SELECTED BY THE ASTRONAUT BY DSKY ENTRY
#
#                       P32 IF THIS VEHICLE IS ACTIVE VEHICLE.
#
#                       P72 IF THIS VEHICLE IS THE PASSIVE VEHICLE.
#
# INPUT

#       (1)     TCSI            TIME OF THE CSI MANEUVER
```

```
#       (2)     NN              NUMBER OF APSIDAL CROSSINGS THRU WHICH THE ACTIVE
#                               VEHICLE ORBIT CAN BE ADVANCED TO OBTAIN THE CDH
#                               MANEUVER POINT.
#       (3)     ELEV            DESIRED LOS ANGLE AT TPI
#       (4)     TTPI            TIME OF THE TPI MANEUVER
#
# OUTPUT

#       (1)     TRKMKCNT        NUMBER OF MARKS
#       (2)     TTOGO           TIME TO GO
#       (3)     +MGA            MIDDLE GIMBAL ANGLE
#       (4)     DIFFALT         DELTA ALTITUDE AT CDH
#       (5)     T1TOT2          DELTA TIME FROM CSI TO CDH
#       (6)     T2TOT3          DELTA TIME FROM CDH TO TPI
#       (7)     DELVLVC         DELTA VELOCITY AT CSI -- LOCAL VERTICAL COORDINATES
#       (8)     DELVLVC         DELTA VELOCITY AT CDH -- LOCAL VERTICAL COORDINATES
#
# DOWNLINK

#       (1)     TCSI            TIME OF THE CSI MANEUVER
#       (2)     TCDH            TIME OF THE CDH MANEUVER
#       (3)     TTPI            TIME OF THE TPI MANEUVER
#       (4)     TIG             TIME OF THE CSI MANEUVER
#       (5)     DELVEET1        DELTA VELOCITY AT CSI -- REFERENCE COORDINATES
#       (6)     DELVEET2        DELTA VELOCITY AT CDH -- REFERENCE COORDINATES
#       (7)     DIFFALT         DELTA ALTITUDE AT CDH
#       (8)     NN              NUMBER OF APSIDAL CROSSINGS THRU WHICH THE ACTIVE
#                               VEHICLE ORBIT CAN BE ADVANCED TO OBTAIN THE CDH
#                               MANEUVER POINT
#       (9)     ELEV            DESIRED LOS ANGLE AT TPI
#
# COMMUNICATION TO THRUSTING PROGRAM

#       (1)     TIG             TIME OF THE CSI MANEUVER
#       (2)     RTIG            POSITION OF ACTIVE VEHICLE AT CSI -- BEFORE ROTATION
#                               INTO PLANE OF PASSIVE VEHICLE
#       (3)     VTIG            VELOCITY OF ACTIVE VEHICLE AT CSE -- BEFORE ROTATION
#                               INTO PLANE OF PASSIVE VEHICLE
#       (4)     DELVSIN         DELTA VELOCITY AT CSI -- REFERENCE COORDINATES
#       (5)     DELVSAB         MAGNITUDE OF DELTA VELOCITY AT CSI
#       (6)     XDELVFLG        SET TO INDICATE EXTERNAL DELTA V VG COMPUTATION
#
# SUBROUTINES USED

#       AVFLAGA
#       AVFLAGP
#       P20FLGON
#       VARALARM
#       BANKCALL
#       GOFLASH
#       GOTOPOOH
```

```
#		VNPOOH
#		GOFLASHR
#		BLANKET
#		ENDOFJOB
#		SELECTMU
#		ADVANCE
#		INTINT
#		PASSIVE
#		CSI/A
#		S32/33.1
#		DISDVLVC
#		VN1645

		BANK	35
		SETLOC	CSI/CDH
		BANK
		EBANK=	SUBEXIT
		COUNT*	$$/P3272
P32		TC	AVFLAGA
		TC	P32STRT
P72		TC	AVFLAGP
P32STRT		EXTEND
		DCA	P30ZERO
		DXCH	CENTANG
		TC	P32/P72A
ALMXITA		SXA,2
			CSIALRM
ALMXIT		LXC,1
			CSIALRM
		SLOAD*	EXIT
			ALARM/TB -1,1
		CA	MPAC
		TC	VARALARM
		CAF	V05N09
		TC	BANKCALL
		CADR	GOFLASH
		TC	GOTOPOOH
		TC	-4
P32/P72A	TC	P20FLGON
		CAF	P30ZERO
		TS	NN	+1
		TS	TCSI
		TS	TCSI	+1
VN0611		CAF	V06N11		# TCSI
		TC	VNPOOH
		TC	INTPRET
		DLOAD	DCOMP
			TCSI
		BMN	DLOAD
			VN0655
```

```
                        TETLEM
                STCALL  TDEC1
                        PRECSET
                VLOAD   VSR*
                        RACT3
                        0,2
                STOVL   RVEC
                        VACT3
                VSR*    SET
                        0,2
                        RVSW
                STODL   VVEC
                        DPPOSMAX
                STCALL  RDESIRED
                        TIMERAD
                DAD
                        TDEC2
                STORE   TCSI
                EXIT
                TC      VN0611
VN0655          EXIT
                CAF     V06N55          # NN, ELEV(RGLOS)
                TC      BANKCALL
                CADR    GOFLASH
                TC      GOTOPOOH
                TC      +2
                TC      -5
                CAF     V06N37          # TTPI
                TC      VNPOOH
                TC      INTPRET
                DLOAD
                        TCSI
                STCALL  TIG
                        SELECTMU
P32/P72B        CALL
                        ADVANCE
                SETPD   VLOAD
                        0D
                        VPASS1
                PDVL    PDDL
                        RPASS1
                        TCSI
                PDDL    PDDL
                        TTPI
                        TWOPI
                PUSH    CALL
                        INTINT
                CALL
                        PASSIVE
                CALL
```

```
                                        CSI/A
        P32/P72C         BON            SET
                                        FINALFLG
                                        P32/P72D
                                        UPDATFLG
        P32/P72D         DLOAD
                                        T1TOT2
        P32/P72E         STORE          T1TOT2
                         DSU            BPL
                                        60MIN
                                        P32/P72E
                         DLOAD
                                        T2TOT3
        P32/P72F         STORE          T2TOT3
                         DSU            BPL
                                        60MIN
                                        P32/P72F
                         EXIT
                         CAF            V06N75
                         TC             VNPOOH
                         TC             INTPRET
                         VLOAD          CALL
                                        DELVEET1
                                        S32/33.1
                         STOVL          DELVEET1
                                        RACT2
                         STOVL          RACT1
                                        DELVEET2
                         AXT,1          CALL
                         VN             0682
                                        DISDVLVC
                         DLOAD
                                        TTPI
                         STCALL         TTPIO
                                        VN1645
                         GOTO
                                        P32/P72B
```

```
# CONSTANT DELTA HEIGHT (CDH) PROGRAMS (P33 AND P73)
# MOD NO -1                       LOC SECTION -- P32-P35, P72-P75
# MOD BY WHITE, P.                DATE: 1 JUNE 67
#
# PURPOSE
#
#       (1)     TO CALCULATE PARAMETERS ASSOCIATED WITH THE CONSTANT DELTA
#               ALTITUDE MANEUVER (CDH).
#
#       (2)     TO CALCULATE THESE PARAMETERS BASED UPON MANEUVER DATA
#               APPROVED AND KEYED INTO THE DSKY BY THE ASTRONAUT.
#
#       (3)     TO DISPLAY TO THE ASTRONAUT AND THE GROUND DEPENDENT VARIABLES
#               ASSOCIATED WITH THE CDH MANEUVER FOR APPROVAL BY THE
#               ASTRONAUT/GROUND.
#
#       (4)     TO STORE THE CDH TARGET PARAMETERS FOR USE BY THE DESIRED
#               THRUSTING PROGRAM.
#
# ASSUMPTIONS
#
#       (1)     THIS PROGRAM IS BASED UPON PREVIOUS COMPLETION OF THE
#               CO-ELLIPTIC SEQUENCE INITIATION (CSI) PROGRAM (P32/P72).
#               THEREFORE --
#
#               (A)     AT A SELECTED TPI TIME (NOW IN STORAGE) THE LINE OF SIGHT
#                       BETWEEN THE ACTIVE AND PASSIVE VEHICLES WAS SELECTED TO BE
#                       A PRESCRIBED ANGLE (E) (NOW IN STORAGE) FROM THE
#                       HORIZONTAL PLANE DEFINED BY THE ACTIVE VEHICLE POSITION.
#
#               (B)     THE TIME BETWEEN CSI IGNITION AND CDH IGNITION WAS
#                       COMPUTED TO BE GREATER THAN 10 MINUTES.
#
#               (C)     THE TIME BETWEEN CDH IGNITION AND TPI IGNITION WAS
#                       COMPUTED TO BE GREATER THAN 10 MINUTES.
#
#               (D)     THE VARIATION OF THE ALTITUDE DIFFERENCE BETWEEN THE
#                       ORBITS WAS MINIMIZED.
#
#               (E)     CSI BURN WAS DEFINED SUCH THAT THE IMPULSIVE DELTA V WAS
#                       IN THE HORIZONTAL PLANE DEFINED BY ACTIVE VEHICLE
#                       POSITION AT CSI IGNITION.
#
#               (F)     THE PERICENTER ALTITUDES OF THE ORBITS FOLLOWING CSI AND
#                       CDH WERE COMPUTED TO BE GREATER THAN 35,000 FT FOR LUNAR
#                       ORBIT OR 85 NM FOR EARTH ORBIT.
#
#               (G)     THE CSI AND CDH MANEUVERS WERE ASSUMED TO BE PARALLEL TO
#                       THE PLANE OF THE PASSIVE VEHICLE ORBIT.  HOWEVER, CREW
```

```
#                       MODIFICATION OF DELTA V (LV) COMPONENTS MAY HAVE RESULTED
#                       IN AN OUT-OF-PLANE MANEUVER.
#
#       (2)     STATE VECTOR UPDATES BY P27 ARE DISALLOWED DURING AUTOMATIC
#               STATE VECTOR UPDATING INITIATED BY P20 (SEE ASSUMPTION 4).
#
#       (3)     COMPUTED VARIABLES MAY BE STORED FOR LATER VERIFICATION BY
#               THE GROUND.  THESE STORAGE CAPABILITIES ARE NORMALLY LIMITED
#               ONLY TO THE PARAMETERS FOR ONE THRUSTING MANEUVER AT A TIME
#               EXCEPT FOR CONCENTRIC FLIGHT PLAN MANEUVER SEQUENCES.
#
#       (4)     THE RENDEZVOUS RADAR MAY OR MAY NOT BE USED TO UPDATE THE LM.
#               OR CSM STATE VECTORS FOR THIS PROGRAM.  IF RADAR USE IS
#               DESIRED THE RADAR WAS TURNED ON AND LOCKED ON THE CSM BY
#               PREVIOUS SELECTION OF P20.  RADAR SIGHTING MARKS WILL BE MADE
#               AUTOMATICALLY APPROXIMATELY ONCE A MINUTE WHEN ENABLED BY THE
#               TRACK AND UPDATE FLAGS (SEE P20).  THE RENDEZVOUS TRACKING
#               MARK COUNTER IS ZEROED BY THE SELECTION OF P20 AND AFTER EACH
#               THRUSTING MANEUVER.
#
#       (5)     THE ISS NEED NOT BE ON TO COMPLETE THIS PROGRAM.
#
#       (6)     THE OPERATION OF THE PROGRAM UTILIZES THE FOLLOWING FLAGS --
#
#                       ACTIVE VEHICLE FLAG -- DESIGNATES THE VEHICLE WHICH IS
#                       DOING RENDEZVOUS THRUSTING MANEUVERS TO THE PROGRAM WHICH
#                       CALCULATES THE MANEUVER PARAMETERS.  SET AT THE START OF
#                       EACH RENDEZVOUS PRE-THRUSTING PROGRAM.
#
#                       FINAL FLAG -- SELECTS FINAL PROGRAM DISPLAYS AFTER CREW HAS
#                       COMPLETED THE FINAL MANEUVER COMPUTATION AND DISPLAY
#                       CYCLE.
#
#                       EXTERNAL DELTA V STEERING FLAG -- DESIGNATES THE TYPE OF
#                       STEERING REQUIRED FOR EXECUTION OF THIS MANEUVER BY THE
#                       THRUSTING PROGRAM SELECTED AFTER COMPLETION OF THIS
#                       PROGRAM.
#
#       (7)     IT IS NORMALLY REQUIRED THAT THE ISS BE ON FOR 1 HOUR PRIOR TO
#               A THRUSTING MANEUVER.
#
#       (8)     THIS PROGRAM IS SELECTED BY THE ASTRONAUT BY DSKY ENTRY.
#
#                       P33 IF THIS VEHICLE IS ACTIVE VEHICLE.
#
#                       P73 IF THIS VEHICLE IS PASSIVE VEHICLE.
#
# INPUT
#
#       (1)     TTPIO   TIME OF THE TPI MANEUVER -- SAVED FROM P32/P72
```

```
#       (2)     ELEV    DESIRED LOS ANGLE AT TPI -- SAVED FROM P32/P72
#       (3)     TCDH    TIME OF THE CDH MANEUVER
#
# OUTPUT
#
#       (1)     TRKMKCNT        NUMBER OF MARKS
#       (2)     TTOGO           TIME TO GO
#       (3)     +MGA            MIDDLE GIMBAL ANGLE
#       (4)     DIFFALT         DELTA ALTITUDE AT CDH
#       (5)     T2TOT3          DELTA TIME FROM CDH TO COMPUTED TPI
#       (6)     NOMTPI          DELTA TIME FROM NOMINAL TPI TO COMPUTED TPI
#       (7)     DELVLVC         DELTA VELOCITY AT CDH -- LOCAL VERTICAL COORDINATES
#
# DOWNLINK
#
#       (1)     TCDH            TIME OF THE CDH MANEUVER
#       (2)     TTPI            TIME OF THE TPI MANEUVER
#       (3)     TIG             TIME OF THE CDH MANEUVER
#       (4)     DELLVEET2       DELTA VELOCITY AT CDH -- REFERENCE COORDINATES
#       (5)     DIFFALT         DELTA ALTITUDE AT CDH
#       (6)     ELEV            DESIRED LOS ANGLE AT TPI
#
# COMMUNICATION TO THRUSTING PROGRAMS
#
#       (1)     TIG             TIME OF THE CDH MANEUVER
#       (2)     RTIG            POSITION OF ACTIVE VEHICLE AT CDH -- BEFORE ROTATION
#                               INTO PLANE OF PASSIVE VEHICLE.
#       (3)     VTIG            VELOCITY OF ACTIVE VEHICLE AT CDH -- BEFORE ROTATION
#                               INTO PLANE OF PASSIVE VEHICLE.
#       (4)     DELVSIN         DELTA VELOCITY AT CDH -- REFERENCE COORDINATES.
#       (5)     DELVSAB         MAGNITUDE OF DELTA VELOCITY AT CDH.
#       (6)     XDELVFLG        SET TO INDICATE EXTERNAL DELTA V VG COMPUTATION.
#
# SUBROUTINES USED
#
#       AVFLAGA
#       AVFLAGP
#       P20FLGON
#       VNPOOH
#       SELECTMU
#       ADVANCE
#       CDHMVR
#       INTINT3P
#       ACTIVE
#       PASSIVE
#       S33/S34.1
#       ALARM
#       BANKCALL
#       GOFLASH
#       GOTOPOOH
#       S32/33.1
```

```
#	VN1645

		COUNT*	$$/P3373
P33		TC	AVFLAGA
		TC	P33/P73A
P73		TC	AVFLAGP
P33/P73A	TC	P20FLGON
		CAF	V06N13		# TCDH
		TC	VNPOOH
		TC	INTPRET
		DLOAD
			TTPIO
		STODL	TTPI
			TCDH
		STCALL	TIG
			SELECTMU
P33/P73B	CALL
			ADVANCE
		CALL
			CDHMVR
		SETPD	VLOAD
			OD
			VACT3
		PDVL	CALL
			RACT2
			INTINT3P
		CALL
			ACTIVE
		SETPD	VLOAD
			OD
			VPASS2
		PDVL	CALL
			RPASS2
			INTINT3P
		CALL
			PASSIVE
		DLOAD	SET
			P30ZERO
			ITSWICH
		STCALL	NOMTPI
			S33/34.1
		BZE	EXIT
			P33/P73C
		TC	ALARM
		OCT	611
		CAF	V05N09
		TC	BANKCALL
		CADR	GOFLASH
		TC	GOTOPOOH
		TC	+2
```

```
                TC      P33/P73A
                TC      INTPRET
                DLOAD
                        P30ZERO
                STORE   NOMTPI
P33/P73C        BON     SET
                        FINALFLG
                        P33/P73D
                        UPDATFLG
P33/P73D        DLOAD   DAD
                        NOMTPI
                        TTPI
                STORE   TTPI
                DSU
                        TCDH
P33/P73E        DSU     BPL
                        60MIN
                        P33/P73E
                DAD
                        60MIN
                STODL   T1TOT2
                        TTPI
                DSU     PUSH
                        TTPIO
P33/P73F        ABS     DSU
                        60MIN
                BPL     DAD
                        P33/P73F
                        60MIN
                SIGN    STADR
                STORE   T2TOT3
                EXIT
                CAF     V06N75
                TC      VNPOOH
                TC      INTPRET
                VLOAD   CALL
                        DELVEET2
                        S32/33.1
                STCALL  DELVEET2
                        VN1645
                GOTO
                        P33/P73B
```

```
# ***** ADFLAG/P *****
#
# SUBROUTINES USED
#
#       UPFLAG
#       DOWNFLAG

AVFLAGA         EXTEND                          # AVFLAG = LEM
                QXCH    SUBEXIT
                TC      UPFLAG
                ADRES   AVFLAG
                TC      SUBEXIT
AVFLAGP         EXTEND                          # AVFLAG = CSM
                QXCH    SUBEXIT
                TC      DOWNFLAG
                ADRES   AVFLAG
                TC      SUBEXIT
P20FLGON        EXTEND
                QXCH    SUBEXIT
                TC      UPFLAG
                ADRES   UPDATFLG        # SET UPDATFLG
                TC      UPFLAG
                ADRES   TRACKFLG        # SET TRACKFLG
                TC      SUBEXIT
```

```
# ***** DISDVLVC *****
#
# SUBROUTINES USED
#
#       S32/33.X
#       VNPOOH

DISDVLVC        STORE   DELVLVC
                STQ     CALL
                        NORMEX
                        S32/33.X
                VLOAD   MXV
                        DELVLVC
                        OD
                VSL1    SXA,1
                        VERBNOUN
                STORE   DELVLVC
                EXIT
                CA      VERBNOUN
                TC      VNPOOH
                TC      INTPRET
                GOTO
                        NORMEX
```

```
# ***** CONSTANTS *****

V06N11          VN      0611
V06N13          VN      0613
V06N75          VN      0675
SN359+          2DEC    -.000086601

CS359+          2DEC    +.499999992

P30ZERO         2DEC    0

60MIN           2DEC    360000

ALARM/TB        OCT     00600           # NO 1
                OCT     00601           #    2
                OCT     00602           #    3
                OCT     00603           #    4
                OCT     00604           #    5
                OCT     00605           #    6
                OCT     00606           #    7
```

```
# ***** CSI/A *****
#
# SUBROUTINES USED
#
#       VECSHIFT
#       TIMETHET
#       PERIAPO
#       SHIFTR1
#       INTINT2C
#       CDHMVR
#       PERIAPO1
#       INTINT
#       ACTIVE

                BANK    34
                SETLOC  CSI/CDH1
                BANK
                EBANK=  SUBEXIT
                COUNT*  $$/CSI
LOOPMX          2DEC    16

INITST          2DEC    .03048 B-7      # INITIAL DELDV = 10 FPS

DVMAX1          2DEC    3.0480 B-7      # MAXIMUM DV1 = 1000 FPS

DVMAX2          2DEC    3.014472 B-7    #                989 FPS

1DPB2           2DEC    1.0 B-2

1DPB28          2DEC    1

PMINE           2DEC    157420 B-29     # 85 NM -- MUST BE 8 WORDS BEFORE PMINM

EPSILN1         2DEC    .0003048 B-7    # .1 FPS

NICKELDP        2DEC    .021336 B-7     # 7 FPS (CHANGED FROM .05 FPS)

FIFPSDP         2DEC    -.152400 B-7    # 50 FPS

PMINM           2DEC    10668 B-29      # 35000 FT -- MUST BE 8 WORDS AFTER PMINE

DELMAX1         2DEC    .6096000 B-7    # 200 FPS

ONETHTH         2DEC    .0001 B-3

TMIN            2DEC    60000           # 10 MIN

CSI/A           CLEAR   SET             # INITIALIZE INDICATORS
                        S32.1F1         # DVT1 HAS EXCEEDED MAX INDICATOR
                        S32.1F2         # FIRST PASS FOR NEWTON ITERATION INDICATOR
```

```
                CLEAR   SET
                        S32.1F3A        # 00=1ST 2 PASSES 2ND CYCLE, 01=FIRST CYCLE
                        S32.1F3B        # 10=2ND CYCLE, 11=50 FPS STAGE 2ND CYCLE
                DLOAD
                        P30ZERO
                STORE   LOOPCT
                STORE   CSIALRM
CSI/B           SETPD   VLOAD
                        0D
                        RACT1
                ABVAL   PUSH            # RA1                         B29 PL02D
                NORM    SR1
                        X2              #                      B29-N2+ B1 PL04D
                PDVL    ABVAL
                        RPASS3
                NORM    BDDV            # RA1/RP3                     B1 PL02D
                        X1
                XSU,2   SR*             #                             B2
                        X1
                        1,2
                DAD     DMP             # (1+(RA1/RP3))RA1      B29+B2=B31 PL00D
                        1DPB2
                NORM    PDDL            #                                  PL02D
                        X1
                        RTMU
                SR1     DDV             #                      B38-B31= B7 PL00D
                SL*     SQRT            #                              B7
                        0       -7,1
                PDVL    UNIT            #                                  PL02D
                        RACT1
                PDVL    VXV
                        UP1
                UNIT                    # UNIT(URP1 X UVP1 X URA1) = UH1
                DOT     SL1             # VA1 . UH1                    B7
                        VACT1
                BDSU    STADR           #                                  PL00D
                STODL   DELVCSI
                        INITST          # 10 FPS
                STORE   DELDV
CSI/B1          DLOAD   DAD             # IF LOOPCT = 16
                        LOOPCT
                        1DPB28
                STORE   LOOPCT
                DSU     AXT,2
                        LOOPMX
                        6
                BPL
                        SCNDSOL
CSI/B2          SETPD
                        0D
```

```
                DLOAD   ABS
                        DELVCSI
                DSU     BMN
                        DVMAX1
                        CSI/B23
                AXT,2   BON
                        7
                        S32.1F1
                        SCNDSOL
                BOFF    BON
                        S32.1F3A
                        CSI/B22         # FLAG 3 NEQ 3
                        S32.1F3B
                        SCNDSOL
CSI/B22         SET     DLOAD
                        S32.1F1
                        DVMAX2
                SIGN
                        DELVCSI
                STORE   DELVCSI
CSI/B23         VLOAD   PUSH
                        RACT1
                UNIT    PDVL
                        UP1
                VXV     UNIT            # UNIT (URP1 X UVP1 X URA1) = UH1
                VXSC    VSL1
                        DELVCSI
                STORE   DELVEET1
                VAD     BOV
                        VACT1
                        CSI/B23D
CSI/B23D        STCALL  VACT4
                        VECSHIFT
                STOVL   VVEC
                SET
                        RVSW
                STOVL   RVEC
                        SN359+
                STCALL  SNTH            # ALSO CSTH
                        TIMETHET
                SR1     LXA,1
                        RTX1
                STCALL  HAFPA1
                        PERIAPO
                CALL
                        SHIFTR1
                STODL   POSTCSI
                        CENTANG
                BZE     GOTO
                        +2
```

```

                      CIRCL
              DLOAD
                      ECC
              DSU     BMN
                      ONETHTH
                      CIRCL
              DLOAD   CALL
                      R1
                      SHIFTR1
              SETPD   NORM
                      2D
                      X1
              PDVL    DOT                #                                       PL04D
                      RACT1
                      VACT4
              ABS     DDV
                      02D                # (/RDOTV/)/R1             B38-B29= B7
              SL*     DSU
                      0,1
                      NICKELDP
              BMN     DLOAD
                      CIRCL
                      P
              SL2     DSU
                      1DPB2
              STODL   14D
                      RTSR1/MU
              SR1     DDV                # (1/ROOTMU)/R1         B-16-B29 = B-45 PL02D
              PDDL    DMP
                      P
                      R1
              CALL
                      SHIFTR1
              SL4     SL1
              SQRT    DMP                # ((P/MU)**.5)/R1       B14+B-14 = B-31 PL02D
              BOFF    SL3
                      CMOONFLG
                      CSI/B3
CSI/B3        PDVL    DOT
                      RACT1
                      VACT4
              STORE   RDOTV
              ABS
              NORM    DMP                # ((P/MU)**.5)RDOTV/R1                  PL02D
                      X2
              XSU,1   SL*                #                      B-31+B36-B3 = B2
                      X2
                      3,1
              STODL   12D
                      P30ZERO
```

```
                STORE   16D
                VLOAD   UNIT
                        12D
                STOVL   SNTH            # ALSO STORES CSTH AND 0
                        RACT1
                PDVL    SIGN
                        VACT4
                        RDOTV
                VCOMP   CALL
                        VECSHIFT
                STOVL   VVEC
                SET
                        RVSW
                STCALL  RVEC
                        TIMETHET
                PDDL    BPL
                        RDOTV
                        NTP/2
                DLOAD   DSU
                        HAFPA1
                PUSH    GOTO
                        NTP/2
CIRCL           SETPD   DLOAD
                        00D
                        P30ZERO
                PUSH
NTP/2           DLOAD   DMP
                        NN
                        HAFPA1
                SL      DSU
                        14D
                DAD
                        TCSI
                STORE   TCDH
                BDSU    AXT,2
                        TTPI
                        5D
                BMN     SETPD
                        SCNDSOL
                        0D
                VLOAD   PDVL
                        VACT4
                        RACT1
                CALL
                        INTINT2C
                STOVL   RACT2
                        VATT
                STOVL   VACT2
                        VPASS1
                SETPD   PDVL
```

```
                OD
                RPASS1
        CALL
                INTINT2C
        STOVL   RPASS2
                VATT
        STCALL  VPASS2
                CDHMVR
        VLOAD   SETPD
                RACT2
                OD
        PDVL    CALL
                VACT3
                PERIAPO1
        CALL
                SHIFTR1
        STOVL   POSTCDH
                VACT3
        SETPD   PDVL
                OD
                RACT2
        PDDL    PDDL
                TCDH
                TTPI
        PDDL    PUSH
                TWOPI
        CALL
                INTINT
        CALL
                ACTIVE
        DLOAD
                ELEV
        SETPD   SINE
                6D
        PDVL    UNIT
                RACT3
        STORE   OOD             # URA3 AT OOD
        PDVL    VXV             # PL14D, PL08D
                UP1
        UNIT
        PDDL    COSINE          # UNIT(URA3 X UVA3 X URA3) = UH3        B1 PL14D
                ELEV
        VXSC    STADR           # (COSLOS)(UH3)                         B2 PL08D
        STORE   18D             #       PLUS
        DLOAD   VXSC            # (SINLOS)(URA3) = U                    B2 PLOOD
        VAD     VSL1
                18D             #                                       B1
        PUSH    DOT             #                                          PL06D
                RACT3           # (U . RA3) = TEMP1           B1 +B29 = B30
        SL1     PUSH            #                                      B29 PL08D
```

```
                DSQ      TLOAD          # TEMP1**2                          B58
                         MPAC
                PDVL     DOT            #                                       PL11D
                         RACT3
                         RACT3
                TLOAD    DCOMP          # RA3 . RA3
                         MPAC
                PDVL     DOT            # RP3 . RP3                         B58 PL14D
                         RPASS3
                         RPASS3         #                                       PL11D
                TAD      TAD            # TEMP1**2 + RA3.RA3 + RP3.RP3 = TEMP2  PL08D
                BPL      DLOAD
                         K1ORK2
                         LOOPCT
                DSU      AXT,2
                         1DPB28
                         1D
                BZE
                         ALMXITA
                DLOAD    SR1
                         DELDV
                STORE    DELDV
                BDSU
                         DVPREV
                STCALL   DELVCSI
                         CSI/B1
K1ORK2          SQRT     PUSH           # TEMP3 = TEMP2**.5                 B29 PL10D
                DCOMP    DSU
                         06D            # -TEMP1-TEMP3 = K2 AT 10D
                STODL    10D            #                                       PL08D
                DSU      STADR          #                                       PL06D
                STORE    12D            # -TEMP1+TEMP3 = K1 AT 12D
                ABS
                STODL    14D
                         10D
                ABS      DSU
                         14D
                BMN      DLOAD
                         K2.
                         12D
                STORE    10D            # K = K1
K2.             DLOAD
                         10D
                VXSC     VSL1
                VAD      UNIT           # V = RA3 + KU UNIT                  B1
                         RACT3
                PDVL     UNIT
                         RPASS3         #                                       PL06D
                PDVL     UNIT
                         VPASS3         #                                       PL12D
```

```
                    VXV      PDVL            # UVP3 X URP3                              PL18D
                             06D
                             06D
                    VXV      DOT
                             00D
                    STADR                    #                                          PL12D
                    STOVL    12D             # (URP3 X V).(UVP3 X URP3)=TEMP            PL06D
                    DOT      SL1             #                                          PL00D
                    ARCCOS   SIGN
                             12D             #                                       B0
                    SR1      PUSH            # GAMMA = SIGN(TEMP)ARCOS(UNITV.URP3)      PL02D
                    BON      DLOAD
                             S32.1F2
                             FRSTPAS
                             00D             # NOT THE FIRST PASS OF A CYCLE
                    DSU      PDDL            # GAMMA-GAMPREV                         B1 PL04D
                             GAMPREV
                             DELVCSI
                    DSU      NORM            #                                       B7
                             DVPREV
                             X1
                    BDDV     PDDL            # (GAM-GAMPREV)/(DV-DVPREV)          B-6+N1 PL06D
                             02D             #        = SLOPE
                             DELVCSI
                    STORE    DVPREV
                    BOFF     BOFF
                             S32.1F3A
                             THRDCHK
                             S32.1F3B
                             THRDCHK
                    DLOAD    DMP
                             02D
                             GAMPREV
                    BPL      DLOAD
                             FIFTYFPS
                             INITST
                    SIGN
                             DELDV
                    STORE    DELDV
                    SET      CLEAR
                             S32.1F3A
                             S32.1F3B
FRSTPAS             DLOAD
                             00D
                    STODL    GAMPREV
                             DELVCSI
                    STORE    DVPREV
                    DSU      CLEAR
                             DELDV
                             S32.1F2
```

```
                STCALL  DELVCSI
                        CSI/B1
THRDCHK         BON     BON
                        S32.1F3A
                        NEWTN
                        S32.1F3B
                        NEWTN
FIFTYFPS        DLOAD   SIGN
                        FIFPSDP
                        04D
                SIGN
                        GAMPREV
                STORE   DELDV
                DCOMP   DAD
                        DELVCSI
                STODL   DELVCSI
                        00D
                SET     SET
                        S32.1F3B
                        S32.1F3A
                STCALL  GAMPREV
                        CSI/B2
NEWTN           DLOAD   NORM
                        04D
                        X2
                BDDV    XSU,1
                        00D
                        X2
                SR*
                        0,1
                STODL   DELDV
                        00D
                STORE   GAMPREV
                DLOAD   ABS
                        DELDV           #                                             PL08D
                PUSH    DSU
                        EPSILN1
                BMN     DLOAD
                        CSI/SOL
                DSU     BMN
                        DELMAX1
                        CSISTEP
                DLOAD   SIGN
                        DELMAX1
                        DELDV
                STORE   DELDV
CSISTEP         DLOAD   DSU
                        DELVCSI
                        DELDV
                STCALL  DELVCSI
```

```
                                CSI/B1
CSI/SOL             DLOAD       AXT,2
                                POSTCSI
                                2
                    LXA,1
                                RTX1
                    DSU*        BMN
                                PMINE       -2,1
                                SCNDSOL
                    AXT,2       DLOAD
                                3
                                POSTCDH
                    DSU*        BMN
                                PMINE       -2,1
                                SCNDSOL
                    DLOAD       DSU
                                TCDH
                                TCSI
                    STORE       T1TOT2
                    AXT,2       DSU
                                4
                                TMIN
                    BMN         AXT,2
                                SCNDSOL
                                5
                    DLOAD       DSU
                                TTPI
                                TCDH
                    STORE       T2TOT3
                    DSU         BPL
                                TMIN
                                P32/P72C
SCNDSOL             BON         BOFF
                                S32.1F3A
                                ALMXIT
                                S32.1F3B
                                ALMXIT
                    SXA,2       DLOAD
                                CSIALRM
                                P30ZERO
                    CLEAR       SET
                                S32.1F1
                                S32.1F2
                    CLEAR       CLEAR
                                S32.1F3A
                                S32.1F3B
                    STCALL      LOOPCT
                                CSI/B
```

```
# ***** ADVANCE *****
#
# SUBROUTINES USED
#       PRECSET
#       ROTATE

ADVANCE         STQ     DLOAD
                        SUBEXIT
                        TIG
                STCALL  TDEC1
                        PRECSET
                SET     VLOAD
                        XDELVFLG
                        VPASS3
                STORE   VPASS2
                STOVL   VPASS1
                        RPASS3
                STORE   RPASS2
                STORE   RPASS1
                UNIT    VXV
                        VPASS1
                UNIT
                STOVL   UP1
                        RACT3
                STCALL  RTIG
                        ROTATE
                STORE   RACT2
                STOVL   RACT1
                        VACT3
                STCALL  VTIG
                        ROTATE
                STORE   VACT2
                STCALL  VACT1
                        SUBEXIT
```

```
# ***** ROTATE *****

ROTATE          PUSH     PUSH
                DOT      VXSC
                         UP1
                         UP1
                VSL2     BVSU
                UNIT     PDVL
                ABVAL    VXSC
                VSL1     RVQ
```

```
# ***** INTINTNA *****

INTINT2C        PDDL     PDDL
                         TCSI
                         TCDH
                PDDL     PUSH
                         TWOPI
                GOTO
                         INTINT
INTINT3P        PDDL     PDDL
                         TCDH
                         TTPI
                PDDL     PUSH
                         P30ZERO
                GOTO
                         INTINT
```

```
# ***** S32/33.1 *****
#
# SUBROUTINES USED
#       S32/33.X

S32/33.1        STQ     AXT,1
                        SUBEXIT
                VN      0681
                CALL
                        DISDVLVC
                CALL
                        S32/33.X
                VLOAD   VXM
                        DELVLVC
                        0D
                VSL1
                STORE   DELVSIN
                PUSH    ABVAL
                STOVL   DELVSAB
                GOTO
                        SUBEXIT
```

```
# ***** S32/33.X *****

S32/33.X        SETPD   VLOAD
                        6D
                        UP1
                VCOMP   PDVL
                        RACT1
                UNIT    VCOMP
                PUSH    VXV
                        UP1
                VSL1
                STORE   0D
                RVQ
```

```
# ***** CDHMVR *****
#
# SUBROUTINES USED
#       VECSHIFT
#       TIMETHET
#       SHIFTR1

CDHMVR          STQ     VLOAD
                        SUBEXIT
                        RACT2
                PUSH    UNIT
                STOVL   UNVEC           # UR SUB A
                        RPASS2
                UNIT    DOT
                        UNVEC
                PUSH    SL1
                STODL   CSTH
                DSQ     PDDL
                        DP1/4TH
                SR2     DSU
                SQRT    SL1
                PDVL    VCOMP
                VXV
                        RPASS2
                DOT     PDDL
                        UP1
                SIGN    STADR
                STOVL   SNTH
                        RPASS2
                PDVL    CALL
                        VPASS2
                        VECSHIFT
                STOVL   VVEC
                CLEAR
                        RVSW
                STCALL  RVEC
                        TIMETHET
                LXA,2   VSL*
                        RTX2
                        0,2
                STORE   18D
                DOT     SL1R
                        UNVEC
                PDVL    ABVAL           # OD = V SUB PV
                SL*     PDVL
                        0,2
                        RACT2
                ABVAL   PDDL            # 2D = LENGTH OF R SUB A
                DSU
```

```
        02D
STODL   DIFFALT         # DELTA H IN METERS             B+29
        R1A
NORM    PDDL            # 2 - R V**/MU                          04D
        X1
        R1
CALL
        SHIFTR1
SR1R    DDV
SL*     PUSH
        0       -5,1
DSU     PDDL            # A SUB A                       B+29    04D
        DIFFALT
SR2     DDV             # A SUB P                       B+31
        04D             #                               B+2
PUSH    SQRT            # A SUB P/A SUB A                       06D
DMPR    DMP
        06D
        00D
SL3R    PDDL            # V SUB AV METERS/CS            B+7     08D
        02D             # R SUB A MAGNITUDE             B+29
NORM    PDDL
        X1
        RTMU
SR1     DDV             # 2MU                           B+38
SL*     PDDL            # 2 MU/R SUBAA                  B+14    10D
        0       -5,1
        04D             # ASUBA                         B+29
NORM    PDDL
        X2
        RTMU
SR1     DDV
SL*     BDSU
        0       -6,2    # 2U/R - U/A            B+14 (METERS/CS)SQ
PDDL    DSQ             #                                       10D
        08D
BDSU    SQRT
PDVL    VXV             # SQRT(MU(2/R SUB A-1/A SUB A)-VSUBA2)  10D
        UP1
        UNVEC
UNIT    VXSC
        10D
PDVL    VXSC
        UNVEC
        08D
VAD     VSL1
STADR
STORE   VACT3
VSU
        VACT2
```

```
                STCALL  DELVEET2        # DELTA VCDH -- REFERENCE COORDINATES
                        SUBEXIT
```

```
# ***** COMPTGO *****
#
# SUBROUTINES USED
#       CLOKTASK
#       2PHSCHNG

                BANK    35
                SETLOC  CSI/CDH
                BANK

                EBANK=  RTRN

                COUNT*  $$/P3575

COMPTGO         EXTEND
                QXCH    RTRN
                CAF     ZERO
                TS      DISPDEX
                CAF     BIT2
                INHINT
                TC      WAITLIST
                EBANK=  WHICH
                2CADR   CLOKTASK

                TC      2PHSCHNG
                OCT     40036
                OCT     05024
                OCT     13000
                TC      RTRN
```

```
# GROUND TRACKING DETERMINATION PROGRAM P21
# PROGRAM DESCRIPTION
# MOD NO - 1
# MOD BY - N.M.NEVILLE
# FUNCTIONAL DECRIPTION-
#
# TO PROVIDE THE ASTRONAUT DETAILS OF THE LM OR CSM GROUND TRACK WITHOUT
# THE NEED FOR GROUND COMMUNICATION (REQUESTED BY DSKY).
# CALLING SEQUENCE -
#
# ASTRONAUT REQUEST THROUGH DSKY V37E21E
# SUBROUTINES CALLED-
#
# GOPERF4
# GOFLASH
# THISPREC
# OTHPREC
# LAT-LONG
# NORMAL EXIT MODES-
#
# ASTRONAUT REQUEST TROUGH DSKY TO TERMINATE PROGRAM V34E
# ALARM OR ABORT EXIT MODES-
#
# NONE
# OUTPUT -
#
# OCTAL DISPLAY OF OPTION CODE AND VEHICLE WHOSE GROUND TRACK IS TO BE
# COMPUTED
#       OPTION CODE     00002
#       THIS            00001
#       OTHER           00002
# DECIMAL DISPLAY OF TIME TO BE INTEGRATED TO HOURS , MINUTES , SECONDS
# DECIMAL DISPLAY OF LAT,LONG,ALT
# ERASABLE INITIALIZATION REQUIRED
# AXO           2DEC    4.652459653 E-5         RADIANS         %68-69 CONSTANTS"
#
# -AYO          2DEC    2.147535898 E-5         RADIANS
#
# AZO           2DEC    .7753206164             REVOLUTIONS
# FOR LUNAR ORBITS 504LM VECTOR IS NEEDED
#
# 504LM         2DEC    -2.700340600 E-5        RADIANS
#
# 504LM _2      2DEC    -7.514128400 E-4        RADIANS
#
# 504LM _4      2DEC    _2.553198641 E-4        RADIANS
#
# NONE
# DEBRIS
#
```

```
# CENTRALS-A,Q,L
# OTHER-THOSE USED BY THE ABOVE LISTED SUBROUTINES
# SEE LEMPREC,LAT-LONG
                SBANK=  LOWSUPER        # FOR LOW 2CADR'S.

                BANK    33
                SETLOC  P20S
                BANK

                EBANK=  P21TIME
                COUNT*  $$/P21
PROG21          CAF     ONE
                TS      OPTION2         # ASSUMED VEHICLE IS LM , R2 = 00001
                CAF     BIT2            #  OPTION 2
                TC      BANKCALL
                CADR    GOPERF4
                TC      GOTOPOOH        # TERMINATE
                TC      +2              # PROCEED VALUE OF ASSUMED VEHICLE OK
                TC      -5              # R2 LOADED THROUGH DSKY
P21PROG1        CAF     V6N34           # LOAD DESIRED TIME OF LAT-LONG.
                TC      BANKCALL
                CADR    GOFLASH
                TC      GOTOPOOH        # TERM
                TC      +2              # PROCEED VALUES OK
                TC      -5              # TIME LOADED THROUGH DSKY
                TC      INTPRET
                DLOAD
                        DSPTEM1
                STCALL  TDEC1           # INTEGRATE TO TIME SPECIFIED IN TDEC
                        INTSTALL
                BON     CLEAR
                        P21FLAG
                        P21CONT         # ON---RECYCLE USING BASE VECTOR
                        VINTFLAG        # OFF--IST PASS CALL BASE VECTOR
                SLOAD   SR1
                        OPTION2
                BHIZ    SET
                        +2              # ZERO--THIS VEHICLE(LM)
                        VINTFLAG        # ONE--OTHER VEHICLE(CM)
                CLEAR   CLEAR
                        DIMOFLAG
                        INTYPFLG        # PRECISION
                CALL
                        INTEGRV         # CALCULATE
                GOTO                    # -AND
                        P21VSAVE        # -SAVE BASE VECTOR
P21CONT         VLOAD
                        P21BASER        # RECYCLE--INTEG FROM BASE VECTOR
                STOVL   RCV             # --POS
```

```
                                P21BASEV
                STODL           VCV                 # --VEL
                                P21TIME
                STORE           TET                 # --TIME
                CLEAR           CLEAR
                                DIMOFLAG
                                MOONFLAG
                SLOAD           BZE
                                P21ORIG
                                +3                  # ZERO=EARTH
                SET                                 # ---2=MOON
                                MOONFLAG
 +3             CALL
                                INTEGRVS
P21VSAVE        DLOAD                               # SAVE CURRENT BASEVECTOR
                                TAT
                STOVL           P21TIME             # --TIME
                                RATT1
                STOVL           P21BASER            # --POS B-29 OR B-27
                                VATT1
                STORE           P21BASEV            # --VEL B-07 OR B-05
                ABVAL           SL*
                                0,2
                STOVL           P21VEL              # VEL/ FOR N91 DISP
                                RATT
                UNIT            DOT
                                VATT                #  U(R).V
                DDV             ASIN                # U(R).U(V)
                                P21VEL
                STORE           P21GAM              # SIN-1 U(R).U(V) , -90 TO &90
                SXA,2           SLOAD
                                P21ORIG             # 0=EARTH
                                OPTION2
                SR1             BHIZ
                                +3
                GOTO
                                +4
 +3             BON
                                SURFFLAG
                                P21DSP
 +4             SET
                                P21FLAG
P21DSP          CLEAR           SLOAD               # GENERATE DISPLAY DATA
                                LUNAFLAG
                                X2
                BZE             SET
                                +2                  # 0 = EARTH
                                LUNAFLAG
                VLOAD
                                RATT
```

```
                STODL   ALPHAV
                        TAT
                CLEAR   CALL
                        ERADFLAG
                        LAT-LONG
                DMP                     # MPAC = ALT, METERS B-29
                        K.01
                STORE   P21ALT          # ALT/100 FOR N91 DISP
                EXIT
                CAF     V06N43          # DISPLAY LAT, LONG, ALT
                TC      BANKCALL        # LAT, LONG = 1/2 REVS B0
                CADR    GOFLASH         # ALT = KM B14
                TC      GOTOPOOH        # TERM
                TC      GOTOPOOH
                TC      INTPRET         # V32E RECYCLE
                DLOAD   DAD
                        P21TIME
                        600SEC          # 600 SECONDS OR 10 MIN
                STORE   DSPTEM1
                RTB
                        P21PROG1
600SEC          2DEC    60000           # 10 MIN

V06N43          VN      00643
V6N34           VN      00634
K.01            2DEC    .01
```

```
# TRANSFER PHASE INITITIATION (TPI) PROGRAMS (P34 AND P74)

# MOD NO -1                       LOG SECTION -- P32-P35, P72-P75
# MOD BY WHITE, P.                DATE: 1 JUNE 67
#
# PURPOSE
#
#      (1)     TO CALCULATE THE REQUIRED DELTA V AND OTHER INITIAL CONDITIONS
#              REQUIRED BY THE ACTIVE VEHICLE FOR EXECUTION OF THE TRANSFER
#              PHASE INITITATION (TPI) MANEUVER, GIVEN --

#              (A)     TIME OF IGNITION TIG (TPI) OR THE ELEVATION ANGLE (E) OF
#                      THE ACTIVE/PASSIVE VEHICLE LOS AT TIG (TPI).

#              (B)     CENTRAL ANGLE OF TRANSFER (CENTANG) FROM TIG (TPI) TO
#                      INTERCEPT TIME (TIG (TPF)).

#      (2)     TO CALCULATE TIG (TPI) GIVEN E OR E GIVEN TIG (TPI).

#      (3)     TO CALCULATE THESE PARAMETERS BASED UPON MANEUVER DATA
#              APPROVED AND KEYED INTO THE DSKY BY THE ASTRONAUT.

#      (4)     TO DISPLAY TO THE ASTRONAUT AND THE GROUND CERTAIN DEPENDENT
#              VARIABLES ASSOCIATED WITH THE MANEUVER FOR APPROVAL BY THE
#              ASTRONAUT/GROUND.

#      (5)     TO STORE THE TPI TARGET PARAMETERS FOR USE BY THE DESIRED
#              THRUSTING PROGRAM.
#
# ASSUMPTIONS

#      (1)     LM ONLY -- THIS PROGRAM IS BASED UPON PREVIOUS COMPLETION OF
#              THE CONSTANT DELTA ALTITUDE (CDH) PROGRAM (P33/P73).
#              THEREFORE --

#              (A)     AT A SELECTED TPI TIME (NOW IN STORAGE) THE LINE OF SIGHT
#                      BETWEEN THE ACTIVE AND PASSIVE VEHICLES WAS SELECTED TO BE
#                      A PRESCRIBED ANGLE (E) (NOW IN STORAGE) FROM THE
#                      HORIZONTAL PLANE DEFINED BY THE ACTIVE VEHICLE POSITION.

#              (B)     THE TIME BETWEEN CDH IGNITION AND TPI IGNITION WAS
#                      COMPUTED TO BE GREATER THAN 10 MINUTES.

#              (C)     THE VARIATION OF THE ALTITUDE DIFFERENCE BETWEEN THE
#                      ORBITS WAS MINIMIZED.

#              (D)     THE PERICENTER ALTITUDES OF ORBITS FOLLOWING CSI AND
#                      CDH WERE COMPUTED TO BE GREATER THAN 35,000 FT FOR LUNAR
```

```
#                       ORBIT OR 85 NM FOR EARTH ORBIT.

#               (E)     THE CSI AND CDH MANEUVERS WERE ASSUMED TO BE PARALLEL TO
#                       THE PLANE OF THE PASSIVE VEHICLE ORBIT.  HOWEVER, CREW
#                       MODIFICATION OF DELTA V (LV) COMPONENTS MAY HAVE RESULTED
#                       IN AN OUT-OF-PLANE MANEUVER.

#       (2)     STATE VECTOR UPDATED BY P27 ARE DISALLOWED DURING AUTOMATIC
#               STATE VECTOR UPDATING INITIATED BY P20 (SEE ASSUMPTION (4)).

#       (3)     THIS PROGRAM MUST BE DONE OVER A TRACKING STATION FOR REAL
#               TIME GROUND PARTICIPATION IN DATA INPUT AND OUTPUT.  COMPUTED
#               VARIABLES MAY BE STORED FOR LATER VERIFICATION BY THE GROUND.
#               THESE STORAGE CAPABILITIES ARE LIMITED ONLY TO THE PARAMETERS
#               FOR ONE THRUSTING MANEUVER AT A TIME EXCEPT FOR CONCENTRIC
#               FLIGHT PLAN MANEUVER SEQUENCES.

#       (4)     THE RENDEZVOUS RADAR MAY OR MAY NOT BE USED TO UPDATE THE LM
#               OR CSM STATE VECTORS FOR THIS PROGRAM.  IF RADAR USE IS
#               DESIRED THE RADAR WAS TURNED ON AND LOCKED ON THE CSM BY
#               PREVIOUS SELECTION OF P20.  RADAR SIGHTING MARKS WILL BE MADE
#               AUTOMATICALLY APPROXIMATELY ONCE A MINUTE WHEN ENABLED BY THE
#               TRACK AND UPDATE FLAGS (SEE P20).  THE RENDEZVOUS TRACKING
#               MARK COUNTER IS ZEROED BY THE SELECTION OF P20 AND AFTER EACH
#               THRUSTING MANEUVER.

#       (5)     THE ISS NEED NOT BE ON TO COMPLETE THIS PROGRAM.

#       (6)     THE OPERATION OF THE PROGRAM UTILIZES THE FOLLOWING FLAGS --
#
#                       ACTIVE VEHICLE FLAG -- DESIGNATES THE VEHICLE WHICH IS
#                       DOING RENDEZVOUS THRUSTING MANEUVERS TO THE PROGRAM WHICH
#                       CALCULATES THE MANEUVER PARAMETERS.  SET AT THE START OF
#                       EACH RENDEZVOUS PRE-THRUSTING PROGRAM.
#
#                       FINAL FLAG -- SELECTS FINAL PROGRAM DISPLAYS AFTER CREW HAS
#                       SELECTED THE FINAL MANEUVER COMPUTATION CYCLE.
#
#                       EXTERNAL DELTA V FLAG -- DESIGNATES THE TYPE OF STEERING
#                       REQUIRED FOR EXECUTION OF THIS MANEUVER BY THE THRUSTING
#                       PROGRAM SELECTED AFTER COMPLETION OF THIS PROGRAM.
#
#       (7)     ONCE THE PARAMETERS REQUIRED FOR COMPUTATION OF THE MANEUVER
#               HAVE BEEN COMPLETELY SPECIFIED, THE VALUE OF THE ACTIVE
#               VEHICLE CENTRAL ANGLE OF TRANSFER IS COMPUTED AND STURED.
#               THIS NUMBER WILL BE AVAILABLE FOR DISPLAY TO THE ASTRONAUT
#               THROUGH THE USE OF V06N52.
#
#               THE ASTRONAUT WILL CALL THIS DISPLAY TO VERIFY THAT THE
#               CENTRAL ANGLE OF TRANSFER OF THE ACTIVE VEHICLE IS NOT WITHIN
```

```
#               170 TO 190 DEGREES.  IF THE ANGLE IS WITHIN THIS ZONE THE
#               ASTRONAUT SHOULD REASSES THE INPUT TARGETING PARAMETERS BASED
#               UPON DELTA V AND EXPECTED MANEUVER TIME.
#
#       (8)     THIS PROGRAM IS SELECTED BY THE ASTRONAUT BY DSKY ENTRY --
#
#                       P34 IF THIS VEHICLE IS ACTIVE VEHICLE.
#
#                       P74 IF THIS VEHICLE IS PASSIVE VEHICLE.
#
# INPUT
#
#       (1)     TTPI    TIME OF THE TPI MANEUVER.
#       (2)     ELEV    DESIRED LOS ANGLE AT TPI
#       (3)     CENTANG ORBITAL CENTRAL ANGLE OF THE PASSIVE VEHICLE DURING
#                       TRANSFER FROM TPI TO TIME OF INTERCEPT
#
# OUTPUT
#
#       (1)     TRKMKCNT        NUMBER OF MARKS
#       (2)     TTOGO           TIME TO GO
#       (3)     +MGA            MIDDLE GIMBAL ANGLE
#       (4)     TTPI            COMPUTED TIME OF TPI MANEUVER
#                OR
#               ELEV            COMPUTED LOS ANGLE AT TPI
#       (5)     POSTTPI         PERIGEE ALTITUDE AFTER THE TPI MANEUVER
#       (6)     DELVTPI         MAGNITUDE OF DELTA V AT TPI
#       (7)     DELVTPF         MAGNITUDE OF DELTA V AT INTERCEPT
#       (8)     DVLOS           DELTA VELOCITY AT TPI -- LINE OF SIGHT
#       (9)     DELVLVC         DELTA VELOCITY AT TPI -- LOCAL VERTICAL COORDINATES
#
# DOWNLINK
#
#       (1)     TTPI            TIME OF TPI MANEUVER
#       (2)     TIG             TIME OF TPI MANEUVER
#       (3)     ELEV            DESIRED LOS ANGLE AT TPI
#       (4)     CENTANG         ORBITAL CENTRAL ANGLE OF THE PASSIVE VEHICLE DURING
#                               TRANSFER FROM TPI TO TIME OF INTERCEPT
#       (5)     DELVEET3        DELTA VELOCITY AT TPI -- REFERENCE COORDINATES
#       (6)     TPASS4          TIME OF INTERCEPT
#
# COMMUNICATION TO THRUSTING PROGRAMS
#
#       (1)     TIG             TIME OF THE TPI MANEUVER
#       (2)     RTARG           OFFSET TARGET POSITION
#       (3)     TPASS4          TIME OF INTERCEPT
#       (4)     XDELVFLG        RESET TO INDICATE LAMBERT (AIMPOINT) VG COMPUTATION
#
# SUBROUTINES USED
#
#       AVFLAGA
```

```
#		AVFLAGP
#		VNPOOH
#		DISPLAYE
#		SELECTMU
#		PRECSET
#		S33/34.1
#		ALARM
#		BANKCALL
#		GOFLASH
#		GOTOPOOH
#		TIMETHET
#		S34/35.2
#		PERIAPO1
#		SHIFTR1
#		S34/35.5
#		VN1645

		SETLOC	CSI/CDH
		BANK
		EBANK=	SUBEXIT
		COUNT*	$$/P3474
P34		TC	AVFLAGA
		TC	P34/P74A
P74		TC	AVFLAGP
P34/P74A	TC	P20FLGON	# SET UPDATFLG, TRACKFLG
		CAF	V06N37		# TTPI
		TC	VNPOOH
		EXTEND
		DCA	130DEG
		DXCH	CENTANG
		CAF	P30ZERO
		TS	NN
		TC	DISPLAYE	# ELEV AND CENTANG
		TC	INTPRET
		CLEAR	DLOAD
			ETPIFLAG
			TTPI
		STODL	TIG
			ELEV
		BZE	SET
			P34/P74B
			ETPIFLAG
P34/P74B	CALL
			SELECTMU
DELELO		EQUALS	26D
P34/P74C	DLOAD	SET
			ZEROVECS
			ITSWICH
		BON	CLEAR
			ETPIFLAG
```

```
                                SWCHSET
                                ITSWICH
SWCHSET             STORE       NOMTPI
INTLOOP             DLOAD       DAD
                                TTPI
                                NOMTPI
                    STCALL      TDEC1
                                PRECSET
                    CALL
                                S33/34.1
                    BZE         EXIT
                                SWCHCLR
                    TC          ALARM
                    OCT         611
                    CAF         V05N09
                    TC          BANKCALL
                    CADR        GOFLASH
                    TC          GOTOPOOH
                    TC          P34/P74A        # PROCEED
                    TC          -7              # V32

SWCHCLR             BONCLR      BON
                                ITSWICH
                                INTLOOP
                                ETPIFLAG
                                P34/P74D        # DISPLAY TTPI
                    EXIT
                    TC          DISPLAYE        # DISPLAY ELEV AND CENTANG
                    TC          P34/P74E
P34/P74D            EXIT
                    CAF         V06N37          # TTPI
                    TC          VNPOOH
P34/P74E            TC          INTPRET
                    SETPD       DLOAD
                                OD
                                RTX1
                    STODL       X1
                                CENTANG
                    PUSH        COS
                    STODL       CSTH
                    SIN
                    STOVL       SNTH
                                RPASS3
                    VSR*
                                0,2
                    STOVL       RVEC
                                VPASS3
                    VSR*        SET
                                0,2
                                RVSW
```

```
                STCALL  VVEC
                        TIMETHET
                DLOAD
                        TTPI
                STORE   INTIME          # FOR INITVEL
                DAD
                        T               # RENDEZVOUS TIME
                STCALL  TPASS4          # FOR INITVEL
                        S34/35.2
                VLOAD   ABVAL
                        DELVEET3
                STOVL   DELVTPI
                        VPASS4
                VSU     ABVAL
                        VTPRIME
                STOVL   DELVTPF
                        RACT3
                PDVL    CALL
                        VIPRIME
                        PERIAPO1
                CALL
                        SHIFTR1
                STODL   POSTTPI
                        TTPI
                STORE   TIG
                EXIT
                CAF     V06N58
                TC      VNPOOH
                TC      INTPRET
                CALL
                        S34/35.5
                CALL
                        VN1645
                GOTO
                        P34/P74C
```

```
# RENDEZVOUS MID-COURSE MANEUVER PROGRAMS (P35 AND P75)

# MOD NO -1                      LOG SECTION -- P32-P35, P72-P75
# MOD BY WHITE, P.               DATE:  1 JUNE 67
#
# PURPOSE
#
#       (1)     TO CALCULATE THE REQUIRED DELTA V AND OTHER INITIAL CONDITIONS
#               REQUIRED BY THE ACTIVE VEHICLE FOR EXECUTION OF THE NEXT
#               MID-COURSE CORRECTION OF THE TRANSFER PHASE OF AN ACTIVE
#               VEHICLE RENDEZVOUS.
#
#       (2)     TO DISPLAY TO THE ASTRONAUT AND THE GROUND CERTAIN DEPENDENT
#               VARIABLES ASSOCIATED WITH THE MANEUVER FOR APPROVAL BY THE
#               ASTRONAUT/GROUND.
#
#       (3)     TO STORE THE TPM TARGET PARAMETERS FOR USE BY THE DESIRED
#               THRUSTING PROGRAM.
#
# ASSUMPTIONS
#
#       (1)     THE ISS NEED NOT BE ON TO COMPLETE THIS PROGRAM.
#
#       (2)     STATE VECTOR UPDATES BY P27 ARE DISALLOWED DURING AUTOMATIC
#               STATE VECTOR UPDATING INITIATED BY P20 (SEE ASSUMPTION (3)).
#
#       (3)     THE RENDEZVOUS RADAR IS ON AND IS LOCKED ON THE CSM.  THIS WAS
#               DONE DURING PREVIOUS SELECTION OF P20.  RADAR SIGHTING MARKS
#               WILL BE MADE AUTOMATICALLY APPROXIMATELY ONCE A MINUTE WHEN
#               ENABLED BY THE TRACK AND UPDATE FLAGS (SEE P20).  THE
#               RENDEZVOUS TRACKING MARK COUNTER IS ZEROED BY THE SELECTION OF
#               P20 AND AFTER EACH THRUSTING MANEUVER.
#
#       (4)     THE OPERATION OF THE PROGRAM UTILIZES THE FOLLOWING FLAGS --
#
#                       THE ACTIVE VEHICLE FLAG -- DESIGNATES THE VEHICLE WHICH IS
#                       DOING RENDEZVOUS THRUSTING MANEUVERS TO THE PROGRAM WHICH
#                       CALCULATES THE MANEUVER PARAMETERS.  SET AT THE START OF
#                       EACH RENDEZVOUS PRE-THRUSTING PROGRAM.
#
#                       FINAL FLAG -- SELECTS FINAL PROGRAM DISPLAYS AFTER CREW HAS
#                       SELECTED THE FINAL MANEUVER COMPUTATION CYCLE.
#
#                       EXTERNAL DELTA V FLAG -- DESIGNATES THE TYPE OF STEERING
#                       REQUIRED FOR EXECUTION OF THIS MANEUVER BY THE THRUSTING
#                       PROGRAM SELECTED AFTER COMPLETION OF THIS PROGRAM.
#
#       (5)     THE TIME OF INTERCEPT (T(INT)) WAS DEFINED BY PREVIOUS
#               COMPLETION OF THE TRANSFER PHASE INITIATION (TPI) PROGRAM
#               (P34/P74) AND IS PRESENTLY AVAILABLE IN STORAGE.
#
```

```
#       (6)     ONCE THE PARAMETERS REQUIRED FOR COMPUTATION OF THE MANEUVER
#               HAVE BEEN COMPLETELY SPECIFIED, THE VALUE OF THE ACTIVE
#               VEHICLE CENTRAL ANGLE OF TRANSFER IS COMPUTED AND STORED.
#               THIS NUMBER WILL BE AVAILABLE FOR DISPLAY TO THE ASTRONAUT
#               THROUGH THE USE OF V06N52
#
#               THE ASTRONAUT WILL CALL THIS DISPLAY TO VERIFY THAT THE
#               CENTRAL ANGLE OF TRANSFER OF THE ACTIVE VEHICLE IS NOT WITHIN
#               170 TO 190 DEGREES.  IF THE ANGLE IS WITHIN THIS ZONE THE
#               ASTRONAUT SHOULD REASSESS THE INPUT TARGETING PARAMETERS BASED
#               UPON DELTA V AND EXPECTED MANEUVER TIME.
#
#       (7)     THIS PROGRAM IS SELECTED BY THE ASTRONAUT BY DSKY ENTRY --
#
#                       P35 IF THIS VEHICLE IS ACTIVE VEHICLE.
#
#                       P75 IF THIS VEHICLE IS PASSIVE VEHICLE.
#
# INPUT
#
#       (1)     TPASS4          TIME OF INTERCEPT -- SAVED FROM P34/P74
#
# OUTPUT
#
#       (1)     TRKMKCNT        NUMBER OF MARKS
#       (2)     TTOGO           TIME TO GO
#       (3)     +MGA            MIDOLF GIMBAL ANGLE
#       (4)     DVLOS           DELTA VELOCITY AT MID -- LINE OF SIGHT
#       (5)     DELVLVC         DELTA VELOCITY AT MID -- LOCAL VERTICAL COORDINATES
#
# DOWNLINK
#
#       (1)     TIG             TIME OF THE TPM MANEUVER
#       (2)     DELVEET3        DELTA VELOCITY AT TPM -- REFERENCE COORDINATES
#       (3)     TPASS4          TIME OF INTERCEPT
#
# COMMUNICATION TO THRUSTING PROGRAMS
#
#       (1)     TIG             TIME OF THE TPM MANEUVER
#       (2)     RTARG           OFFSET TARGET POSITION
#       (3)     TPASS4          TIME OF INTERCEPT
#       (4)     XDELVFLG        RESET TO INDICATE LAMBERT (AIMPOINT) VG COMPUTATION.
#
# SUBROUTINES USED
#
#       AVFLAGA
#       AVFLAGP
#       LOADTIME
#       SELECTMU
#       PRECSET
#       S34/35.1
#       S34/35.2
```

```
#       S34/35.5
#       VN1645

                COUNT*  $$/P3575
                EBANK=  KT

P35             TC      AVFLAGA
                EXTEND
                DCA     ATIGINC
                TC      P35/P75A
P75             TC      AVFLAGP
                EXTEND
                DCA     PTIGINC
P35/P75A        DXCH    KT
                TC      P20FLGON        # SET UPDATFLG, TRACKFLG
                TC      INTPRET
                CALL
                        SELECTMU
P35/P75B        RTB
                        LOADTIME
                STORE   TSTRT
                DAD
                        KT
                STORE   TIG
                STORE   INTIME          # FOR INITVEL
                STCALL  TDEC1
                        PRECSET         # ADVANCE BOTH VEHICLES
                CALL
                        S34/35.1        # GET NORM AND LOS FOR TRANSFORM
                CALL
                        S34/35.2        # GET DELTA V(LV)
                CALL
                        S34/35.5
                CALL
                        VN1645
                GOTO
                        P35/P75B
```

```
# ***** S33/34.1 *****

S33/34.1        STQ     SSP
                        NORMEX
                        TITER
                OCT     40000
                DLOAD   SETPD
                        MAX250
                        OD
                STOVL   SECMAX
                        RACT3
                STOVL   RAPREC
                        VACT3
                STOVL   VAPREC
                        RPASS3
                STOVL   RPPREC
                        VPASS3
                STORE   VPPREC
ELCALC          CALL
                        S34/35.1        # NORMAL AND LOS
                VXV     PDVL
                        RACT3           # (RA*VA)*RA OD
                PDVL    UNIT            # ULOS AT 6D
                        RACT3
                PDVL    VPROJ           # XCHNJ AND UP
                VSL2    BVSU
                        ULOS
                UNIT    PDVL            # UP AT OD
                DOT     PDVL            # UP.UN*RA AT OD
                        OD              # UP IN MPAC
                DOT     SIGN
                        ULOS
                SL1     ACOS
                PDVL    DOT             # EA AT OD
                        ULOS
                        RACT3
                BPL     DLOAD
                        TESTY
                        DPPOSMAX
                DSU     PUSH
TESTY           BOFF    DLOAD
                        ITSWICH
                        ELEX
                        DELEL
                STODL   DELELO
                DSU
                        ELEV
                STORE   DELEL
                ABS     DSU
                        ELEPS
```

```
                BMN
                        TIMEX           # COMMERCIALS EVERYWHERE
FIGTIME         SLOAD   SR1
                        TITER
                BHIZ    LXA,1
                        NORMEX          # TOO MANY ITERATIONS
                        MPAC
                SXA,1   VLOAD
                        TITER
                        RPASS3
                UNIT    PDDL
                        36D
                PDVL    UNIT
                        RACT3
                PDDL
                PDDL    PUSH
                        36D
                BDSU
                        12D
                STODL   30D             # RP - RA MAGNITUDES
                        DPHALF
                DSU     PUSH
                        ELEV
                SIGN    BMN
                        30D
                        NORMEX
                DLOAD   COS
                DMP     DDV
                        14D
                        12D
                DCOMP                   # SINCE COS(180-A)=-COS A
                STORE   28D
                ABS     BDSU
                        DPHALF
                BMN     VLOAD
                        NORMEX
                        UNRM
                VXV     UNIT
                        6D              # UN*RA
                DOT     DMP
                        VACT3
                        12D
                PDVL    VXV
                        0D
                        VPASS3
                VXV     UNIT
                        0D              # (RP*VP)*RP
                DOT     DMP
                        VPASS3
                        14D
```

```
                BDSU
                NORM    PDVL            # NORMALIZED WA - WP 12D
                        X1
                        6D
                VXV     DOT
                        OD
                        UNRM            # RA*RP.UN 14D
                PDVL    DOT
                        OD
                        6D
                SL1     ACOS
                SIGN
                DSU     DAD             # ALPHA PI
                        DPHALF
                        ELEV
                PDDL    ACOS
                        28D
                BDSU    SIGN
                        DPHALF
                        30D             # CONTAINS RP-RA
                DAD
                DMP     DDV
                        TWOPI
                DMP
                SL*     DMP
                        0       -3,1
                PUSH    ABS
                DSU     BMN
                        SECMAX
                        OKMAX
                DLOAD   SIGN            # REPLACE TIME WITH MAX TIME SIGNED
                        SECMAX
                PUSH
OKMAX           SLOAD   BPL             # TEST FIRST ITERATION
                        TITER
                        REPETE
                SSP     DLOAD
                        TITER
                OCT     37777
                GOTO
                        STORDELT
REPETE          DLOAD   DMP
                        DELEL
                        DELELO
                BPL     DLOAD
                        NEXTES
                        SECMAX
                DMP
                        THIRD
                STODL   SECMAX
```

```
                ABS      SR1            # CROSSED OVER SOLUTION
                DCOMP    GOTO           # DT=(-SIGN(DTO)//DT//)/2
                         RESIGN
NEXTES          DLOAD    ABS
                         DELEL
                PDDL     ABS
                         DELELO
                DSU
                BMN      DLOAD
                         REVERS         # WRONG DIRECTION
                ABS
RESIGN          SIGN     GOTO
                         DELTEEO
                         STORDELT
REVERS          DLOAD    DCOMP
                         DELTEEO
                PUSH     SR1
                STORE    DELTEEO
                DAD
                GOTO
                         ADTIME
STORDELT        STORE    DELTEEO
ADTIME          DAD
                         NOMTPI         # SUM OF DELTA T'S
                STORE    NOMTPI
                VLOAD    PDVL
                         VAPREC
                         RAPREC
                CALL
                         GOINT
                CALL
                         ACTIVE         # STORE NEW RACT3 VACT3
                VLOAD    PDVL
                         VPPREC
                         RPPREC
                CALL
                         GOINT
                CALL
                         PASSIVE        # STORE NEW RPASS3 VPASS3
                GOTO
                         ELCALC
ELEX            DLOAD    DAD
                         TTPI
                         NOMTPI
                STODL    TTPI
                BON
                         ETPIFLAG
                         TIMEX
                STORE    ELEV
TIMEX           DLOAD    GOTO
```

ZEROVECS
NORMEX

```
# ***** S34/35.1 *****

# COMPUTE UNIT NORMAL AND LINE OF SIGHT VECTORS GIVEN THE ACTIVE AND
# PASSIVE POS AND VEL AT TIME T3

S34/35.1        VLOAD   VSU
                        RPASS3
                        RACT3
                UNIT    PUSH
                STOVL   ULOS
                        RACT3
                VXV     UNIT
                        VACT3
                STORE   UNRM
                RVQ
```

```
# ***** S34/35.2 *****

# ADVANCE PASSIVE VEH TO RENDEZVOUS TIME AND GET REQ VEL FROM LAMBERT

S34/35.2        STQ     VLOAD
                        SUBEXIT
                        VPASS3
                PDVL    PDDL
                        RPASS3
                        INTIME
                PDDL    PDDL
                        TPASS4
                        TWOPI           # CONIC
                PDDL    BHIZ
                        NN
                        S3435.23
                DLOAD
                DLOAD   PUSH
                        ZEROVECS        # PRECISION
S3435.23        CALL
                        INTINT          # GET TARGET VECTOR
S3435.25        STOVL   RTARG
                        VATT
                STOVL   VPASS4
                        RTARG
# COMPUTE PHI = PI + (ACOS(UNIT RA.UNIT RP) - PI)SIGN(RA*RP.U)
                UNIT    PDVL            # UNIT RP
                        RACT3
                UNIT    PUSH            # UNIT RA
                VXV     DOT
                        OD
                        UNRM            # RA*RP.U
                PDVL
                DOT     SL1             # UNIT RA.UNIT RP
                        OD
                ACOS    SIGN
                BPL     DAD
                        NOPIE
                        DPPOSMAX        # REASONABLE TWO PI
NOPIE           STODL   ACTCENT
                        TPASS4
                DSU
                        INTIME
                STORE   DELLT4
                SLOAD   SETPD
                        NN              # NUMBER OF OFFSETS
                        OD
                PDDL    PDVL
                        EPSFOUR
                        RACT3
                STOVL   RINIT
```

```
                        VACT3
                STCALL  VINIT
                        INITVEL
                CALL
                        LOMAT
                VLOAD   MXV
                        DELVEET3
                        OD
                VSL1
                STCALL  DELVLVC
                        SUBEXIT
```

```
# ***** S34/35.3 *****

S34/35.3        STQ     CALL
                        NORMEX
                        LOMAT           # GET MATRIX IN PUSH LIST
                VLOAD   VXM
                        DELVLVC         # NEW DEL V TPI
                        OD
                VSL1
                STORE   DELVEET3        # SAVE FOR TRANSFORM
                VAD     PDVL
                        VACT3           # NEW V REQ
                        RACT3
                PDDL    PDDL
                        TIG
                        TPASS4
                PDDL    PUSH
                        DPPOSMAX
                CALL                    # INTEG. FOR NEW TARGET VEC
                        INTINT
                VLOAD
                        RATT
                STORE   RTARG
NOVRWRT         VLOAD   PUSH
                        ULOS
                VXV     VCOMP
                        UNRM
                UNIT    PUSH
                VXV     VSL1
                        ULOS
                PDVL
                PDVL    MXV
                        DELVEET3
                        OD
                VSL1
                STCALL  DVLOS
                        NORMEX
```

```
# ***** S34/35.4 *****

S34/35.4        STQ     SETPD           # NO ASTRONAUT OVERWRITE
                        NORMEX
                        OD
                GOTO
                        NOVRWRT
```

```
# ***** LOMAT *****

LOMAT           VLOAD   VCOMP
                        UNRM
                STOVL   6D              # Y
                        RACT3
                UNIT    VCOMP
                STORE   12D
                VXV     VSL1
                        UNRM            # Z*-Y
                STORE   0D
                SETPD   RVQ
                        18D
GOINT           PDDL    PDDL            # DO
                        ZEROVECS        #       NOT
                        NOMTPI          #
                PUSH    PUSH            #               ORDER OR INSERT BEFORE INTINT
INTINT          STQ     CALL
                        RTRN
                        INTSTALL
                CLEAR   DLOAD
                        INTYPFLG
                BZE     SET
                        +2
                        INTYPFLG
                DLOAD   STADR
                STODL   TDEC1
                SET     LXA,2
                        MOONFLAG
                        RTX2
                BON     CLEAR
                        CMOONFLG
                        ALLSET
                        MOONFLAG
ALLSET          STOVL   TET
                VSR*
                        0,2
                STOVL   RCV
                VSR*
                        0,2
                STCALL  VCV
                        INTEGRVS
                VLOAD   GOTO
                        RATT
                        RTRN
```

```
# ***** S34/35.5 *****
# SUBROUTINES USED
#       BANKCALL
#       GOFLASH
#       GOTOPOOH
#       S34/35.3
#       S34.35.4
#       VNPOOH

S34/35.5        STQ     BON
                        SUBEXIT
                        FINALFLG
                        FLAGON
                SET     GOTO
                        UPDATFLG
                        FLAGOFF
FLAGON          CLEAR   VLOAD
                        NTARGFLG
                        DELVLVC
                STORE   GDT/2
                EXIT
 +5             CAF     V06N81
                TC      BANKCALL
                CADR    GOFLASH
                TC      GOTOPOOH
                TC      +2              # PRO
                TC      FLAGON  +5      # LOAD
 +2             CA      EBANK7
                TS      EBANK           # TO BE SURE

                ZL
                CA      FIVE
NTARGCHK        TS      Q
                INDEX   Q
                CS      DELVLVC
                INDEX   Q
                AD      GDT/2
                ADS     L
                CCS     Q
                TCF     NTARGCHK
                LXCH    A
                EXTEND
                BZF     +3
                TC      UPFLAG
                ADRES   NTARGFLG

                TC      INTPRET
                BOFF    CALL
                        NTARGFLG
```

```
                                NOCHG
                                S34/35.3
NOCHG               CLEAR       VLOAD
                                XDELVFLG
                                DELVEET3
                    STORE       DELVSIN
FLAGOFF             CALL
                                S34/35.4
                    EXIT
                    CAF         V06N59
                    TC          VNPOOH
                    TC          INTPRET
                    GOTO
                                SUBEXIT
```

```
# ***** VN1645 *****
#
# SUBROUTINES USED
#       P3XORP7X
#       GET+MGA
#       BANKCALL
#       DELAYJOB
#       COMPTGO
#       GOFLASHR
#       GOTOPOOH
#       FLAGUP

VN1645          STQ     DLOAD
                        SUBEXIT
                        DP-.01
                STORE   +MGA            # MGA = -.01
                BOFF    DLOAD
                        FINALFLG
                        GET45
                        DP-.01
                DAD
                        DP-.01
                STORE   +MGA            # MGA = -.02
                BOFF    EXIT
                        REFSMFLG
                        GET45
                TC      P3XORP7X
                TC      +2              # P3X
                TC      GET45   +1      # P7X
                TC      INTPRET
                VLOAD   PUSH
                        DELVSIN
                CALL                    # COMPUTE MGA
                        GET+MGA
GET45           EXIT
                TC      COMPTGO         # INITIATE TASK TO UPDATE TTOGO
                CA      SUBEXIT
                TS      QSAVED
                CAF     1SEC
                TC      BANKCALL
                CADR    DELAYJOB
                CAF     V16N45          # TRKMKCNT, TTOGO, +MGA
                TC      BANKCALL
                CADR    GOFLASH
                TC      KILCLOCK        # TERMINATE
                TC      N45PROC         # PROCEED
                TC      CLUPDATE        # RECYCLE -- RETURN FOR INITIAL COMPUTATION
KILCLOCK        CA      Z
                TS      DISPDEX
```

```
                TC      GOTOPOOH
N45PROC         CS      FLAGWRD2
                MASK    BIT6
                EXTEND
                BZF     KILCLOCK        # FINALFLG IS SET -- FLASH V37 -- AWAIT NEW PGM
                TC      PHASCHNG
                OCT     04024
                TC      UPFLAG          # SET
                ADRES   FINALFLG        # FINALFLG
CLUPDATE        CA      Z
                TS      DISPDEX
                TC      PHASCHNG
                OCT     04024
                TC      INTPRET
                CLEAR   GOTO
                        UPDATFLG
                        QSAVED
```

```
# ***** DISPLAYE *****
#
# SUBROUTINES USED
#       BANKCALL
#       GOFLASHR
#       GOTOPOOH
#       BLANKET
#       ENDOFJOB

DISPLAYE        EXTEND
                QXCH    NORMEX
                CAF     V06N55
                TCR     BANKCALL
                CADR    GOFLASH
                TCF     GOTOPOOH
                TC      NORMEX
                TCF     -5
```

```
# ***** P3XORP7X *****

P3XORP7X        CAF     HIGH9
                MASK    MODREG
                EXTEND
                BZF     +2
                INCR    Q
                RETURN

# ***** VNPOOH *****
#
# SUBROUTINES USED
#       BANKCALL
#       GOFLASH
#       GOTOPOOH

VNPOOH          EXTEND
                QXCH    RTRN
                TS      VERBNOUN
                CA      VERBNOUN
                TCR     BANKCALL
                CADR    GOFLASH
                TCF     GOTOPOOH
                TC      RTRN
                TCF     -5
```

```
# ***** CONSTANTS *****

V06N37          VN      0637
V06N55          VN      0655
V06N58          VN      0658
V06N59          VN      0659
V06N81          VN      0681
V16N45          VN      1645
TWOPI           2DEC    6.283185307 B-4

MAX250          2DEC    25 E3

THIRD           2DEC    .333333333

ELEPS           2DEC    .27777777 E-3

DP-.01          OCT     77777           # CONSTANTS
                OCT     61337           # ADJACENT      -.01 FOR MGA DSP
EPSFOUR         2DEC    .0416666666

130DEG          2DEC    .3611111111
```

```
# ***** INITVEL *****

# MOD NO -1                        LOG SECTION -- P34-P35, P74-P75
# MOD BY WHITE, P.                 DATE:  21 NOV 67
#
# FUNCTIONAL
#
#       THIS SUBROUTINE COMPUTES THE REQUIRED INITIAL VELOCITY VECTOR FOR
#       A TRAJECTORY OF SPECIFIC TRANSFER TIME BETWEEN SPECIFIED INITIAL
#       AND TARGET POSITIONS.  THE TRAJECTORY MAY BE EITHER CONIC OR
#       PRECISION DEPENDING ON AN INPUT PARAMETER (NAMELY, NUMBER OF
#       OFFSETS).  IN ADDITION, IN THE PRECISION TRAJECTORY CASE, THE
#       SUBROUTINE ALSO COMPUTES AN OFFSET TARGET VECTOR, TO BE USED
#       DURING PURE-CONIC CROSS-PRODUCT STEERING.  THE OFFSET TARGET
#       VECTOR IS THE TERMINAL POSITION VECTOR OF A CONIC TRAJECTORY WHICH
#       HAS THE SAME INITIAL STATE AS A PRECISION TRAJECTORY WHOSE
#       TERMINAL POSITION VECTOR IS THE SPECIFIED TARGET VECTOR.
#
#       IN ORDER TO AVOID THE INHERENT SINGULARITIES IN THE 180 DEGREE
#       TRANSFER CASE WHEN THE (TRUE OR OFFSET) TARGET VECTOR MAY BE
#       SLIGHTLY OUT OF THE ORBITAL PLANE, THIS SUBROUTINE ROTATES THIS
#       VECTOR INTO A PLANE DEFINED BY THE INPUT INITIAL POSITION VECTOR
#       AND ANOTHER INPUT VECTOR (USUALLY THE INITIAL VELOCITY VECTOR),
#       WHENEVER THE INPUT TARGET VECTOR LIES INSIDE A CONE WHOSE VERTEX
#       IS THE ORIGIN OF COORDINATES, WHOSE AXIS IS THE 180 DEGREE
#       TRANSFER DIRECTION, AND WHOSE CONE ANGLE IS SPECIFIED BY THE USER.
#
#       THE LAMBERT SUBROUTINE IS UTILIZED FOR THE CONIC COMPUTATIONS AND
#       THE COASTING INTEGRATION SUBROUTINE IS UTILIZED FOR THE PRECISION
#       TRAJECTORY COMPUTATIONS.
#
# CALLING SEQUENCE
#
#       L       CALL
#       L+1             INITVEL
#       L+2     (RETURN -- ALWAYS)
#
# INPUT
#
#       (1)     RINIT           INITIAL POSITION RADIUS VECTOR
#       (2)     VINIT           INITIAL POSITION VELOCITY VECTOR
#       (3)     RTARG           TARGET POSITION RADIUS VECTOR
#       (4)     DELLT4          DESIRED TIME OF FLIGHT FROM RINIT TO RTARG
#       (5)     INTIME          TIME OF RINIT
#       (6)     OD              NUMBER OF ITERATIONS OF LAMBERT/INTEGRVS
#       (7)     2D              ANGLE TO 180 DEGREES WHEN ROTATION STARTS
#       (8)     RTX1            -2 FOR EARTH, -10D FOR LUNAR
#       (9)     RTX2            COORDINATE SYSTEM ORIGIN -- 0 FOR EARTH, 2 FOR LUNAR
#       PUSHLOC SET AT 4D
#
```

```
# OUTPUT
#
#       (1)     RTARG           OFFSET TARGET POSITION VECTOR
#       (2)     VIPRIME         MANEUVER VELOCITY REQUIRED
#       (3)     VTPRIME         VELOCITY AT TARGET AFTER MANEUVER
#       (4)     DELVEET3        DELTA VELOCITY REQUIRED FOR MANEUVER
#
# SUBROUTINES USED
#
#       LAMBERT
#       INTSTALL
#       INTEGRVS

                SETLOC  INTVEL
                BANK

                COUNT*  $$/INITV
INITVEL         SET                     # COGA GUESS NOT AVAILABLE
                        GUESSW
HAVEGUES        VLOAD   STQ
                        RTARG
                        NORMEX
                STORE   RTARG1
                ABVAL
                STORE   RTMAG
                SLOAD   BHIZ
                        RTX2
                        INITVEL1
                VLOAD   VSL2
                        RINIT           # B29
                STOVL   RINIT           # B27
                        VINIT           # B7
                VSL2
                STOVL   VINIT           # B5
                        RTARG1
                VSL2
                STORE   RTARG1
                ABVAL
                STORE   RTMAG

# INITIALIZATION

INITVEL1        SSP     DLOAD           # SET ITCTR TO -1,LOAD MPAC WITH E4 (PL 2D)
                        ITCTR
                        0       -1
                COSINE  SR1             # CALCULATE COSINE (E4) (+2)
                STODL   COZY4           # SET COZY4 TO COSINE (E4)          (PL OD)
                LXA,2   SXA,2
                        MPAC
                        VTARGTAG        # SET VTARGTAG TO OD (SP)
                VLOAD
```

```
                                RINIT
                        STOVL   R1VEC           # R1VEC EQ RINIT
                                RTARG1
                        STODL   R2VEC           # R2VEC EQ RTARG
                                DELLT4
                        STORE   TDESIRED        # TDESIRED EQ DELLT4
                        SETPD   VLOAD
                                OD              # INITIALIZE PL TO OD
                                RINIT           # MPAC EQ RINIT (+29)
                        UNIT    PUSH            # UNIT(RI) (+1)                 (PL 6D)
                        VXV     UNIT
                                VINIT           # MPAC EQ UNIT(RI) X VI (+8)
                        STOVL   UN
                                RTARG1
                        UNIT    DOT             # TEMP*RT.URI (+2)              (PL OD)
                        DAD     CLEAR
                                COZY4
                                NORMSW
                        STORE   COZY4
        INITVEL2        BPL     SET
                                INITVEL3        # UN CALCULATED IN LAMBERT
                                NORMSW

# ROTATE RC INTO YC PLANE -- SET UNIT NORMAL TO YC

                        VLOAD   PUSH            #                               (PL 6D)
                                R2VEC           # RC TO 6D (+29)
                        ABVAL   PDVL            # RC TO MPAC, ABVAL(RC) (+29) TO OD (PL 2D)
                        PUSH    VPROJ           #                               (PL 8D)
                                UN
                        VSL2    BVSU
                        UNIT    VXSC            #                               (PL OD)
                        VSL1
                        STORE   R2VEC
                        TLOAD   SLOAD
                                ZEROVEC
                                ITCTR
                        BPL     VLOAD
                                INITVEL3
                                R2VEC
                        STORE   RTARG1
        INITVEL3        DLOAD   PDVL            #                               (PL 2D)
                                MUEARTH         # POSITIVE VALUE
                                R2VEC
                        UNIT    PDVL            # 2D = UNIT(R2VEC)              (PL 8D)
                                R1VEC
                        UNIT    PUSH            # 8D = UNIT(R1VEC)             (PL 14D)
                        VXV     VCOMP           # -N = UNIT(R2VEC) X UNIT(R1VEC)
                                2D
                        PUSH                    #                              (PL 20D)
                        LXA,1   DLOAD
```

```
                                RTX1
                                18D
                        BMN     INCR,1
                                +2
                        DEC     -8
                        INCR,1  SLOAD
                                10D
                                X1
                        BHIZ    VLOAD           #                                (PL 14D)
                                +2
                        VCOMP   PUSH            #                                (PL 20D)
                        VLOAD                   #                                (PL 14D)
                        VXV     DOT             #                                 (PL 2D)
                        BPL     DLOAD           #                                 (PL 0D)
                                INITVEL4
                        DCOMP   PUSH            #                                 (PL 2D)
        INITVEL4        LXA,2   SXA,2
                                OD
                                GEOMSGN

        # SET INPUTS UP FOR LAMBERT

                        LXA,1   CALL
                                RTX1

        #  OPERATE THE LAMBERT CONIC ROUTINE (COASTFLT SUBROUTINE)

                                SETITCTR        # GO TO END OF BANK TO SET ITERCTR BEFORE
                                                # CALLING LAMBER (FOR REMANUFACTURE ONLY)

        # ARRIVED AT SOLUTION IS GOOD ENOUGH ACCORDING TO SLIGHTLY WIDER BOUNDS.

                        CLEAR   VLOAD
                                GUESSW
                                VVEC

        # STORE CALCULATED INITIAL VELOCITY REQUIRED IN VIPRIME

                        STODL   VIPRIME         # INITIAL VELOCITY REQUIRED (+7)

        # IF NUMIT IS ZERO, CONTINUE AT INITVELB, OTHERWISE
        # SET UP INPUTS FOR ENCKE INTEGRATION (INTEGRVS).

                                VTARGTAG
                        BHIZ    CALL
                                INITVEL7
                                INTSTALL
                        SLOAD   CLEAR
                                RTX2
                                MOONFLAG
                        BHIZ    SET
                                INITVEL5
```

```
                                MOONFLAG
INITVEL5        VLOAD
                                RINIT
                STORE           R1VEC
                STOVL           RCV
                                VIPRIME
                STODL           VCV
                                INTIME
                STORE           TET
                DAD             CLEAR
                                DELLT4
                                INTYPFLG
                STCALL          TDEC1
                                INTEGRVS
                VLOAD
                                VATT1
                STORE           VTARGET

# IF ITERATION COUNTER (ITCTR) EQ NO. ITERATIONS (NUMIT), CONTINUE AT
# INITVELC, OTHERWISE REITERATE LAMBERT AND ENCKE

                LXA,2           INCR,2
                                ITCTR
                                1D              # INCREMENT ITCTR
                SXA,2           XSU,2
                                ITCTR
                                VTARGTAG
                SLOAD           BHIZ            # IF SP(MPAC) EQ 0, CONTINUE AT INITVELC
                                X2
                                INITVEL6

# OFFSET CONIC TARGET VECTOR

                VLOAD           VSU
                                RTARG1
                                RATT1
                VAD
                                R2VEC
                STODL           R2VEC
                                COZY4
                GOTO
                                INITVEL2        # CONTINUE ITERATING AT INITVEL2

# COMPUTE THE DELTA VELOCITY

INITVEL6        VLOAD
                                R2VEC
                STORE           RTARG1
INITVEL7        VLOAD           VSU
                                VIPRIME
                                VINIT
                STOVL           DELVEET3        # DELVEET3 = VIPRIME-VINIT (+7)
```

```
                                        VTARGET
                              STORE     VTPRIME
                              SLOAD     BHIZ
                                        RTX2
                                        INITVELX
                              VLOAD     VSR2
                                        VTPRIME
                              STOVL     VTPRIME
                                        VIPRIME
                              VSR2
                              STOVL     VIPRIME
                                        RTARG1
                              VSR2
                              STOVL     RTARG1
                                        DELVEET3
                              VSR2
                              STORE     DELVEET3
        INITVELX              LXA,1     DLOAD*
                                        RTX1
                                        MUTABLE -2,1
                              PUSH      DMP
                                        R1A
                              SR1       DDV
                                        R1
                              STODL     MU/A
                              SR
                                        6
                              STORE     MUASTEER
                              SETPD     VLOAD
                                        0D
                                        RTARG1
                              STCALL    RTARG
                                        NORMEX

# ***** END OF INITVEL ROUTINE *****
```

```
# ***** MIDGIM *****

# MOD NO. 0, BY WILLMAN, SUBROUTINE RENDGUID, LOG P34-P35, P74-P75
# REVISION 03, 17 FEB 67
#
# IF THE ACTIVE VEHICLE IS DOING THE COMPUTATION, MIDGIM COMPUTES
# THE POSITIVE MIDDLE GIMBAL ANGLE OF THE ACTIVE VEHICLE TO THE INPUT
# DELTA VELOCITY VECTOR (OD IN PUSH LIST), OTHERWISE
# MIDGIM CONVERTS THE INPUT DELTA VELOCITY VECTOR FROM INERTIAL COORDIN-
# ATES TO LOCAL VERTICAL COORDINATES OF THE ACTIVE VEHICLE.
#
# ** INPUTS **
#
#       NAME    MEANING                                         UNITS/SCALING/MODE
#
#       AVFLAG  INT FLAG -- 0 IS CSM ACTIVE, 1 IS LEM ACTIVE                    BIT
#       RINIT   ACTIVE VEHICLE RADIUS VECTOR                    METERS/CSEC (+7) VT
#       VINIT   ACTIVE VEHICLE VELOCITY VECTOR                  METERS/CSEC (+7) VT
#       OD(PL)  ACTIVE VEHICLE DELTA VELOCITY VECTOR            METERS/CSEC (+7) VT
#
# ** OUTPUTS **
#
#    NAME       MEANING                                         UNITS/SCALING/MODE
#
#    +MGA       + MIDDLE GIMBAL ANGLE                           REVOLUTIONS (+0) DP
#    DELVLVC    DELTA VELOCITY VECTOR IN LV COORD.              METERS/CSEC (+7) VT
#    MGLVFLAG   INT FLAG: 0 IS +MGA COMPUTED, 1 IS DELVLVC COMP.                BIT
#
# ** CALLING SEQUENCE **
#
#       L       CALL
#       L+1             MIDGIM
#       L+2     (RETURN -- ALWAYS)
#
# ** NO SUBROUTINES CALLED **
#
# ** DEBRIS -- ERASABLE TEMPORARY USAGE **
#
#       A,Q,L, PUSH LIST, MPAC.
#
# ** ALARMS -- NONE **
```

```
# MIDDLE GIMBAL ANGLE COMPUTATION.

                SETLOC  MIDDGIM
                BANK

                COUNT*  $$/MIDG

HALFREV         2DEC    1 B-1

GET+MGA         VLOAD   UNIT            # (PL OD) V (+7) TO MPAC UNITIZE UV (+1)
                UNIT
                DOT     SL1             # DOT UV WITH Y(STABLE MEMBER) AND RESCALE
                        REFSMMAT +6     # FROM +2 TO +1 FOR ASIN ROUTINE
                ARCSIN  BPL
                        SETMGA
                DAD     DAD             # CONVERT -MGA TO +MGA BY
                        HALFREV         # ADDING ONE REVOLUTION
                        HALFREV
SETMGA          STORE   +MGA
                CLR     RVQ             # CLEAR MGLVFLAG TO INDICATE +MGA CALC
                        MGLVFLAG        # AND EXIT
GET.LVC         VLOAD   UNIT            # (PL 6D) R (+29) IN MPAC UNITIZE UR
                        RINIT
                VCOMP                   # U(-R)
                STORE   18D             # U(-R) TO 18D
                VXV     UNIT            # U(-R)*V EQ V*U(R), U(V*R)
                        VINIT
                STORE   12D             # U(V*R) TO 12D
                VXV     UNIT            # U(V*R)*U(-R), U((V*R)*(-R))
                        18D
                STOVL   6D              # TRANSFORMATION MATRIX IS IN 6D (+1)
                        OD              # DELTA V (+7) IN OD
                MXV     VSL1            # CONVERT FROM INER COOR TO LV COOR (+8)
                        6D              # AND SCALE +7 IN MPAC
                STORE   DELVLVC         # STORE IN DELVLVC (+7)
                SET     RVQ             # SET MGLVFLAG TO INDICATE LVC CALC
                        MGLVFLAG        # AND EXIT

# ***** END OF MIDGIM ROUTINE *****
```

```
                BANK    10
                SETLOC  SLCTMU
                BANK
                COUNT*  $$/MIDG

SELECTMU        AXC,1   AXT,2
                        2D
                        0D
                BOFF
                        CMOONFLG
                        SETMUER
                AXC,1   AXT,2
                        10D
                        2D
SETMUER         DLOAD*  SXA,1
                        MUTABLE +4,1
                        RTX1
                STODL*  RTSR1/MU
                        MUTABLE -2,1
                BOFF    SR
                        CMOONFLG
                        RTRNMU
                        6D
RTRNMU          STORE   RTMU
                SXA,2   CLEAR
                        RTX2
                        FINALFLG
                GOTO
                        VN1645
```

```
# ***** PERIAPO *****
#
# MOD NO -1                          LOG SECTION -- P34-P35, P74-P75
# MOD BY WHITE, P.                   DATE 18 JAN 68
#
# FUNCTIONAL DESCRIPTION
#
#       THIS SUBROUTINE COMPUTES THE TWO BODY APOCENTER AND PERICENTER
#       ALTITUDES GIVEN THE POSITION AND VELOCITY VECTORS FOR A POINT ON
#       TRAJECTORY AND THE PRIMARY BODY.
#
#       SETRAD IS CALLED TO DETERMINE THE RADIUS OF THE PRIMARY BODY.
#
#       APSIDES IS CALLED TO SOVE FOR THE TWO BODY RADII OF APOCENTER AND
#       PERICENTER AND THE ECCENTRICITY OF THE TRAJECTORY.
#
# CALLING SEQUENCE
#
#       L       CALL
#       L+1             PERIAPO
#       L+2     (RETURN -- ALWAYS)
#
# INPUT
#
#       (1)     RVEC    POSITION VECTOR IN METERS
#                       SCALE FACTOR -- EARTH +29, MOON +27
#       (2)     VVEC    VELOCITY VECTOR IN METERS/CENTISECOND
#                       SCALE FACTOR -- EARTH +7, MOON +5
#       (3)     X1      PRIMARY BODY INDICATOR
#                       EARTH -1, MOON -10
#
# OUTPUT
#
#       (1)     2D      APOCENTER RADIUS IN METERS
#                       SCALE FACTOR -- EARTH +29, MOON +27
#       (2)     4D      APOCENTER ALTITUDE IN METERS
#                       SCALE FACTOR -- EARTH +29, MOON +27
#       (3)     6D      PERICENTER RADIUS IN METERS
#                       SCALE FACTOR -- EARTH +29, MOON +27
#       (4)     8D      PERICENTER ALTITUDE IN METERS
#                       SCALE FACTOR -- EARTH +29, MOON +27
#       (5)     ECC     ECCENTRICITY OF CONIC TRAJECTORY
#                       SCALE FACTOR -- +3
#       (6)     XXXALT  RADIUS OF THE PRIMARY BODY IN METERS
#                       SCALE FACTOR -- EARTH +29, MOON +27
#       (7)     PUSHLOC EQUALS 10D
#
# SUBROUTINES USED
#
#       SETRAD
```

```
#       APSIDES

                SETLOC  APOPERI
                BANK

                COUNT*  $$/PERAP

RPAD            2DEC    6373338 B-29     # STANDARD RADIUS OF PAD 37-B.
                                         # = 20 909 901.57 FT

PERIAPO1        LXA,2   VSR*
                        RTX2
                        0,2
                STOVL   VVEC
                LXA,1   VSR*
                        RTX1
                        0,2
                STORE   RVEC
PERIAPO         STQ     CALL
                        NORMEX
                        SETRAD
                STCALL  XXXALT
                        APSIDES
                SETPD   PUSH             # 2D = APOCENTER RADIUS          B29 OR B27
                        2D
                DSU     PDDL             # 4D = APOGEE ALTITUDE           B29 OR B27
                        XXXALT
                        OD
                PUSH    DSU              # 6D = PERICENTER RADIUS         B29 OR B27
                        XXXALT
                PUSH    GOTO             # 8D = PERIGEE ALTITUDE          B29 OR B27
                        NORMEX
```

```
SETRAD          DLOAD   PUSH
                        RPAD
                SXA,1   INCR,2
                        X2
                        2D
                SLOAD   BHIZ
                        X2
                        SETRADX
                VLOAD   ABVAL
                        RLS
                PDDL
SETRADX         DLOAD   RVQ
```

```
PRECSET         STQ
                        NORMEX
                STCALL  TDEC2
                        LEMPREC
                CALL
                        LEMSTORE
                DLOAD
                        TDEC2
                STCALL  TDEC1
                        CSMPREC
                CALL
                        CSMSTORE
                GOTO
                        NORMEX
LEMSTORE        VLOAD   BOFF
                        RATT
                        AVFLAG
                        PASSIVE
ACTIVE          STOVL   RACT3
                        VATT
                STORE   VACT3
                RVQ
CSMSTORE        VLOAD   BOFF
                        RATT
                        AVFLAG
                        ACTIVE
PASSIVE         STOVL   RPASS3
                        VATT
                STORE   VPASS3
                RVQ
```

```
VECSHIFT        LXA,2   VSR*
                        RTX2
                        0,2
                LXA,1   PDVL
                        RTX1
                VSR*    PDVL
                        0,2
                RVQ
```

```
SHIFTR1          LXA,2    SL*
                          RTX2
                          0,2
                 RVQ
```

```
# PROGRAM DESCRIPTION
# SUBROUTINE NAME       R36     OUT-OF-PLANE RENDEZVOUS ROUTINE
# MOD NO. 0             DATE    22 DECEMBER 67
# MOD BY N.M.NEVILLE    LOG SECTION EXTENDED VERBS
# FUNCTIONAL DESCRIPTION
#
# TO DISPLAY AT ASTRONAUT REQUEST LGC CALCULATED RENDEZVOUS
# OUT-OF-PLANE PARAMETERS (Y, YDOT, PSI).  (REQUESTED BY DSKY).
#
# CALLING SEQUENCE
#
#       ASTRONAUT REQUEST THROUGH DSKY V 90 E
#
# SUBROUTINES CALLED
#
#       EXDSPRET
#       GOMARKF
#       CSMPREC
#       LEMPREC
#       SGNAGREE
#       LOADTIME
#
# NORMAL EXIT MODES
#
#       ASTRONAUT REQUEST THROUGH DSKY TO TERMINATE PROGRAM V 34 E
#
# ALARM OR ABORT EXIT MODES
#
#       NONE
#
# OUTPUT
#
#       DECIMAL DISPLAY OF TIME, Y, YDOT AND PSI
#
#       DISPLAYED VALUES Y, YDOT, AND PSI, ARE STORED IN ERASABLE
#       REGISTERS RANGE, RRATE, AND RTHETA RESPECTIVELY.
#
# ERASABLE INITIALIZATION REQUIRED
#
#       CSM AND LEM STATE VECTORS
#
# DEBRIS
#
#       CENTRALS A,Q,L
#       OTHER:  THOSE USED BY THE ABOVE LISTED SUBROUTINES

                BANK    20
                SETLOC  R36LM
                BANK
```

```
                EBANK=  RPASS36

                COUNT*  $$/R36

R36             ZL
                CAF     ZERO            # SET TIME OF EVENT TO ZERO FOR FIRST
                DXCH    DSPTEMX         # DISPLAY
                CAF     V06N16N
                TC      BANKCALL
                CADR    GOMARKF
                TCF     ENDEXT          # TERMINATE
                TCF     +2              # PROCEED
                TCF     -5              # RECYCLE FOR ASTRONAUT INPUT TIME
                DXCH    DSPTEMX
                EXTEND
                BZF     LREGCHK         # A-REG ZERO GOTO CHECK L-REG FOR ZERO
ASTROTIM        DXCH    MPAC            # A-REG NON-ZERO, TIME = ASTRO INPUT TIME
                TC      INTPRET
                RTB
                        DPMODE
R36INT          STCALL  TDEC1
                        OTHPREC
                VLOAD   PDVL
                        VATT
                        RATT            # _
                STORE   RPASS36         # R
                UNIT    PDVL            #  P
                VXV     UNIT
                STADR                   # _
                STODL   UNP36           # U
                        TAT
                STCALL  TDEC1
                        THISPREC
                VLOAD   PDVL            #                  -
                        VATT            # VELOCITY VECTOR V             00D
                        RATT            #                  A
                PDDL
                        TAT             # SAVE TIME IN LOCATION 30D FOR REDISPLAY
                STOVL   30D             #                  _
                PUSH    PUSH            # POSITION VECTOR R  IN 06D AND 12D
                BVSU    PDVL            #                  A    _   _
                        RPASS36         # LINE OF SIGHT VECTOR R - R    12D
                DOT     SL1             #                       P   A
                        UNP36           #       _   _
                STOVL   RANGE           # Y = U . R
                        00D             #           A
                DOT     SL1
                        UNP36           # .     _   _
                STOVL   RRATE           # Y = U . V
                        06D             #           A
```

```
                                                  #  _         _
                UNIT     PUSH                     # U  = UNIT ( R  )               18D
                VXV      VXV                      #  RA          A
                         00D                      #   _    _     _     _
                         18D                      # (U  X V ) X U   = U
                VSL2     UNIT                     #   RA   A     RA    A
                UNIT
                STOVL    00D                      # UNIT HORIZONTAL IN FORWARD DIR. 00D
                         18D
                DOT      VXSC                     #  _
                         12D                      # U
                VSL2                              #  L
                BVSU     UNIT
                UNIT
                PUSH     DOT                      # LOS PROJECTED INTO HORIZONTAL  12D
                         00D                      # PLANE
                SL1      ARCCOS                   #               _   _
                STOVL    RTHETA                   # PSI = ARCCOS(U . U )
                VXV      DOT                      #                A   L
                         00D
                BPL      DLOAD
                         R36TAG2
                         LODPMAX
                DSU
                         RTHETA
                STORE    RTHETA
R36TAG2         DLOAD    RTB
                         30D
                         SGNAGREE
                STORE    DSPTEMX
                EXIT
                CAF      V06N90N                  # DISPLAY Y, YDOT, AND PSI.
                TC       BANKCALL
                CADR     GOMARKF
                TCF      ENDEXT                   # TERMINATE
                TCF      ENDEXT                   # PROCEED, END OF PROGRAM
                TCF      R36      +3              # REDISPLAY OUTPUT
LREGCHK         XCH      L
                EXTEND
                BZF      ENTTIM2                  # L-REG ZERO, SET TIME = PRESENT TIME
                XCH      L                        # L-REG NON ZERO, TIME = ASTRO INPUT TIME
                TCF      ASTROTIM
ENTTIM2         TC       INTPRET
                RTB      GOTO
                         LOADTIME
                         R36INT
V06N16N         VN       00616
V06N90N         VN       00690
```

```
                BANK    40
                SETLOC  R31LOC
                BANK

                COUNT*  $$/R31

R31CALL         CAF     PRIO3
                TC      FINDVAC
                EBANK=  SUBEXIT
                2CADR   V83CALL

DSPDELAY        TC      FIXDELAY
                DEC     100
                CA      EXTVBACT
                MASK    BIT12
                EXTEND
                BZF     DSPDELAY

                CAF     PRIO5
                TC      NOVAC
                EBANK=  TSTRT
                2CADR   DISPN5X

                TCF     TASKOVER

                BANK    37
                SETLOC  R31
                BANK
                COUNT*  $$/R31

DISPN5X         CAF     V16N54
                TC      BANKCALL
                CADR    GOMARKF
                TC      B5OFF
                TC      B5OFF
                TCF     DISPN5X

V83CALL         CS      FLAGWRD7        # TEST AVERAGE G FLAG
                MASK    AVEGFBIT
                EXTEND
                BZF     MUNG?           # ON - TEST MUNFLAG

                CS      FLAGWRD8
                MASK    SURFFBIT
                EXTEND
                BZF     ONEBASE         # ON SURFACE - BYPASS LEMPREC

                TC      INTPRET         # EXTRAPOLATE BOTH STATE VECTORS
                RTB
```

```
                        LOADTIME
                STCALL  TDEC1
                        LEMPREC         # PRECISION BASE VECTOR FOR LM
                VLOAD
                        RATT1
                STOVL   BASETHP
                        VATT1
                STODL   BASETHV
                        TAT
DOCMBASE        STORE   BASETIME        # PRECISION BASE VECTOR FOR CM
                STCALL  TDEC1
                        CSMPREC
                VLOAD
                        RATT1
                STOVL   BASEOTP
                        VATT1
                STORE   BASEOTV
                EXIT

REV83           CS      FLAGWRD7
                MASK    AVEGFBIT
                EXTEND
                BZF     GETRVN          # IF AVEGFLAG SET, USE RN,VN

                CS      FLAGWRD8
                MASK    SURFFBIT
                EXTEND
                BZF     R31SURF         # IF ON SURFACE,USE LEMAREC

                TC      INTPRET         # DO CONIC EXTRAPOLATION FOR BOTH VEHICLES
                RTB
                        LOADTIME
                STCALL  TDEC1
                        INTSTALL
                VLOAD   CLEAR
                        BASETHP
                        MOONFLAG
                STOVL   RCV
                        BASETHV
                STODL   VCV
                        BASETIME
                BOF     SET             # GET APPROPRIATE MOONFLAG SETTING
                        MOONTHIS
                        +2
                        MOONFLAG
                SET
                        INTYPFLG        # CONIC EXTRAP.
                STCALL  TET
                        INTEGRVS        # INTEGRATION --- AT LAST---
OTHCONIC        VLOAD
```

```
                        RATT
                STOVL   RONE
                        VATT
                STCALL  VONE            # GET SET FOR CONIC EXTRAP.,OTHER.
                        INTSTALL
                SET     DLOAD
                        INTYPFLG
                        TAT
OTHINT          STORE   TDEC1
                VLOAD   CLEAR
                        BASEOTP
                        MOONFLAG
                STOVL   RCV
                        BASEOTV
                STODL   VCV
                        BASETIME
                BOF     SET
                        MOONTHIS
                        +2
                        MOONFLAG
                STCALL  TET
                        INTEGRVS
COMPDISP        VLOAD   VSU
                        RATT
                        RONE
                RTB     PDDL
                        NORMUNX1        # UNIT(RANGE) TO PD 0-5
                        36D
                SL*                     # RESCALE AFTER NORMUNIT
                        0,1
                STOVL   RANGE           # SCALED 2(29)M
                        VATT
                VSU     DOT             # (VCM- VLM).UNIT(LOS). PD=0
                        VONE
                SL1                     # SCALED 2(7)M/CS
                STOVL   RRATE
                        RONE
                UNIT    PDVL            # UNIT(R) TO PD 0-5
                        UNITZ
                CALL
                        CDU*NBSM
                VXM     PUSH            # UNIT (Z)/4 TO PD 6-11
                        REFSMMAT
                VPROJ   VSL2            # UNIT(P)=UNIT(UZ -(UZ)PROJ(UR))
                        0D
                BVSU    UNIT
                        6D
                PDVL    VXV             # UNIT(P) TO PD 12-17
                        0D              # UNIT(RL)
                        VONE
```

```
                VXV     DOT             # (UR * VL)*UR . U(P)
                        0D
                        12D
                PDVL                    # SIGN TO 12-13 , LOAD U(P)
                DOT     SIGN
                        6D
                        12D
                SL2     ACOS            # ARCCOS(UP.UZ(SIGN))
                STOVL   RTHETA
                        0D
                DOT     BPL             # IF UR.UZ NEG,
                        6D              #   RTHETA = 1 - RTHETA
                        +5
                DLOAD   DSU
                        DPPOSMAX
                        RTHETA
                STORE   RTHETA
                EXIT

                CA      BIT5
                MASK    EXTVBACT
                EXTEND                  # IF ANSWERED,
                BZF     ENDEXT          #        TERMINATE

                CS      EXTVBACT
                MASK    BIT12
                ADS     EXTVBACT        # SET BIT 12
                TCF     REV83           # AND START AGAIN.

GETRVN          CA      PRIO22          # INHIBIT SERVICER
                TC      PRIOCHNG
                TC      INTPRET
                VLOAD   SETPD
                        RN              # LM STATE VECTOR IN RN,VN
                        0
                STOVL   RONE
                        VN
                STOVL   VONE            # LOAD R(CSM),V(CSM) IN CASE MUNFLAG SET
                        V(CSM)          # (TO INSURE TIME COMPATABILITY)
                PDVL    PDDL
                        R(CSM)
                        PIPTIME
                EXIT
                CA      PRIO3
                TC      PRIOCHNG
                TC      INTPRET
                BOFF    VLOAD
                        MUNFLAG
                        GETRVN2         # IF MUNFLAG RESET, DO CM DELTA PRECISION
```

```
                VXM     VSR4            # CHANGE TO REFERENCE SYSTEM AND RESCALE
                        REFSMMAT
                PDVL                    # R TO PD 0-5
                VXM     VSL1
                        REFSMMAT
                PUSH    SETPD           # V TO PD 5-11
                        0
                GOTO
                        COMPDISP

GETRVN2         CALL
                        INTSTALL
                CLEAR   GOTO
                        INTYPFLG        # PREC EXTRAP FOR OTHER
                        OTHINT
R31SURF         TC      INTPRET
                RTB                     # LM IS ON SURFACE, SO PRECISION
                        LOADTIME        # INTEGRATION USES PLANETARY INERTIAL
                STCALL  TDEC1           # ORIENTATION SUBROUTINE
                        LEMPREC
                GOTO                    # DO CSM CONIC
                        OTHCONIC
MUNG?           CS      FLAGWRD6
                MASK    MUNFLBIT
                EXTEND
                BZF     GETRVN          # IF MUNFLAG SET, CSM BASE NOT NEEDED

ONEBASE         TC      INTPRET         # GET CSM BASE VECTOR
                RTB     GOTO
                        LOADTIME
                        DOCMBASE

V16N54          VN      1654
```

```
# 1) PROGRAM NAME - TARGET DELTA V PROGRAM (P76).
# 2) FUNCTIONAL DESCRIPTION - UPON ENTRY BY ASTRONAUT ACTION, P76 FLASHES DSKY REQUESTS TO THE ASTRONAUT
#    TO PROVIDE VIA DSKY (1) THE DELTA V TO BE APPLIED TO THE OTHER VEHICLE STATE VECTOR AND (2) THE
#    TIME (TIG) AT WHICH THE OTHER VEHICLE VELOCITY WAS CHANGED BY EXECUTION OF A THRUSTING MANEUVER. THE
#    OTHER VEHICLE STATE VECTOR IS INTEGRATED TO TIG AND UPDATED BY THE ADDITION OF DELTA V (DELTA V HAVING
#    BEEN TRANSFORMED FROM LV TO REF COSYS).  USING INTEGRVS, THE PROGRAM THEN INTEGRATES THE OTHER
#    VEHICLE STATE VECTOR TO THE STATE VECTOR OF THIS VEHICLE, THUS INSURING THAT THE W-MATRIX AND BOTH VEHICLE
#    STATES CORRESPOND TO THE SAME TIME.
# 3) ERASABLE INIITIALIZATION REQUIRED - NONE.
# 4) CALLING SEQUENCES AND EXIT MODES - CALLED BY ASTRONAUT REQUEST THRU DSKY V 37 E 76 E.
#    EXITS BY TCF ENDOFJOB.
# 5) OUTPUT -- OTHER VEHICLE STATE VECTOR INTEGRATED TO TIG AND INCREMENTED BY DELTA V IN REF COSYS.
#    THE PUSHLIST CONTAINS THE MATRIX BY WHICH THE INPUT DELTA V MUST BE POST-MULTIPLIED TO CONVERT FROM LV
#    TO REF COSYS.
# 6) DEBRIS - OTHER VEHICLE STATE VECTOR.
# 7) SUBROUTINES CALLED - BANKCALL, GOXDSPF, CSMPREC (OR LEMPREC), ATOPCSM (OR ATOPLEM), INTSTALL, INTWAKE, PHASCHNG
#    INTPRET, INTEGRVS, AND MINIRECT.
# 8) FLAG USE - MOONFLAG, CMOONFLG, INTYPFLG, RASFLAG, AND MARKCTR.

                BANK    30
                SETLOC  P76LOC
                BANK

                COUNT*  $$/P76

                EBANK=  TIG

P76             TC      UPFLAG
                ADRES   TRACKFLG

                TC      INTPRET
                VLOAD
                        DELVLVC
                STORE   DELVOV
                EXIT

                CAF     V06N84          # FLASH LAST DELTA V,
                TC      BANKCALL        # AND WAIT FOR KEYBOARD ACTION.
                CADR    GOFLASH
                TCF     ENDP76
                TC      +2              # PROCEED
                TC      -5              # STORE DATA AND REPEAT FLASHING
                CAF     V06N84 +1       # FLASH VERB 06 NOUN 33, DISPLAY LAST TIG,
                TC      BANKCALL        # AND WAIT FOR KEYBOARD ACTION.
                CADR    GOFLASH
                TCF     ENDP76
                TC      +2
                TC      -5
                TC      INTPRET         # RETURN TO INTERPRETIVE CODE
```

```
                DLOAD                  # SET D(MPAC)=TIG IN CSEC B28
                        TIG
                STCALL  TDEC1          # SET TDEC1=TIG FOR ORBITAL INTEGRATION
                        OTHPREC
COMPMAT         VLOAD   UNIT
                        RATT
                VCOMP                  # U(-R)
                STORE   24D            # U(-R) TO 24D
                VXV     UNIT           # U(-R) XV = U(VXR)
                        VATT
                STORE   18D
                VXV     UNIT           # U(VXR)XU(-R) = U((RXV)XR)
                        24D
                STOVL   12D
                        DELVOV
                VXM     VSL1           # V(MPAC)=DELTA V IN REFCOSYS
                        12D
                VAD
                        VATT
                STORE   6              # V(PD6)=VATT + DELTA V
                CALL                   # PRESENT WOULD-BE USER OF ORBITAL
                        INTSTALL       # INTEG FROM INTERFERING WITH UPDATING
                CALL
                        P76SUB1
                VLOAD   VSR*
                        6
                        0,2
                STOVL   VCV
                        RATT
                VSR*
                        0,2
                STODL   RCV
                        TIG
                STORE   TET
                CLEAR   DLOAD
                        INTYPFLG
                        TETTHIS
INTOTHIS        STCALL  TDEC1
                        INTEGRVS
                CALL
                        INTSTALL
                VLOAD
                        RATT1
                STORE   RRECT
                STODL   RCV
                        TAT
                STOVL   TET
                        VATT1
                CALL
                        MINIRECT
```

```
                 EXIT
                 TC       PHASCHNG
                 OCT      04024

                 TC       UPFLAG
                 ADRES    REINTFLG

                 TC       INTPRET
                 CALL
                          ATOPOTH
                 SSP      EXIT
                          QPRET
                          OUT
                 TC       BANKCALL        # PERMIT USE OF ORBITAL INTEGRATION
                 CADR     INTWAKE1
OUT              EXIT
ENDP76           CAF      ZERO
                 TS       MARKCTR         # CLEAR RR TRACKING MARK COUNTER
                 TCF      GOTOPOOH

V06N84           NV       0684
                 NV       0633
P76SUB1          AXT,2    SET
                          2
                          MOONFLAG        # SET MEANS MOON IS SPHERE OF INFLUENCE.
                 BON      AXT,2
                          CMOONFLG        # SET MEANS PERM CM STATE IN LUNAR SPHERE.
                          QPRET
                          0
                 CLEAR    RVQ
                          MOONFLAG
```

```
# SUBROUTINE NAME:  V82CALL
# MOD NO: 0                                                                      DATE: 16 FEB 67
# MOD BY: R. R. BAIRNSFATHER                                                     LOG SECTION:  R30
# MOD NO: 1     MOD BY:  R. R. BAIRNSFATHER     DATE: 11 APR 67          SR30.1 CHANGED TO ALLOW MONITOR OPERN
# MOD NO: 2     MOD BY:  ALONSO                 DATE: 11 DEC 67          VB82 PROGRAM REWRITTEN
# MOD NO: 3     MOD BY:  ALONSO                 DATE: 26 MAR 68          PROG MOD TO HANDLE DIF EARTH/MOON SCALE
#
# NEW FUNCTIONAL DESCRIPTION:   CALLED BY VERB 82 ENTER.  PRIORITY 10.
# USED THROUGHOUT.              CALCULATE AND DISPLAY ORBITAL PARAMETERS
#
# 1.    IF AVERAGE G IS OFF:
#               FLASH DISPLAY V04N06. R2 INDICATES WHICH SHIP'S STATE VECTOR IS
#                TO BE UPDATED. INITIAL CHOICE IS THIS SHIP (R2=1). ASTRONAUT
#                CAN CHANGE TO OTHER SHIP BY V22EXE, WHERE X NOT EQ 1.
#               SELECTED STATE VECTOR UPDATED BY THISPREC (OTHPREC).
#               CALLS SR30.1 (WHICH CALLS TFFCONMU + TFFRP/RA) TO CALCULATE
#                RPER (PERIGEE RADIUS), RAPO (APOGEE RADIUS), HPER (PERIGEE
#                HEIGHT ABOVE LAUNCH PAD OR LUNAR LANDING SITE), HARD (APOGEE
#                HEIGHT AS ABOVE), TPER (TIME TO PERIGEE), TFF (TIME TO
#                INTERSECT 300 KFT ABOVE PAD OR 35KFT ABOVE LANDING SITE).
#               FLASH MONITOR V16N44 (HAPO, HPER, TFF).TFF IS -59M59S IF IT WAS
#                NOT COMPUTABLE, OTHERWISE IT INCREMENTS ONCE PER SECOND.
#                ASTRONAUT HAS OPTION TO MONITOR TPER BY KEYING IN N 32 E.
#                DISPLAY IS IN HMS, IS NEGATIVE (AS WAS TFF), AND INCREMENTS
#                ONCE PER SECOND ONLY IF TFF DISPLAY WAS -59M59S.
#
# 2.    IF AVERAGE G IS ON:
#               CALLS SR30.1 APPROX EVERY TWO SECS.  STATE VECTOR IS ALWAYS
#                FOR THIS VEHICLE. V82 DOES NOT DISTURB STATE VECTOR.  RESULTS
#                OF SR30.1 ARE RAPO, RPER, HAPO, HPER, TPER, TFF.
#               FLASH MONITOR V16N44 (HAPO, HPER, TFF).
# ADDENDUM: HAPO AND HPER SHOULD BE CHANGED TO READ HAPOX AND HPERX IN THE
#                ABOVE REMARKS.
#
# CALLING SEQUENCE:  VERB 82 ENTER.
#
# SUBROUTINES CALLED:  SR30.1, GOXDSPF
#                       MAYBE - THISPREC , OTHPREC, LOADTIME, DELRSPL
# NORMAL EXIT MODES:  TC ENDEXT
#
# ALARMS:  NONE
#
# OUTPUT:  HAPOX        (-29) M
#          HPERX        (-29) M
#          RAPO         (-29) M EARTH
#                       (-27) M MOON
#          RPER         (-29) M EARTH
#                       (-27) M MOON
#          TFF          (-28) CS        CONTAINS NEGATIVE QUANTITY
#          -TPER        (-28) CS        CONTAINS NEGATIVE QUANTITY
```

```
#
# ERASABLE INITIALIZATION REQUIRED: STATE VECTOR.
#
# DEBRIS:         QPRET, RONE, VONE,TFF/RTMU, HPERMIN, RPADTEM, V82EMFLG.
#                 MAYBE: TSTART82, V82FLAGS, TDEC1.

                EBANK=  HAPOX
                BANK    31
                SETLOC  R30LOC
                BANK
                COUNT*  $$/R30

V82CALL         TC      INTPRET
                BON     GOTO
                        AVEGFLAG
                        V82GON          # IF AVERAGE G ON
                        V82GOFF         # IF AVERAGE G OFF

V82GOFF         EXIT                    # ALLOW ASTRONAUT TO SELECT VEHICLE
                CAF     TWO             # DESIRED FOR ORBITAL PARAMETERS
                TS      OPTIONX         # CALCULATION AND DISPLAY.
                CAF     ONE
                TS      OPTIONX +1
                CAF     OPTIONVN        # V 04 N 06
                TC      BANKCALL
                CADR    GOXDSPF
                TC      ENDEXT          # TERMINATE
                TC      +2              # PROCEED
                TC      -5              # DATA IN. OPTION1+1 = 1 FOR THIS VEHICLE.
                                        #                UNEQ 1 FOR OTHER VEHICLE.
                CAF     BIT4            # 80 MS
                TC      WAITLIST
                EBANK=  TFF
                2CADR   TICKTEST

                RELINT
V82GOFLP        CAF     TFFBANK         # MAJOR RECYCLE LOOP ENTRY
                TS      EBANK
                CAF     ZERO
                TS      V82FLAGS        # ZERO FLAGS FOR TICKTEST, INHIBITS
                                        # DECREMENTING OF TFF AND -TPER.
                CAF     PRIO7
                TC      FINDVAC         # V82GOFF1 WILL EXECUTE STATE VECTOR
                EBANK=  TFF             # UPDATE AND ORBIT CALCULATIONS FOR
                2CADR   V82GOFF1        # SELECTED VEHICLE ABOUT PROPER BODY.

                RELINT
V82STALL        CAF     THREE           # STALL IN THIS LOOP AND WITHOLD V 16 N 44
```

```
                MASK    V82FLAGS        # UNTIL STATE VECTOR UPDATE SETS ONE OF
                CCS     A               # OUR FLAG BITS.
                TC      FLAGGON         # EXIT FROM STALL LOOP.
                CAF     1SEC
                TC      BANKCALL
                CADR    DELAYJOB
                TC      V82STALL

FLAGGON         CAF     V16N44          # MONITOR HAPO,HPER,TFF.
                TC      BANKCALL
                CADR    GOXDSPF
                TC      B5OFF           # TERM  THIS TELLS TICKTEST TO KILL ITSELF
                TC      B5OFF           # PROCEED  DITTO
                TC      V82GOFLP        # RECYCLE  RECOMPUTE STATE VECT + DISPLAY

OPTIONVN        VN      412
V16N44          VN      1644
TFFBANK         ECADR   TFF

V82GOFF1        TC      INTPRET
                RTB
                        LOADTIME
                STORE   TDEC1           # TIME FOR STATE VECTOR UPDATE.
                STORE   TSTART82        # TIME FOR INTERNAL USE.
                EXIT
                CS      OPTIONX +1      # 1 FOR THIS VEHICLE, NOT 1 FOR OTHER.
                AD      ONE
                EXTEND
                BZF     THISSHIP
OTHSHIP         TC      INTPRET
                CALL                    # CALL STATE VECTOR UPDATE FOR OTHER SHIP.
                        OTHPREC
BOTHSHIP        VLOAD                   # MOVE RESULTS INTO TFFCONIC STORAGE AREAS
                        RATT            #  TO BE CALLED BY SR30.1.
                STOVL   RONE            # RATT AT (-29)M FOR EARTH OR MOON
                        VATT
                STORE   VONE            #  VATT AT (-7)M/CS FOR EARTH OR MOON
                DLOAD*
                        1/RTMUE,2       #  X2 IS 0 FOR EARTH CENTERED STATE VEC
                STORE   TFF/RTMU        #  X2 IS 2 FOR MOON
                DLOAD*                  # AS LEFT BY THISPREC OR OTHPREC.
                        MINPERE,2
                STORE   HPERMIN         #   TFFRTMU, HPERMIN AND RPADTEM ARE ALL
                SLOAD   BHIZ            # EARTH/MOON PARAMETERS AS SET HERE.
                        X2
                        EARTHPAD
                GOTO
                        MOONPAD
```

```
THISSHIP        TC      INTPRET
                CALL                    # CALL STATE VECTOR UPDATE FOR THIS SHIP.
                        THISPREC
                GOTO
                        BOTHSHIP

# THE FOLLOWING CONSTANTS ARE PAIRWISE INDEXED. DO NOT SEPARATE PAIRS.

MINPERM         2DEC    10668 B-27      # 35 KFT MIN PERIGEE HEIGHT FOR MOON(-27)M

MINPERE         2DEC    91440 B-29      # 300 KFT (-29)M FOR EARTH

EARTHPAD        DLOAD   CLRGO           #  PAD 37-B RADIUS.  SCALED AT (-29)M.
                        RPAD
                        V82EMFLG        #  INDICATE EARTH SCALING FOR SR30.1
                        BOTHPAD

MOONPAD         VLOAD   ABVAL           # COMPUTE MOON PAD RADIUS FROM RLS VECTOR.
                        RLS             #  SCALED AT (-27)M.
                SET
                        V82EMFLG        #  INDICATE MOON SCALING FOR SR30.1
BOTHPAD         STCALL  RPADTEM
                        SR30.1          # CALCULATE ORBITAL PARAMETERS
                RTB     DSU
                        LOADTIME
                        TSTART82        # PRESENT TIME - TIME V82GOFF1 BEGAN
                STORE   TSTART82        #                SAVE IT
                DLOAD   BZE             # SR30.1 SETS -TPER=0 IF HPER L/
                        -TPER           #  HPERMIN (300 OR 35) KFT.
                        TICKTIFF        # (-TPER = 0)
TICKTPER        DLOAD   DAD             # (-TPER NON ZERO)  TFF WAS NOT COMPUTED.
                        -TPER           # BUT WAS SET TO 59M59S.DONT TICK TFF, DO
                        TSTART82        # TICK -TPER. DISPLAY BOTH.
                STORE   -TPER           # -TPER CORRECTED FOR TIME SINCE V82GOFF1
                EXIT                    # BEGAN.
                CAF     BIT1
                TS      V82FLAGS        # INFORMS TICKTEST TO INCREMENT ONLY -TPER
                TC      ENDOFJOB

TICKTIFF        DLOAD   DAD             # (-TPER=0)  TFF WAS COMPUTED.TICK TFF.
                        TFF             # DO NOT TICK -TPER.DISPLAY TFF, BUT NOT
                        TSTART82        # -TPER.
                STORE   TFF             # TFF CORRECTED FOR TIME SINCE V82GOFF1
                EXIT                    # BEGAN.
                CAF     BIT2
                TS      V82FLAGS        # INFORMS TICKTEST TO INCREMENT ONLY TFF.
                TC      ENDOFJOB
```

```
TICKTEST        CAF     BIT5            # THIS WAITLIST PROGRAM PERPETUATES ITSELF
                MASK    EXTVBACT        # ONCE A SEC UNTIL BIT 5 OF EXTVBACT =0.
                CCS     A
                TC      DOTICK
                CAF     PRIO25
                TC      NOVAC           # TERMINATE V 82.CANT CALL ENDEXT IN RUPT.
                EBANK=  EXTVBACT
                2CADR   ENDEXT

                TC      TASKOVER
DOTICK          CAF     1SEC            # RE-REQUEST TICKTEST.
                TC      WAITLIST
                EBANK=  TFF
                2CADR   TICKTEST

                CAF     THREE
                MASK    V82FLAGS
                INDEX   A
                TC      +1
                TC      TASKOVER        # IF NO FLAGBITS SET DONT CHANGE TFF OR
                                        # -TPER, BUT CONTINUE LOOP.
                TC      TPERTICK        # ONLY BIT 1 SET. INCR -TPER BY 1 SEC.
TFFTICK         CAF     1SEC            # ONLY BIT 2 SET. INCR TFF BY 1 SEC.
                TS      L
                CAF     ZERO
                DAS     TFF
                TC      TASKOVER
TPERTICK        CAF     1SEC
                TS      L
                CAF     ZERO
                DAS     -TPER
                TC      TASKOVER
```

```
V82GON          EXIT                            #  AVERAGE G ON. USE CURRENT STATE VECTOR
                                                # FOR ORBITAL PARAMETER CALCULATIONS.
                CAF     PRIO7                   # LESS THAN LAMBERT
                TC      FINDVAC                 # V82GON1 WILL PERFORM ORBIT CALCULATIONS
                EBANK=  TFF                     # ABOUT PROPER BODY APPROX ONCE PER SEC.
                2CADR   V82GON1

                RELINT
                CCS     NEWJOB                  # WITHOLD V16 N44 UNTIL FIRST ORBIT CALC
                TC      CHANG1                  # IS DONE. NOTE: V82GON1 (PRIO7, FINDVAC
                                                # JOB) IS COMPLETED BEFORE V82GON (PRIO7,
                                                # NOVAC JOB).
V82REDSP        CAF     V16N44                  # MONITOR HAPO, HPER, TFF
                TC      BANKCALL
                CADR    GOXDSPF
                TC      B5OFF                   # TERM THIS TELLS V82GON1 TO KILL ITSELF.
                TC      B5OFF                   # PROC DITTO.
                TC      V82REDSP                # RECYCLE

V82GON1         TC      INTPRET                 # THIS EXEC PROGRAM PERPETUATES ITSELF
                                                # ONCE A SEC UNTIL BIT 5 OF EXTVBACT =0.
                VLOAD   GOTO                    # HOLDS OFF CCS NEWJOB BETWEEN RN AND
                        RN                      #   VN FETCH SO RN , VN ARE FROM SAME
                        NEXTLINE                #   STATE VECTOR UPDATE.
NEXTLINE        STOVL   RONE                    #  RN AT (-29)M FOR EARTH OR MOON
                        VN
                STORE   VONE                    # VN AT (-7)M/CS FOR EARTH OR MOON
                BON     GOTO
                        MOONTHIS                # FLAG INDICATES BODY ABOUT WHICH ORBITAL
                        MOONGON                 #    CALCULATIONS ARE TO BE PERFORMED.
                        EARTHGON                #   IF SET - MOON , IF RESET - EARTH.

MOONGON         SET     DLOAD
                        V82EMFLG                # INDICATE MOON SCALING FOR SR30.1
                        1/RTMUM                 # LUNAR PARAMETERS LOADED HERE FOR SR30.1
                STODL   TFF/RTMU
                        MINPERM
                STOVL   HPERMIN
                        RLS                     #  SCALED AT (-27)M.
                ABVAL   GOTO
                        V82GON2
EARTHGON        CLEAR   DLOAD
                        V82EMFLG                # INDICATE EARTH SCALING FOR SR30.1
                        1/RTMUE                 # EARTH PARAMETERS LOADED HERE FOR SR30.1
                STODL   TFF/RTMU
                        MINPERE
                STODL   HPERMIN
                        RPAD
V82GON2         STCALL  RPADTEM                 # COMMON CODE FOR EARTH & MOON.
                        SR30.1
```

```
                EXIT
V82GON3         CAF     BIT5
                MASK    EXTVBACT        # SEE IF ASTRONAUT HAS SIGNALLED TERMINATE
                EXTEND
                BZF     ENDEXT          # YES, TERMINATE VB 82 LOOP
                CAF     1SEC
                TC      BANKCALL        # WAIT ONE SECOND BEFORE REPEATING
                CADR    DELAYJOB        #   ORBITAL PARAMETER COMPUTATION.
                TC      V82GON1

SPLRET          =       V82GON3
```

```
# SUBROUTINE NAME: SR30.1
# MOD NO: 0                                                                      DATE: 16 FEB 67
# MOD BY: R. R. BAIRNSFATHER                                                     LOG SECTION: R32
# MOD NO: 1      MOD BY: R. R. BAIRNSFATHER      DATE: 11 APR 67         SR30.1 CHANGED TO ALLOW MONITOR OPERN
# MOD NO: 2      MOD BY: R. R. BAIRNSFATHER      DATE: 14 APR 67         ADD OVFL CK FOR RAPO
# MOD NO: 3      MOD BY ALONSO                   DATE: 11 DEC 67         SUBROUTINE REWRITTEN
# MOD NO: 4      MOD BY ALONSO                   DATE: 26 MAR 68         PROG MOD TO HANDLE DIF EARTH/MOON SCALE
# MOD NO: 5      MOD BY: R. R. BAIRNSFATHER      DATE: 6 AUG 68          OVFL CK FOR HAPO & HPER. VOIDS MOD #2.
#
# NEW FUNCTIONAL DESCRIPTION:  ORBITAL PARAMETERS DISPLAY FOR NOUNS 32 AND 44.
# SR30.1 CALLS TFFCONMU AND TFFRP/RA TO CALCULATE RPER (PERIGEE RADIUS),
# RAPO (APOGEE RADIUS), HPER (PERIGEE HEIGHT ABOVE LAUNCH PAD OR LUNAR
# LANDING SITE), HAPO (APOGEE HEIGHT AS ABOVE), TPER (TIME TO PERIGEE),
# TFF (TIME TO INTERSECT 300 KFT ABOVE PAD OR 35KFT ABOVE LANDING SITE).
# IF HPER IS GREATER THAN OR EQUAL TO HPERMIN, CALCULATES TPER AND STORES
# NEGATIVE IN -TPER.  OTHERWISE STORES +0 IN -TPER.  WHENEVER TPER IS
# CALCULATED, TFF IS NOT COMPUTABLE AND DEFAULTS TO -59MIN 59SEC.  IF HAPO
# WOULD EXCEED 9999.9 NM, IT IS LIMITED TO THAT VALUE FOR DISPLAY.
#
# ADDENDUM:      HAPO AND HPER SHOULD BE CHANGED TO READ HAPOX AND HPERX IN THE
#                ABOVE REMARKS.
#
# CALLING SEQUENCE:     CALL
#                               SR30.1
#
# SUBROUTINES CALLED:   TFFCONMU, TFFRP/RA, CALCTPER, CALCTFF
#
# NORMAL EXIT MODE:     CALLING LINE +1 (STILL IN INTERPRETIVE MODE)
#
# ALARMS:        NONE
#
# OUTPUT:        RAPO    (-29) M EARTH   APOGEE RADIUS   EARTH CENTERED COORD.
#                        (-27) M MOON                    MOON CENTERED COORD.
#                RPER    (-29) M EARTH   PERIGEE RADIUS  EARTH CENTERED COORD.
#                        (-27) M MOON                    MOON CENTERED COORD.
#                HAPOX   (-29) M         APOGEE ALTITUDE ABOVE PAD OR LAND. SITE MAX VALUE LIMITED TO 9999.9 NM.
#                HPERX   (-29) M         PERIGEE ALT. ABOVE PAD OR LAND. SITE    MAX VALUE LIMITED TO 9999.9 NM.
#                TFF     (-28) CS        TIME TO 300KFT OR 35KFT ALTITUDE
#                -TPER   (-28) CS        TIME TO PERIGEE
#
# ERASABLE INITIALIZATION REQUIRED --
#       TFF/RTMU        (+17) EARTH     RECIPROCAL OF PROPER GRAV CONSTANT FOR
#                       (+14) MOON      EARTH OR MOON = 1/SQRT(MU).
#       RONE            (-29) M         STATE VECTOR
#       VONE            (-7)  M/CS      STATE VECTOR
#       RPADTEM         (-29) M EARTH   RADIUS OF LAUNCH PAD OR LUNAR LANDING
#                       (-27) M MOON    SITE.
#       HPERMIN         (-29) M EARTH   (300 OR 35) KFT MINIMUM PERIGEE ALTITUDE
#                       (-27) M MOON    ABOVE LAUNCH PAD OR LUNAR LANDING SITE.
#       V82EMFLG        (INT SW BIT)    RESET FOR EARTH, SET FOR MOON.
#
# DEBRIS:        QPREG, PDL, S2
```

```
                COUNT*  $$/SR30S

SR30.1          SETPD   STQ             # INITIALIZE PUSHDOWN LIST.
                        0
                        S2
                                        # SR30.1 INPUT: RONE AT (-29)M EARTH/MOON
                                        #               VONE AT (-7)M/CS
                                        # TFFCONMU, TFFRP/RA, CALCTPER, AND CALCTFF
                                        # CALLS REQUIRE:
                                        # EARTH CENTERED (NO RESCALING REQUIRED)
                                        #       RONE SCALED TO B-29 M
                                        #       VONE SCALED TO B-7  M/CS
                                        # MOON CENTERED (RESCALING REQUIRED)
                                        #       RONE SCALED TO B-27 M
                                        #       VONE SCALED TO B-5  M/CS
                BOFF    VLOAD
                        V82EMFLG        # OFF FOR EARTH, ON FOR MOON.
                        TFFCALLS
                        RONE
                VSL2
                STOVL   RONE
                        VONE
                VSL2
                STORE   VONE
TFFCALLS        CALL
                        TFFCONMU
                CALL                    # TFFRP/RA COMPUTES RAPO,RPER.
                        TFFRP/RA
                                        # RETURNS WITH RAPO IN D(MPAC).
                DSU
                        RPADTEM
                BOFF    SR2R            # NEED HAPO AT (-29)M FOR DISPLAY.
                                        # IF MOON CENTERED, RESCALE FROM (-27)M.
                                        # IF EARTH CENTERED ALREADY AT (-29)M.
                        V82EMFLG        # OFF FOR EARTH, ON FOR MOON.
                        +1
                CALL                    # IF RAPO > MAXNM, SET RAPO =9999.9 NM.
                        MAXCHK          # OTHERWISE STORE (RAPO-RPADTEM) IN HAPO.
STORHAPO        STODL   HAPOX
                        RPER
                DSU
                        RPADTEM         # GIVES HPER AT (-29)M EARTH, (-27)M MOON.
                STORE   MPAC +4         # SAVE THIS FOR COMPARISON TO HPERMIN.
                BOFF    SR2R            # NEED HPER AT (-29)M FOR DISPLAY.
                                        # IF MOON CENTERED, RESCALE FROM (-27)M.
                                        # IF EARTH CENTERED ALREADY AT (-29)M.
                        V82EMFLG        # OFF FOR EARTH, ON FOR MOON.
                        +1
                CALL                    # IF HPER > MAXNM, SET HPER = 9999.9 NM.
                        MAXCHK
```

```
STORHPER        STODL   HPERX           # STORE (RPER - RPADTEM) INTO HPERX.
                        MPAC    +4
                DSU     BPL             # HPERMIN AT (-29)M FOR EARTH, (-27)M MOON
                        HPERMIN         # IF HPER L/   HPERMIN (300 OR 35)KFT,
                        DOTPER          #  THEN ZERO INTO -TPER.
                DLOAD   GOTO            #   OTHERWISE CALCULATE TPER.
                        HI6ZEROS
                        SKIPTPER
DOTPER          DLOAD   CALL
                        RPER
                        CALCTPER
                DCOMP                   # TPER IS PUT NEG INTO -TPER.
SKIPTPER        STODL   -TPER
                        HPERMIN         # HPERMIN AT (-29)M FOR EARTH, (-27)M MOON
                DAD     CALL
                        RPADTEM         # RPADTEM AT (-29)M FOR EARTH, (-27)M MOON
                        CALCTFF         # GIVES 59M59S FOR TFF IF HPER G/
                DCOMP                   #   HPERMIN + RPADTEM. (TPER WAS NON ZERO)
                STCALL  TFF             # OTHERWISE COMPUTES TFF.       (GOTO)
                        S2

MAXCHK          DSU     BPL             # IF C(MPAC) > 9999.9 NM. MPAC = 9999.9 NM.
                        MAXNM
                        +3              # OTHERWISE C(MPAC) = B(MPAC).
                DAD     RVQ
                        MAXNM
 +3             DLOAD   RVQ             # (USED BY P30 - P37 ALSO)
                        MAXNM

MAXNM           2OCT    0106505603
```

```
# STABLE ORBIT RENDEZVOUS PROGRAMS (P38 AND P78)
#
# MOD NO -1                LOG SECTION - STABLE ORBIT - P38-P39
# MOD BY RUDNICKI.S        DATE 25JAN68
#
# FUNCTIONAL DESCRIPTION
#
#       P38 AND P78 CALCULATE THE REQUIRED DELTA V AND OTHER INITIAL
#       CONDITIONS REQUIRED BY THE AGC TO (1) PUT THE ACTIVE VEHICLE
#       ON A TRANSFER TRAJECTORY THAT INTERCEPTS THE PASSIVE VEHICLE
#       ORBIT A GIVEN DISTANCE, DELTA R, EITHER AHEAD OF OR BEHIND THE
#       PASSIVE VEHICLE AND (2) ACTUALLY PLACE THE ACTIVE VEHICLE IN THE
#       PASSIVE VEHICLE ORBIT WITH A DELTA R SEPARATION BETWEEN THE TWO
#       VEHICLES
#
# CALLING SEQUENCE
#
#       ASTRONAUT REQUEST THRU DSKY
#
#       V37E38E         IF THIS VEHICLE IS ACTIVE VEHICLE
#       V37E78E         IF OTHER VEHICLE IS ACTIVE VEHICLE
#
# INPUT
#
#       (1)     SOI MANEUVER
#
#               (A)  TIG        TIME OF SOI MANEUVER
#               (B)  CENTANG    ORBITAL CENTRAL ANGLE OF THE PASSIVE VEHICLE
#                               DURING THE TRANSFER FROM TIG TO TIME OF INTERCEPT
#               (C)  DELTAR     THE DESIRED SEPARATION OF THE TWO VEHICLES
#                               SPECIFIED AS A DISTANCE ALONG THE PASSIVE VEHICLE
#                               ORBIT
#               (D)  OPTION     EQUALS 1 FOR SOI
#
#       (2)     SOR MANEUVER
#
#               (A)  TIG        TIME OF SOR MANEUVER
#               (B)  CENTANG    AN OPTIONAL RESPECIFICATION OF 1 (B) ABOVE
#               (C)  OPTION     EQUALS 2 FOR SOR
#               (D)  DELTTIME   THE TIME REQUIRED TO TRAVERSE DELTA R WHEN
#                               TRAVELING AT A VELOCITY EQUAL TO THE HORIZONTAL
#                               VELOCITY OF THE PASSIVE VEHICLE - SAVED FROM
#                               SOI PHASE
#               (E)  TINT       TIME OF INTERCEPT (SOI) - SAVED FROM SOI PHASE
#
# OUTPUT
#
#       (1)  TRKMKCNT   NUMBER OF MARKS
#       (2)  TTOGO      TIME TO GO
#       (3)  +MGA       MIDDLE GIMBAL ANGLE
```

```
#       (4)  DSPTEM1    TIME OF INTERCEPT OF PASSIVE VEHICLE ORBIT
#                       (FOR SOI ONLY)
#       (5)  POSTTPI    PERIGEE ALTITUDE OF ACTIVE VEHICLE ORBIT AFTER
#                       THE SOI (SOR) MANEUVER
#       (6)  DELVTPI    MAGNITUDE OF DELTA V AT SOI (SOR) TIME
#       (7)  DELVTPF    MAGNITUDE OF DELTA V AT INTERCEPT TIME
#       (8)  DELVLVC    DELTA VELOCITY AT SOI (AND SOR) - LOCAL VERTICAL
#                       COORDINATES
#
# SUBROUTINE USED
#
#       AVFLAGA
#       AVFLAGP
#       VNDSPLY
#       BANKCALL
#       GOFLASHR
#       GOTOPOOH
#       BLANKET
#       ENDOFJOB
#       PREC/TT
#       SELECTMU
#       INTRPVP
#       MAINRTNE

                BANK    04
                SETLOC  STBLEORB
                BANK

                EBANK=  SUBEXIT
                COUNT*  $$/P3879

P38             TC      BANKCALL
                CADR    AVFLAGA         # THIS VEHICLE ACTIVE
                TC      +3
P78             TC      BANKCALL
                CADR    AVFLAGP         # OTHER VEHICLE ACTIVE
                TC      BANKCALL
                CADR    P20FLGON        # SET UPDATFLG, TRACKFLG
                CAF     DECTWO
                TS      NN
                CAF     V06N33SR        # DISPLAY TIG
                TC      VNDSPLY
                CAF     V06N55SR        # DISPLAY CENTANG
                TCR     BANKCALL
                CADR    GOFLASHR
                TCF     GOTOPOOH        # TERMINATE
                TCF     +5              # PROCEED
                TCF     -5              # RECYCLE
                CAF     THREE           # IMMEDIATE RETURN - BLANK R1, R2
                TCR     BLANKET
```

```
                TCF     ENDOFJOB
                CAF     FIVE
                TS      OPTION1
                CAF     ONE
                TS      OPTION2         # OPTION CODE IS SET TO 1
                CAF     V04N06SR        # DISPLAY OPTION CODE - 1 = SOI, 2 = SOR
                TCR     BANKCALL
                CADR    GOFLASHR
                TCF     GOTOPOOH        # TERMINATE
                TCF     +5              # PROCEED
                TCF     -5              # RECYCLE
                CAF     BIT3            # IMMEDIATE RETURN - BLANK R3
                TCR     BLANKET
                TCF     ENDOFJOB
                TC      INTPRET
                SLOAD   SR1
                        OPTION2
                BHIZ    DLOAD
                        OPTN1
                        TINT
                STORE   TINTSOI         # STORE FOR SOR PHASE
                CLRGO
                        OPTNSW          # OPTNSW: ON = SOI, OFF = SOR
                        JUNCTN1
OPTN1           SET     CLEAR           # SOI
                        OPTNSW
                        UPDATFLG
                CALL
                        PREC/TT
                DAD     SET
                        TIG
                        UPDATFLG
                STORE   TINT            # TI = TIG + TF
                EXIT
                CAF     V06N57SR        # DISPLAY DELTA R
                TCR     BANKCALL
                CADR    GOFLASHR
                TCF     GOTOPOOH        # TERMINATE
                TCF     +5              # PROCEED
                TCF     -5              # RECYCLE
                CAF     SIX             # IMMEDIATE RETURN - BLANK R2, R3
                TCR     BLANKET
                TCF     ENDOFJOB
 +5             EXTEND
                DCA     TINT
                DXCH    DSPTEM1         # FOR DISPLAY
                CAF     V06N34SR        # DISPLAY TIME OF INTERCEPT
                TC      VNDSPLY
                TC      INTPRET
JUNCTN1         CLEAR   CALL
```

```
                                P39/79SW
                                SELECTMU        # SELECT MU, CLEAR FINALFLG, GO TO VN1645
RECYCLE         CALL
                                PREC/TT
                BOFF            DLOAD
                                OPTNSW
                                OPTN2
                                TINT
                STCALL          TDEC1           # PRECISION UPDATE PASSIVE VEHICLE TO
                                INTRPVP         #       INTERCEPT TIME
                VLOAD           UNIT
                                RATT            # RP/(RP)
                PDVL            VXV
                                VATT
                ABVAL           NORM            # (VP X RP/(RP))
                                X1
                PDDL            DDV
                                DELTAR
                SL*                             # DELTA R / (VP X RP/RP)
                                0       -7,1
                STCALL          DELTTIME        # DELTA T = (RP) DELTA R / (VP X RP)
                                JUNCTN2
OPTN2           DLOAD           DAD
                                TINTSOI
                                T
                STORE           TINT            # TI = TI + TF
JUNCTN2         DLOAD           DSU
                                TINT
                                DELTTIME
                STORE           TARGTIME        # TT = TI - DELTA T

# .... MAINRTNE ....
# SUBROUTINES USED:
#
#       S3435.25
#       PERIAPO1
#       SHIFTR1
#       VNDSPLY
#       BANKCALL
#       GOFLASH
#       GOTOPOOH
#       VN1645

MAINRTNE        STCALL          TDEC1           # PRECISION UPDATE PASSIVE VEHICLE TO
                                INTRPVP         #       TARGET TIME
                DLOAD
                                TIG
                STORE           INTIME
                SSP             VLOAD
                                SUBEXIT
                                TEST3979
```

```
                                RATT
                        CALL
                                S3435.25
TEST3979                BOFF    BON
                                P39/79SW
                                MAINRTN1
                                FINALFLG
                                P39P79
                        SET
                                UPDATFLG
P39P79                  EXIT
                        TC      DSPLY81         # FOR P39 AND P79
MAINRTN1                VLOAD   ABVAL
                                DELVEET3
                        STOVL   DELVTPI         # DELTA V
                                VPASS4
                        VSU     ABVAL
                                VTPRIME
                        STOVL   DELVTPF         # DELTA V (FINAL) = V'T - VT
                                RACT3
                        PDVL    CALL
                                VIPRIME
                                PERIAPO1        # GET PERIGEE ALTITUDE
                        CALL
                                SHIFTR1
                        STORE   POSTTPI
                        BON     SET
                                FINALFLG
                                DSPLY58
                                UPDATFLG
DSPLY58                 EXIT
                        CAF     V06N58SR        # DISPLAY HP, DELTA V, DELTA V (FINAL)
                        TC      VNDSPLY
DSPLY81                 CAF     V06N81SR        # DISPLAY DELTA V (LV)
                        TC      VNDSPLY
                        TC      INTPRET
                        CLEAR   VLOAD
                                XDELVFLG
                                DELVEET3
                        STCALL  DELVSIN
                                VN1645          # DISPLAY TRKMKCNT, TTOGO, +MGA
                        BON     GOTO
                                P39/79SW
                                P39/P79B
                                RECYCLE

# STABLE ORBIT MIDCOURSE PROGRAM (P39 AND P79)
#
# MOD NO -1                     LOG SECTION - STABLE ORBIT - P38-P39
# MOD BY RUDNICKI.S             DATE 25JAN68
#
```

```
# FUNCTIONAL DESCRIPTION
#
#       P39 AND P79 CALCULATE THE REQUIRED DELTA V AND OTHER INITIAL
#       CONDITIONS REQUIRED BY THE AGC TO MAKE A MIDCOURSE CORRECTION
#       MANEUVER AFTER COMPLETING THE SOI MANEUVER BUT BEFORE MAKING
#       THE SOR MANEUVER.
#
# CALLING SEQUENCE
#
#       ASTRONAUT REQUEST THRU DSKY
#
#       V37E39E         IF THIS VEHICLE IS ACTIVE VEHICLE
#       V37E79E         IF OTHER VEHICLE IS ACTIVE VEHICLE
#
# INPUT
#
#       (1)  TPASS4     TIME OF INTERCEPT - SAVED FROM P38/P78
#       (2)  TARGTIME   TIME THAT PASSIVE VEHICLE IS AT INTERCEPT POINT -
#                       SAVED FROM P38/P78
#
# OUTPUT
#
#       (1)  TRKMKCNT   NUMBER OF MARKS.
#       (2)  TTOGO      TIME TO GO
#       (3)  +MGA       MIDDLE GIMBAL ANGLE
#       (4)  DELVLVC    DELTA VELOCITY AT MID - LOCAL VERTICAL COORDINATES
#
# SUBROUTINES USED
#
#       AVFLAGA
#       AVFLAGP
#       LOADTIME
#       SELECTMU
#       PRECSET
#       S34/35.1
#       MAINRTNE

P39             TC      BANKCALL
                CADR    AVFLAGA         # THIS VEHICLE ACTIVE
                EXTEND
                DCA     ATIGINC
                TC      P39/P79A
P79             TC      BANKCALL
                CADR    AVFLAGP         # OTHER VEHICLE ACTIVE
                EXTEND
                DCA     PTIGINC
P39/P79A        DXCH    KT              # TIME TO PREPARE FOR BURN
                TC      BANKCALL
                CADR    P20FLGON        # SET UPDATFLG, TRACKFLG
                TC      INTPRET
```

```
                SET     CALL
                        P39/79SW
                        SELECTMU        # SELECT MU, CLEAR FINALFLG, GO TO VN1645
P39/P79B        RTB     DAD
                        LOADTIME
                        KT
                STORE   TIG             # TIG = T (PRESENT) + PREPARATION TIME
                STCALL  TDEC1           # PRECISION UPDATE ACTIVE AND PASSIVE
                        PRECSET         #       VEHICLES TO TIG
                CALL
                        S34/35.1        # GET UNIT NORMAL
                DLOAD   GOTO
                        TARGTIME
                        MAINRTNE        # CALCULATE DELTA V AND DELTA V (LV)

# .... PREC/TT ....
# SUBROUTINES USED
#
#       PRECSET
#       TIMETHET
#       S34/35.1

PREC/TT         STQ     DLOAD
                        RTRN
                        TIG
                STCALL  TDEC1           # PRECISION UPDATE ACTIVE AND PASSIVE
                        PRECSET         #       VEHICLES TO TIG
                VLOAD   VSR*
                        RPASS3
                        0,2
                STODL   RVEC
                        CENTANG
                PUSH    COS
                STODL   CSTH
                SIN     SET
                        RVSW
                STOVL   SNTH
                        VPASS3
                VSR*
                        0,2
                STCALL  VVEC            # GET TRANSFER TIME BASED ON CENTANG OF
                        TIMETHET        #       PASSIVE VEHICLE
                CALL
                        S34/35.1        # GET UNIT NORMAL
                DLOAD   GOTO
                        T
                        RTRN

# .... INTRPVP ....
# SUBROUTINES USED
#
#       CSMPREC
```

```
#       LEMPREC

INTRPVP         STQ     BOFF            # PRECISION UPDATE PASSIVE VEHICLE TO
                        RTRN            #       TDEC1
                        AVFLAG
                        OTHERV
                CALL
                        CSMPREC
                GOTO
                        RTRN
OTHERV          CALL
                        LEMPREC
                GOTO
                        RTRN

# .... VNDSPLY ....
# SUBROUTINES USED
#
#       BANKCALL
#       GOFLASH
#       GOTOPOOH

VNDSPLY         EXTEND                  # FLASH DISPLAY
                QXCH    RTRN
                TS      VERBNOUN
                CA      VERBNOUN
                TCR     BANKCALL
                CADR    GOFLASH
                TCF     GOTOPOOH        # TERMINATE
                TC      RTRN            # PROCEED
                TCF     -5              # RECYCLE
V06N33SR        VN      0633
V06N55SR        VN      0655
V04N06SR        VN      0406
V06N57SR        VN      0657
V06N34SR        VN      0634
V06N58SR        VN      0658
V06N81SR        VN      0681
DECTWO          OCT     2

# *** END OF KISSING  .050 ***
```

```
##
# BURN, BABY, BURN -- MASTER IGNITION ROUTINE

                BANK    36
                SETLOC  P40S
                BANK
                EBANK=  WHICH
                COUNT*  $$/P40

# THE MASTER IGNITION ROUTINE IS DESIGNED FOR USE BY THE FOLLOWING LEM PROGRAMS:  P12, P40, P42, P61, P63.
# IT PERFORMS ALL FUNCTIONS IMMEDIATELY ASSOCIATED WITH APS OR DPS IGNITION:  IN PARTICULAR, EVERYTHING LYING
# BETWEEN THE PRE-IGNITION TIME CHECK -- ARE WE WITHIN 45 SECONDS OF TIG? -- AND TIG + 26 SECONDS, WHEN DPS
# PROGRAMS THROTTLE UP.
#
# VARIATIONS AMONG PROGRAMS ARE ACCOMODATED BY MEANS OF TABLES CONTAINING CONSTANTS (FOR AVEGEXIT, FOR
# WAITLIST, FOR PINBALL) AND TCF INSTRUCTIONS.  USERS PLACE THE ADRES OF THE HEAD OF THE APPROPRIATE TABLE
# (OF P61TABLE FOR P61LM, FOR EXAMPLE) IN ERASABLE REGISTER `WHICH' (E4).  THE IGNITION ROUTINE THEN INDEXES BY
# WHICH TO OBTAIN OR EXECUTE THE PROPER TABLE ENTRY.  THE IGNITION ROUTINE IS INITIATED BY A TCF BURNBABY,
# THROUGH BANKJUMP IF NECESSARY.  THERE IS NO RETURN.
#
# THE MASTER IGNITION ROUTINE WAS CONCEIVED AND EXECUTED, AND (NOTA BENE) IS MAINTAINED BY ADLER AND EYLES.
#
#                  HONI SOIT QUI MAL Y PENSE
#
#       ***********************************************
#               TABLES FOR THE IGNITION ROUTINE
#       ***********************************************
#
#                       NOLI SE TANGERE

P12TABLE        VN      0674            # (0)
                TCF     ULLGNOT         # (1)
                TCF     COMFAIL3        # (2)
                TCF     GOCUTOFF        # (3)
                TCF     TASKOVER        # (4)
                TCF     P12SPOT         # (5)
                DEC     0               # (6)   NO ULLAGE
                EBANK=  WHICH
                2CADR   SERVEXIT        # (7)

                TCF     DISPCHNG        # (11)
                TCF     WAITABIT        # (12)
                TCF     P12IGN          # (13)

P40TABLE        VN      0640            # (0)
                TCF     ULLGNOT         # (1)
                TCF     COMFAIL4        # (2)
                TCF     GOPOST          # (3)
                TCF     TASKOVER        # (4)
                TCF     P40SPOT         # (5)
```

```
                DEC     2240            # (6)
                EBANK=  OMEGAQ
                2CADR   STEERING        # (7)

                TCF     P40SJUNK        # (11)
                TCF     WAITABIT        # (12)
                TCF     P40IGN          # (13)
                TCF     REP40ALM        # (14)

P41TABLE        TCF     P41SPOT         # (5)
                DEC     -1              # (6)
                EBANK=  OMEGAQ
                2CADR   CALCN85         # (7)

                TCF     COMMON          # (11)
                TCF     TIGTASK         # (12)

P42TABLE        VN      0640            # (0)
                TCF     WANTAPS         # (1)
                TCF     COMFAIL4        # (2)
                TCF     GOPOST          # (3)
                TCF     TASKOVER        # (4)
                TCF     P42SPOT         # (5)
                DEC     2640            # (6)
                EBANK=  OMEGAQ
                2CADR   STEERING        # (7)

                TCF     P40SJUNK        # (11)
                TCF     WAITABIT        # (12)
                TCF     P42IGN          # (13)
                TCF     P42STAGE        # (14)

P63TABLE        VN      0662            # (0)
                TCF     ULLGNOT         # (1)
                TCF     COMFAIL3        # (2)
                TCF     V99RECYC        # (3)
                TCF     TASKOVER        # (4)
                TCF     P63SPOT         # (5)
                DEC     2240            # (6)
                EBANK=  WHICH
                2CADR   SERVEXIT        # (7)

                TCF     DISPCHNG        # (11)
                TCF     WAITABIT        # (12)
```

```
                TCF     P63IGN          # (13)

ABRTABLE        VN      0663            # (0)
                TCF     ULLGNOT         # (1)
                TCF     COMFAIL3        # (2)
                TCF     GOCUTOFF        # (3)
                TCF     TASKOVER        # (4)
                NOOP                    # (5)
                NOOP                    # (6)
                NOOP                    # (7)
                NOOP
                TCF     DISPCHNG        # (11)
                TCF     WAITABIT        # (12)
                TCF     ABRTIGN         # (13)

#       *********************************
#       GENERAL PURPOSE IGNITION ROUTINES
#       *********************************

BURNBABY        TC      PHASCHNG        # GROUP 4 RESTARTS HERE
                OCT     04024

                CAF     ZERO            # EXTIRPATE JUNK LEFT IN DVTOTAL
                TS      DVTOTAL
                TS      DVTOTAL +1

                TC      BANKCALL        # P40AUTO MUST BE BANKCALLED EVEN FROM ITS
                CADR    P40AUTO         # OWN BANK TO SET UP RETURN PROPERLY

B*RNB*B*        EXTEND
                DCA     TIG             # STORE NOMINAL TIG FOR OBLATENESS COMP.
                DXCH    GOBLTIME        # AND FOR P70 OR P71.

                INHINT
                TC      IBNKCALL
                CADR    ENGINOF3
                RELINT

                INDEX   WHICH
                TCF     5

P42SPOT         =       P40SPOT         # (5)
P12SPOT         =       P40SPOT         # (5)
P63SPOT         =       P41SPOT         # (5)   IN P63 CLOKTASK ALREADY GOING
P40SPOT         CS      CNTDNDEX        # (5)
```

```
                TC      BANKCALL        # MUST BE BANKCALLED FOR GENERALIZED
                CADR    STCLOK2         #       RETURN
P41SPOT         TC      INTPRET         # (5)
                DLOAD   DSU
                        TIG
                        D29.9SEC
                STCALL  TDEC1
                        INITCDUW
                BOFF    CALL
                        MUNFLAG
                        GOMIDAV
                        CSMPREC
                VLOAD   MXV
                        VATT1
                        REFSMMAT
                VSR1
                STOVL   V(CSM)          # CSM VELOCITY -- M/CS*2(7)
                        RATT1
                VSL4    MXV
                        REFSMMAT
                STCALL  R(CSM)          # CSM POSITION -- M*2(24)
                        MUNGRAV
                STODL   G(CSM)          # CSM GRAVITY VEC. -- M/CS*2(7)
                        TAT
                STORE   TDEC1           # RELOAD TDEC1 FOR MIDTOAV.
GOMIDAV         CALRB
                        MIDTOAV1
                TCF     CALLT-35        # MADE IT IN TIME.

                EXTEND                  # TIG WAS SLIPPED, SO RESET TIG TO 29.9
                DCA     PIPTIME1        # SECONDS AFTER THE TIME TO WHICH WE DID
                DXCH    TIG             # INTEGRATE.
                EXTEND
                DCA     D29.9SEC
                DAS     TIG

CALLT-35        DXCH    MPAC
                DXCH    SAVET-30        # DELTA-T UNTIL TIG-30
                EXTEND
                DCS     5SECDP
                DAS     SAVET-30        # DELTA-T UNTIL TIG-35
                EXTEND
                DCA     SAVET-30
                TC      LONGCALL
                EBANK=  TTOGO
                2CADR   TIG-35

                TC      PHASCHNG
                OCT     20254           # 4.25SPOT FOR TIG-35 RESTART.
```

```
                TC      CHECKMM
                DEC     63
                TCF     ENDOFJOB        # NOT P63
                CS      CNTDNDEX        # P63 CAN START DISPLAYING NOW.
                TS      DISPDEX
                TC      INTPRET
                VLOAD   ABVAL
                        VN1
                STORE   ABVEL           # INITIALIZE ABVEL FOR P63 DISPLAY
                EXIT
                TCF     ENDOFJOB

#       ********************************

TIG-35          CAF     5SEC
                TC      TWIDDLE
                ADRES   TIG-30

                TC      PHASCHNG
                OCT     40154           # 4.15SPOT FOR TIG-30 RESTART

                CS      BLANKDEX        # BLANK DSKY FOR 5 SECONDS
                TS      DISPDEX

                INDEX   WHICH
                CS      6               # CHECK ULLAGE TIME.
                EXTEND
                BZMF    TASKOVER
                CAF     4.9SEC          # SET UP TASK TO RESTORE DISPLAY AT TIG-30
                TC      TWIDDLE
                ADRES   TIG-30.1

                CAF     PRIO17          # A NEGATIVE ULLAGE TIME INDICATES P41, IN
                TC      NOVAC           # WHICH CASE WE HAVE TO SET UP A JOB TO
                EBANK=  TTOGO           # BLANK THE DSKY FOR FIVE SECONDS, SINCE
                2CADR   P41BLANK        # CLOKJOB IS NOT RUNNING DURING P41.

                TCF     TASKOVER

P41BLANK        TC      BANKCALL        # BLANK DSKY.
                CADR    CLEANDSP
                TCF     ENDOFJOB

TIG-30.1        CAF     PRIO17          # SET UP JOB TO RESTORE DISPLAY AT TIG-30
                TC      NOVAC
                EBANK=  TTOGO
                2CADR   TIG-30A

                TCF     TASKOVER
```

```
TIG-30A         CAF     V16N85B
                TC      BANKCALL        # RESTORE DISPLAY.
                CADR    REGODSP         # REGODSP DOES A TCF ENDOFJOB

#       *********************************

TIG-30          CAF     S24.9SEC
                TC      TWIDDLE
                ADRES   TIG-5

                CS      CNTDNDEX        # START UP CLOKTASK AGAIN
                TS      DISPDEX

                INDEX   WHICH           # PICK UP APPROPRIATE ULLAGE -- ON TIME
                CA      6               # WAS CAF --- RSB 2009.
                EXTEND
                BZMF    ULLGNOT         # DON'T SET UP ULLAGE IF DT IS NEG OR ZERO
                TS      SAVET-30        # SAVE DELTA-T FOR RESTART
                TC      TWIDDLE
                ADRES   ULLGTASK

                CA      THREE           # RESTART PROTECT ULLGTASK (1.3SPOT)
                TS      L
                CS      THREE
                DXCH    -PHASE1
                CS      TIME1
                TS      TBASE1

                INDEX   WHICH
                TCF     1

WANTAPS         CS      FLGWRD10        # (1) FOR P42 ENSURE APSFLAG IS SET.  IF IT
                MASK    APSFLBIT        # WASN'T SET, DAP WILL BE INITIALIZED TO
                ADS     FLGWRD10        # ASCENT VALUES BY 1/ACCS IN 2 SECONDS.

ULLGNOT         EXTEND                  # (1)
                INDEX   WHICH
                DCA     7               # LOAD AVEGEXIT WITH APPROPRIATE 2CADR
                DXCH    AVEGEXIT

                CAF     TWO             # 4.2SPOT RESTARTS IMMEDIATELY AT REDO4.2
                TS      L
                CS      TWO             # AND ALSO AT TIG-5 AT THE CORRECT TIME.
                DXCH    -PHASE4

                CS      TIME1
                TS      TBASE4          # SET TBASE4 FOR TIG-5 RESTART

REDO2.17        EXTEND
```

```
                DCA     NEGO            # CLEAR OUT GROUP 2 SO LAMBERT CAN START
                DXCH    -PHASE2         # IF NEEDED.

REDO4.2         CCS     PHASE5          # IF SERVICER GOING?
                TCF     TASKOVER        # YES, DON'T START IT UP AGAIN.

                TC      POSTJUMP
                CADR    PREREAD         # PREREAD END THIS TASK

#       *********************************

ULLGTASK        TC      ONULLAGE        # THIS COMES AT TIG-7.5 OR TIG-3.5
                TC      PHASCHNG
                OCT     1
                TCF     TASKOVER

#       *********************************

TIG-5           EXTEND
                DCA     NEGO            # INSURE THAT GROUP 3 IS INACTIVE.
                DXCH    -PHASE3

                CAF     5SEC
                TC      TWIDDLE
                ADRES   TIG-0

                TC      DOWNFLAG        # RESET IGNFLAG AND ASINFLAG
                ADRES   IGNFLAG         #       FOR LIGHT-UP LOGIC.
                TC      DOWNFLAG
                ADRES   ASTNFLAG

                INDEX   WHICH
                TCF     11

P40SJUNK        CCS     PHASE3          # (11) P40 AND P42.  S40.13 IN PROGRESS?
                TCF     DISPCHNG        # YES

                CAF     PRIO20
                TC      FINDVAC
                EBANK=  TTOGO
                2CADR   S40.13

                TC      PHASCHNG        # 3.5SPOT FOR S40.13
                OCT     00053
DISPCHNG        CS      VB99DEX         # (11)
                TS      DISPDEX
```

```
COMMON          TC      PHASCHNG        # RESTART TIG-0 (4.7SPOT)
                OCT     40074
                TCF     TASKOVER

#       ********************************

TIG-0           CS      FLAGWRD7        # SET IGNFLAG SINCE TIG HAS ARRIVED
                MASK    IGNFLBIT
                ADS     FLAGWRD7

                TC      CHECKMM         # IN P63 CASE, THROTTLE-UP IS ZOOMTIME
                DEC     63              # AFTER NOMINAL IGNITION, NOT ACTUAL
                TCF     IGNYET?
                CA      ZOOMTIME
                TC      WAITLIST
                EBANK=  DVCNTR
                2CADR   P63ZOOM

                TC      2PHSCHNG
                OCT     40033

                OCT     05014
                OCT     77777

IGNYET?         CAF     ASTNBIT         # CHECK ASTNFLAG:  HAS ASTRONAUT RESPONDED
                MASK    FLAGWRD7        # TO OUR ENGINE ENABLE REQUEST?
                EXTEND
                INDEX   WHICH
                BZF     12              # BRANCH IF HE HAS NOT RESPONDED YET

IGNITION        CS      FLAGWRD5        # INSURE ENGONFLG IS SET.
                MASK    ENGONBIT
                ADS     FLAGWRD5
                CS      PRIO30          # TURN ON THE ENGINE.
                EXTEND
                RAND    DSALMOUT
                AD      BIT13
                EXTEND
                WRITE   DSALMOUT
                EXTEND                  # SET TEVENT FOR DOWNLINK
                DCA     TIME2
                DXCH    TEVENT

                EXTEND                  # UPDATE TIG USING TGO FROM S40.13
                DCA     TGO
                DXCH    TIG
                EXTEND
                DCA     TIME2
                DAS     TIG
```

```
                CS      FLUNDBIT        # PERMIT GUIDANCE LOOP DISPLAYS
                MASK    FLAGWRD8
                TS      FLAGWRD8

                INDEX   WHICH
                TCF     13

P63IGN          EXTEND                  # (13)  INITIATE BURN DISPLAYS
                DCA     DSP2CADR
                DXCH    AVGEXIT

                CA      Z               # ASSASSINATE CLOKTASK
                TS      DISPDEX

                CS      FLAGWRD9        # SET FLAG FOR P70-P71
                MASK    LETABBIT
                ADS     FLAGWRD9

                CS      FLAGWRD7        # SET SWANDISP TO ENABLE R10.
                MASK    SWANDBIT
                ADS     FLAGWRD7

                CS      PULSES          # MAKE SURE DAP IS NOT IN MINIMUM-IMPULSE
                MASK    DAPBOOLS        # MODE, IN CASE OF SWITCH TO P66
                TS      DAPBOOLS

                EXTEND                  # INITIALIZE TIG FOR P70 AND P71.
                DCA     TIME2
                DXCH    TIG

                CAF     ZERO            # INITIALIZE WCHPHASE, AND FLPASS0
                TS      WCHPHASE
                TS      WCHPHOLD        # ALSO WHCPHOLD
                CA      TWO
                TS      FLPASS0

                TCF     P42IGN
P40IGN          CS      FLAGWRD5        # (13)
                MASK    NOTHRBIT
                EXTEND
                BZF     P42IGN
                CA      ZOOMTIME
                TC      WAITLIST
                EBANK=  DVCNTR
                2CADR   P40ZOOM

P63IGN1         TC      2PHSCHNG
                OCT     40033           # 3.3SPOT FOR ZOOM RESTART.
                OCT     05014           # TYPE C RESTARTS HERE IMMEDIATELY
                OCT     77777
```

```
                TCF     P42IGN
P12IGN          CAF     EBANK6
                TS      EBANK
                EBANK=  AOSQ

                CA      IGNAOSQ         # INITIALIZE DAP BIAS ACCELERATION
                TS      AOSQ            # ESTIMATES AT P12 IGNITION.
                CA      IGNAOSR
                TS      AOSR

                CAF     EBANK7
                TS      EBANK
                EBANK=  DVCNTR

ABRTIGN         CA      Z               # (13) KILL CLOKTASK
                TS      DISPDEX

                EXTEND                  # CONNECT ASCENT GYIDANCE TO SERVICER.
                DCA     ATMAGADR
                DXCH    AVGEXIT

                CS      FLAGWRD7        # ENABLE R10.
                MASK    SWANDBIT
                ADS     FLAGWRD7

P42IGN          CS      DRIFTBIT        # ENSURE THAT POWERED-FLIGHT SWITCHING
                MASK    DAPBOOLS        # CURVES ARE USED.
                TS      DAPBOOLS
                CAF     IMPULBIT        # EXAMINE IMPULSE SWITCH
                MASK    FLAGWRD2
                CCS     A
                TCF     IMPLBURN

DVMONCON        TC      DOWNFLAG
                ADRES   IGNFLAG         # CONNECT DVMON
                TC      DOWNFLAG
                ADRES   ASTNFLAG
                TC      DOWNFLAG
                ADRES   IDLEFLAG

                TC      PHASCHNG
                OCT     40054

                TC      FIXDELAY        # TURN ULLAGE OFF HALF A SECOND AFTER
                DEC     50              # LIGHT UP.

ULLAGOFF        TC      NOULLAGE

WAITABIT        EXTEND                  # KILL GROUP 4
                DCA     NEGO
```

```
                DXCH    -PHASE4

                TCF     TASKOVER

TIGTASK         TC      POSTJUMP        # (12)
                CADR    TIGTASK1

#       ********************************

                BANK    31
                SETLOC  P40S3
                BANK
                COUNT*  $$/P40

TIGTASK1        CAF     PRIO16
                TC      NOVAC
                EBANK=  TRKMKCNT
                2CADR   TIGNOW

                TC      PHASCHNG
                OCT     6               # KILL GROUP 6

                TCF     TASKOVER

#       ********************************

P63ZOOM         EXTEND
                DCA     LUNLANAD
                DXCH    AVEGEXIT

                TC      IBNKCALL
                CADR    FLATOUT
                TCF     P40ZOOMA

P40ZOOM         CAF     BIT13
                TS      THRUST
                CAF     BIT4

                EXTEND
                WOR     CHAN14

P40ZOOMA        TC      PHASCHNG
                OCT     3
                TCF     TASKOVER

                EBANK=  DVCNTR
LUNLANAD        2CADR   LUNLAND
```

```
ZOOM            =       P40ZOOMA
                BANK    36
                SETLOC  P40S
                BANK
                COUNT*  $$/P40

#       ********************************

COMFAIL         TC      UPFLAG          # (15)
                ADRES   IDLEFLAG
                TC      UPFLAG          # SET FLAG TO SUPPRESS CONFLICTING DISPLAY
                ADRES   FLUNDISP
                CAF     FOUR            # RESET DVMON
                TS      DVCNTR
                CCS     PHASE6          # CLOCKTASK ACTIVE?
                TCF     +3              # YES
                TC      BANKCALL        # OTHERWISE, START IT UP
                CADR    STCLOK1
 +3             CS      VB97DEX
                TS      DISPDEX
                TC      PHASCHNG        # TURN OFF GROUP 4.
                OCT     00004
                TCF     ENDOFJOB

COMFAIL1        INDEX   WHICH
                TCF     2

COMFAIL3        CA      Z               # (15)  KILL CLOKTASK USING Z
                TCF     +2

COMFAIL4        CS      CNTDNDEX
                TS      DISPDEX

                TC      DOWNFLAG        # RECONNECT DV MONITOR
                ADRES   IDLEFLAG
                TC      DOWNFLAG        # PERMIT GUIDANCE LOOP DISPLAYS
                ADRES   FLUNDISP
                TCF     ENDOFJOB

COMFAIL2        TC      PHASCHNG        # KILL ZOOM RESTART PROTECTION
                OCT     00003

                INHINT
                TC      KILLTASK        # KILL ZOOM IN CASE IT'S STILL TO COME
                CADR    ZOOM
                TC      IBNKCALL        # COMMAND ENGINE OFF
                CADR    ENGINOF4
                TC      UPFLAG          # SET THE DRIFT BIT FOR THE DAP.
                ADRES   DRIFTDFL
```

```
                TC      INVFLAG         # USE OTHER RCS SYSTEM
                ADRES   AORBTFLG
                TC      UPFLAG          # TURN ON ULLAGE
                ADRES   ULLAGFLG
                CAF     BIT1
                INHINT
                TC      TWIDDLE
                ADRES   TIG-5
                TCF     ENDOFJOB

#       ***********************************
#       SUBROUTINES OF THE IGNITION ROUTINE
#       ***********************************

INVFLAG         CA      Q
                TC      DEBIT
                COM
                EXTEND
                RXOR    LCHAN
                TCF     COMFLAG

#       ***********************************

NOULLAGE        CS      ULLAGER         # MUST BE CALLED IN A TASK OR UNDER INHINT
                MASK    DAPBOOLS
                TS      DAPBOOLS
                TC      Q

#       ***********************************

ONULLAGE        CS      DAPBOOLS        # TURN ON ULLAGE.  MUST BE CALLED IN
                MASK    ULLAGER         # A TASK OR WHILE INHINTED.
                ADS     DAPBOOLS
                TC      Q

#       ***********************************

STCLOK1         CA      ZERO            # THIS ROUTINE STARTS THE COUNT-DOWN
STCLOK2         TS      DISPDEX         # (CLOKTASK AND CLOKJOB).  SETTING
STCLOK3         TC      MAKECADR        # SETTING DISPDEX POSITIVE KILLS IT.
                TS      TBASE4          # RETURN SAVE (NOT FOR RESTARTS).
                EXTEND
                DCA     TIG
                DXCH    MPAC
                EXTEND
                DCS     TIME2
```

```
                DAS     MPAC            # HAVE TIG -- TIME2, UNDOUBTEDLY A + NUMBER
                TC      TPAGREE         # POSITIVE, SINCE WE PASSED THE
                CAF     1SEC            # 45 SECOND CHECK.
                TS      Q
                DXCH    MPAC
                MASK    LOW5            # RESTRICT MAGNITUDE OF NUMBER IN A
                EXTEND
                DV      Q
                CA      L               # GET REMAINDER
                AD      TWO
                INHINT
                TC      TWIDDLE
                ADRES   CLOKTASK
                TC      2PHSCHNG
                OCT     40036           # 6.3SPOT FOR CLOKTASK
                OCT     05024
                OCT     13000

                CA      TBASE4
                TC      BANKJUMP

CLOKTASK        CS      TIME1           # SET TBASE6 FOR GROUP 6 RESTART
                TS      TBASE6

                CCS     DISPDEX
                TCF     KILLCLOK
                NOOP
                CAF     PRIO27
                TC      NOVAC
                EBANK=  TTOGO
                2CADR   CLOKJOB

                TC      FIXDELAY        # WAIT A SECOND BEFORE STARTING OVER
                DEC     100
                TCF     CLOKTASK

KILLCLOK        EXTEND                  # KILL RESTART
                DCA     NEGO
                DXCH    -PHASE6
                TCF     TASKOVER

CLOKJOB         EXTEND
                DCS     TIG
                DXCH    TTOGO
                EXTEND
```

```
                DCA     TIME2
                DAS     TTOGO
                INHINT
                CCS     DISPDEX         # IF DISPDEX HAS BEEN SET POSITIVE BY A
                TCF     ENDOFJOB        # TASK OR A HIGHER PRIORITY JOB SINCE THE
                TCF     ENDOFJOB        # LAST CLOKTASK, AVOID USING IT AS AN
                COM                     # INDEX.
                RELINT                  # ***** DISPDEX MUST NEVER B -0 *****
                INDEX   A
                TCF     DISPNOT -1      # (-1 DUE TO EFFECT OF CCS)

VB97DEX         =       OCT35           # NEGATIVE OF THIS IS PROPER FOR DISPDEX

 -35            CS      ZERO            # INDICATE VERB 97 PASTE
                TS      NVWORD1
                CA      NVWORD  +2      # NVWORD+2 CONTAINS V06 & APPROPRIATE NOUN
                TC      BANKCALL
                CADR    CLOCPLAY
                TCF     STOPCLOK        # TERMINATE CLOKTASK ON THE WAY TO POOH
                TCF     COMFAIL1
                TCF     COMFAIL2

                                        # THIS DISPLAY IS CALLED VIA ASTNCLOK
 -25            CAF     V06N61          # IT IS PRIMARILY USED BY THE CREW IN P63
                TC      BANKCALL        # TO RESET HIS EVENT TIMER TO AGREE WITH
                CADR    REFLASH         # TIG.
                TCF     STOPCLOK
                TCF     ASTNRETN
                TCF     -6

CNTDNDEX        =       LOW4            # OCT17:  NEGATIVE PROPER FOR DISPDEX

 -17            INDEX   WHICH           # THIS DISPLAY COMES UP AT ONE SECOND
                # WAS CAF --- RSB 2009
                CA      0               # INTERVALS.  IT IS NORMALLY OPERATED
                TC      BANKCALL        # BETWEEN TIG-30 SECONDS AND TIG-5 SECONDS
                CADR    REGODSP         # REGODSP DOES ITS OWN TCF ENDOFJOB

VB99DEX         =       ELEVEN          # OCT13:  NEGATIVE PROPER FOR DISPDEX

V99RECYC        EQUALS

 -13            CS      BIT9            # INDICATE VERB 99 PASTE
                TS      NVWORD1
                INDEX   WHICH           # THIS IS THE "PLEASE ENABLE ENGINE"
                # WAS CAF --- RSB 2004
                CA      0               # DISPLAY; IT IS INITIATED AT TIG-5 SEC.
                TC      BANKCALL        # THE DISPLAY IS A V99NXX, WHERE XX IS
                CADR    CLOCPLAY        # NOUN THAT HAD PREVIOUSLY BEEN DISPLAYED
                TCF     STOPCLOK        # TERMINATE GOTOPOOH TURNS OFF ULLAGE.
                TCF     *PROCEED
                TCF     *ENTER
```

```
BLANKDEX        =       TWO             # NEGATIVE OF THIS IS PROPER FOR DISPDEX

 -2             TC      BANKCALL        # BLANK DSKY.  THE DSKY IS BLANKED FOR
                CADR    CLEANDSP        # 5 SECONDS AT TIG-35 TO INDICATE THAT
DISPNOT         TCF     ENDOFJOB        # AVERAGE G IS STARTING.

STOPCLOK        TC      NULLCLOK        # STOP CLOKTASK & TURN OFF ULLAGE ON THE
                TCF     GOTOPOOH        # WAY TO POO (GOTOPOOH RELINTS)

NULLCLOK        INHINT
                EXTEND
                QXCH    P40/RET
                TC      NOULLAGE        # TURN OFF ULLAGE ...
                TC      KILLTASK        #       DON'T LET IT COME ON, EITHER ...
                CADR    ULLGTASK
                TC      PHASCHNG        #               NOT EVEN IF THERE'S A RESTART.
                OCT     1
                CA      Z               # KILL CLOKTASK
                TS      DISPDEX
                TC      P40/RET

ASTNRETN        TC      PHASCHNG
                OCT     04024
                CAF     ZERO            # STOP DISPLAYING BUT KEEP RUNNING
                TS      DISPDEX
                CAF     PRIO13
                TC      FINDVAC
                EBANK=  STARIND
                2CADR   ASTNRET

                TCF     ENDOFJOB

*PROCEED        TC      UPFLAG
                ADRES   ASTNFLAG

                TCF     IGNITE

*ENTER          INHINT
                INDEX   WHICH
                TCF     3

GOPOST          CAF     PRIO12          # (3) MUST BE LOWER PRIORITY THAN CLOKJOB
                TC      FINDVAC
                EBANK=  TTOGO
                2CADR   POSTBURN
```

```
                INHINT                  # SET UP THE DAP FOR COASTING FLIGHT.
                TC      IBNKCALL
                CADR    ALLCOAST
                TC      NULLCLOK
                TC      PHASCHNG        # 4.13 RESTART FOR POSTBURN
                OCT     00134

                TCF     ENDOFJOB

GOCUTOFF        CAF     PRIO17          # (3)
                TC      FINDVAC
                EBANK=  TGO
                2CADR   CUTOFF

                TC      DOWNFLAG
                ADRES   FLUNDISP

                INHINT                  # SET UP THE DAP FOR COASTING FLIGHT.
                TC      IBNKCALL
                CADR    ALLCOAST
                TC      NULLCLOK
                TC      PHASCHNG
                OCT     07024
                OCT     17000
                EBANK=  TGO
                2CADR   CUTOFF

                TCF     ENDOFJOB

IGNITE          CS      FLAGWRD7        # (2)
                MASK    IGNFLBIT
                CCS     A
                TCF     IGNITE1
                CAF     BIT1
                INHINT
                TC      TWIDDLE
                ADRES   IGNITION

                CAF     OCT23           # IMMEDIATE RESTART AT IGNITION
                TS      L
                COM
                DXCH    -PHASE4

IGNITE1         CS      CNTDNDEX        # RESTORE OLD DISPLAY.
                TS      DISPDEX

                TCF     ENDOFJOB
```

```
#       ********************************

P40ALM          TC      ALARM           # PROGRAM SELECTION NOT CONSISTENT WITH
                OCT     1706            # VEHICLE CONFIGURATION

REP40ALM        CAF     V05N09          # (14)
                TC      BANKCALL
                CADR    GOFLASH

                TCF     GOTOPOOH        # V34E          TERMINATE
                TCF     +2              # PROCEED       CHECK FOR P42
                TCF     REP40ALM        # V32E          REDISPLAY ALARM

                INDEX   WHICH           # FOR P42, ALLOW CREW TO PROCEED EVEN
                TCF     14              # THOUGH VEHICLE IS UNSTAGED.

#       ********************************

                BANK    31
                SETLOC  P40S2
                BANK

                COUNT*  $$/P40

P40AUTO         TC      MAKECADR        # HELLO THERE.
                TS      TEMPR60         # FOR GENERALIZED RETURN TO OTHER BANKS.
P40A/P          TC      BANKCALL        # SUBROUTINE TO CHECK PGNCS CONTROL
                CADR    G+N,AUTO        # AND AUTO STABILIZATION MODES
                CCS     A               # +0 INDICATES IN PGNCS, IN AUTO
                TCF     TURNITON        # + INDICATES NOT IN PGNCS AND/OR AUTO
                CAF     APSFLBIT        # ARE WE ON THE DESCENT STAGE?
                MASK    FLGWRD10
                CCS     A
                TCF     GOBACK          # RETURN
                CAF     BIT5            # YES, CHECK FOR AUTO-THROTTLE MODE
                EXTEND
                RAND    CHAN30
                EXTEND
                BZF     GOBACK          # IN AUTO-THROTTLE MODE -- RETURN
TURNITON        CAF     P40A/PMD        # DISPLAYS V50N25 R1=203 PLEASE PERFORM
                TC      BANKCALL        # CHECKLIST 203 TURN ON PGNCS ETC.
                CADR    GOPERF1
                TCF     GOTOPOOH        # V34E TERMINATE
                TCF     P40A/P          # RECYCLE
GOBACK          CA      TEMPR60
                TC      BANKJUMP        # GOODBYE.  COME AGAIN SOON.

P40A/PMD        OCT     00203
```

```
                BANK    36
                SETLOC  P40S
                BANK

                COUNT*  $$/P40

#       ***********************************
#       CONSTANTS FOR THE IGNITION ROUTINE
#       ***********************************

SERVCADR        =       P63TABLE +7

P40ADRES        ADRES   P40TABLE

P41ADRES        ADRES   P41TABLE -5

P42ADRES        ADRES   P42TABLE

                EBANK=  DVCNTR
DSP2CADR        2CADR   P63DISPS -2

                EBANK=  DVCNTR
ATMAGADR        2CADR   ATMAG

?               =       GOTOPOOH

D29.9SEC        2DEC    2990

S24.9SEC        DEC     2490

4.9SEC          DEC     490

OCT20           =       BIT5

V06N61          VN      0661
```

```
# KILLTASK
# MOD NO:  NEW PROGRAM
# MOD BY:  COVELLI
#
# FUNCTIONAL DESCRIPTION:
#
#       KILLTASK IS USED TO REMOVE A TASK FROM THE WAITLIST BY SUBSTITUTING A NULL TASK CALLED `NULLTASK' (OF COURSE),
#       WHICH MERELY DOES A TC TASKOVER.  IF THE SAME TASK IS SCHEDULED MORE THAN ONCE, ONLY THE ONE WHICH WILL OCCUR
#       FIRST IS REMOVED.  IF THE TASK IS NOT SCHEDULED, KILLTASK TAKES NO ACTION AND RETURNS WITH NO ALARM.  KILLTASK
#       LEAVES INTERRUPTS INHIBITED SO CALLER MUST RELINT
#
# CALLING SEQUENCE
#       L       TC      KILLTASK        # IN FIXED-FIXED
#       L+1     CADR    ????????        # CADR (NOT 2CADR) OF TASK TO BE REMOVED.
#       L+2     (RELINT)                # RETURN
#
# EXIT MODE:  AT L+2 OF CALLING SEQUENCE.
#
# ERASABLE INITIALIZATION:  NONE.
#
# OUTPUT:  2CADR OF NULLTASK IN LST2
#
# DEBRIS:  ITEMP1 - ITEMP4, A, L, Q.

                EBANK=  LST2
                BLOCK   3               # KILLTASK MUST BE IN FIXED-FIXED.
                SETLOC  FFTAG6
                BANK
                COUNT*  $$/KILL
KILLTASK        CA      KILLBB
                INHINT
                LXCH    A
                INDEX   Q
                CA      0               # GET CADR.
                LXCH    BBANK
                TCF     KILLTSK2        # CONTINUE IN SWITCHED FIXED.

                EBANK=  LST2
KILLBB          BBCON   KILLTSK2

                BANK    27

                SETLOC  P40S1
                BANK
                COUNT*  $$/KILL

KILLTSK2        LXCH    ITEMP2          # SAVE CALLER'S BBANK
```

```
                INCR    Q
                EXTEND
                QXCH    ITEMP1          # RETURN 2ADR IN ITEMP1,ITEMP2

                TS      ITEMP3          # CADR IS IN A
                MASK    LOW10
                AD      BIT11
                TS      ITEMP4          # GENADR OF TASK

                CS      LOW10
                MASK    ITEMP3
                TS      ITEMP3          # FBANK OF TASK

                ZL
ADRSCAN         INDEX   L
                CS      LST2
                AD      ITEMP4          # COMPARE GENADRS
                EXTEND
                BZF     TSTFBANK        # IF THEY MATCH, COMPARE FBANKS
LETITLIV        CS      LSTLIM
                AD      L
                EXTEND                  # ARE WE DONE?
                BZF     DEAD            # YES -- DONE, SO RETURN
                INCR    L
                INCR    L
                TCF     ADRSCAN         # CONTINUE LOOP.

DEAD            DXCH    ITEMP1
                DTCB

TSTFBANK        CS      LOW10
                INDEX   L
                MASK    LST2    +1      # COMPARE FBANKS ONLY.
                EXTEND
                SU      ITEMP3
                EXTEND
                BZF     KILLDEAD        # MATCH -- KILL IT.
                TCF     LETITLIV        # NO MATCH -- CONTINUE.

KILLDEAD        CA      TCTSKOVR
                INDEX   L
                TS      LST2            # REMOVE TASK BY INSERTING TASKOVER
                TCF     DEAD

LSTLIM          EQUALS  BIT5            # DEC 16
```

```
# PROGRAM DESCRIPTION: P40BOTH           DECEMBER 22, 1966
# MOD 03 BY PETER ADLER                  MARCH 3, 1967
# CALLED VIA JOB FROM V37E
#
# FUNCTIONAL DESCRIPTION
#
#       1)      TO COMPUTE A PREFERRED IMU ORIENTATION AND A PREFERRED VEHICLE ATTITUDE FOR A LM DPS
#               THRUSTING MANEUVER.
#       (THERE IS NO ITEM #2 IN THE ORIGINAL PROGRAM LISTING --- RSB 2009.)
#       3)      TO DO THE VEHICLE MANEUVER TO THE THRUSTING ATTITUDE.
#       4)      TO CONTROL THE PGNCS DURING COUNTDOWN, IGNITION, THRUSTING, AND THRUST TERMINATION OF A
#               PGNCS CONTROLLED DPS MANEUVER.
#       5)      IN POSTBURN --- ZERO RENDEZVOUS COUNTER, MAINTAIN VG CALCULATIONS FOR POSSIBLE RCS MANEUVER,
#               SET MAXIMUM DEADBAND IN DAP, RESET STEERLAW CSTEER TO ZERO.
#
#       NOTE:   P42, WHICH IS IN THIS LOG SECTION, DOES THE SAME FOR AN APS BURN, AND P41 DOES 1-3 FOR
#               RCS PLUS DISPLAYS PARAMETERS FOR MANUAL CONTROL.
#
# SUBROUTINES USED
#
#       R02             IMU STATUS CHECK
#       S40.1           COMPUTATION OF THRUST DIRECTION
#       S40.13          LENGTH OF BURN
#       S40.2,3         PREFERRED IMU ORIENTATION
#       S40.8           X PRODUCT STEERING
#       S40.9           LAMBERT VTOGAIN
#       R60LEM          ATTITUDE MANEUVER
#       LEMPREC         EXTRAPOLATE STATE VECTOR
#       PREREAD         AVERAGE G, SERVICER
#       ALLCOAST        DAP COASTING INITIALIZATION
#       CLOKTASK        ERGO CLOCKJOB -- COUNT DOWN
#       PHASCHANG, INTPRET, FLAGUP, FLAGDOWN, WAITLIST, LONGCALL, GOFLASH, GOFLASHR, GOPERF1, ALARM,
#       PRIOLARM, GOTOPOOH, ENDOFJOB, BANKCALL, SETMAXDB, SETMINDB, CHECKMM, FLATOUT, OUTFLAT,
#       KILLTASK, SGNAGREE, TPAGREE, ETC.
#
# RESTARTS VIA GROUP 4
#
# DISPLAYS
#
#       V50N25  203 A/P TO PGNCS, AUTO-THROTTLE MODE, AUTO ATTITUDE CONTROL
#       V06N40  TTI, VG, DELTAVM (DISPLAYED ONCE/SECOND BY CLOKTASK)
#       V50N99  PLEASE PERFORM ENGINE ON ENABLE
#       V06N40  TG (TIME TO GO TO CUTOFF), VG, DELTAVM -- ONCE/SECOND
#       V16N40  FINAL VALUES OF TG, VG, DELTAVM
#       V16N85  COMP OF VG (BODY AXES) FOR POSS. RCS MANUAL MANEUVER
#       V05N09  POSSIBLE ALARMS
#       V50N07  PLEASE SELECT P00
#
```

```
#       VIA R30
#
#       V06N44  HAPO, PERI, TFF
#       V06N35  TIME TO PERIGEE, HMS
#
# ALARM OR ABORT EXIT MODES
#
#       PROGRAM ALARM, FLASHING DISPLAY OF ALARM CODE 1706 IF P40 SELECTED WITH DESCENT UNIT STAGED.
#       V34E (TERMINATE) IS THE ONLY RESPONSE ACCEPTED.  TC GOTOPOOH.
#
#       PROGRAM ALARM, FLASH CODE 1703:  TIG LESS THAN 45 SECS AWAY.  V34E= GOTOPOOH OR V33E= SLIP
#       TIG BY 45 SECS.
#
# ERASABLE INITIALIZATION
#
# DEBRIS
#
# OUTPUT
#
#       SEE SUBROUTINES E.G.:  S40.1, S40.2,3, S40.13, S40.8, S40.9, TRIMGIMB
#       XDELVFLG = 1 FOR EXT DELV COMPUTATION
#                = 0 FOR AIMPT (LAMBERT COMP

                COUNT*  $$/P40
                EBANK=  WHICH

                BANK    36
                SETLOC  P40S
                BANK

P40LM           TC      PHASCHNG
                OCT     04024

                CAF     P40ADRES        # INITIALIZATION FOR BURNBABY
                TS      WHICH

                CA      FLGWRD10
                MASK    APSFLBIT
                CCS     A
                TCF     P40ALM
                TC      BANKCALL        # GO DO IMU STATUS CHECK ROUTINE.
                CADR    R02BOTH

                CS      DAPBOOLS        # INITIALIZE DVMON
                MASK    CSMDOCKD
                CCS     A
                CAF     THRESH1
                AD      THRESH3
                TS      DVTHRUSH
                CAF     FOUR
                TS      DVCNTR
```

```
                TC      INTPRET         # LOAD CONSTANTS FOR DPS BURN
                VLOAD   CLEAR           # LOAD F, MDOT, TDECAY
                        FDPS
                        NOTHROTL
                STORE   F
                SLOAD
                        DPSVEX
P40IN           DCOMP   SR1
                STCALL  VEX             # LOAD EXHAUST VELOCITY FOR TGO COMP.
                        S40.1           # COMPUTES UT AND VGTIG
                CALL
                        S40.2,3         # COMPUTES PREFERRED IMU ORIENTATION
                EXIT

                INHINT
                TC      IBNKCALL
                CADR    PFLITEDB        # ZERO ATTITUDE ERRORS, SET DB TO ONE DEG.

                TC      P40SXT4

#       ********************************

                TCF     BURNBABY

#       ********************************

P40SXT4         EXTEND
                QXCH    P40/RET
P41MANU         RELINT

                TC      DOWNFLAG        # CLEAR 3AXISFLG -- R60 USE VECPOINT.
                ADRES   3AXISFLG

                TC      BANKCALL
                CADR    R60LEM          # DO ATTITUDE MANEUVER ROUTINE
                TC      P40/RET

                EBANK=  TRKMKCNT
POSTBURN        CA      Z
                TS      DISPDEX
                EXTEND
                DCA     ACADN85
                DXCH    AVEGEXIT
                CAF     V16N40
                TC      BANKCALL
                CADR    GOFLASHR
                TC      TERM40
                TCF     TIGNOW
                TC      POSTBURN
```

```
P40PHS1         TC      PHASCHNG
                OCT     00014
                TCF     ENDOFJOB

TIGNOW          INHINT
                TC      IBNKCALL
                CADR    ZATTEROR
                TC      IBNKCALL
                CADR    SETMINDB
                RELINT
                CAF     V16N85B
                TC      BANKCALL
                CADR    REFLASHR
                TC      TERM40
                TCF     TERM40
                TC      -5

                TCF     P40PHS1

TERM40          EXTEND
                DCA     SERVCADR
                DXCH    AVEGEXIT
                CAF     ZERO
                TS      TRKMKCNT        # ZERO RENDZVS CNTERS
                CA      Z
                TS      DISPDEX
                INHINT
                TC      IBNKCALL
                CADR    RESTORDB
                RELINT
                TC      GOTOPOOH

                EBANK=  WHICH
                COUNT*  $$/P41
P41LM           CAF     P41ADRES        # INITIALIZATION FOR BURNBABY
                TS      WHICH

                TC      BANKCALL
                CADR    R02BOTH

                TC      INTPRET         # BOTH LM
                BON     DLOAD           # IF NJETSFLAG IS SET, LOAD Z JET F
                        NJETSFLG
                        P41FJET1
                        FRCS4           # IF NJETSFLG IS CLEAR, LOAD 4 JET F

P41FJET         STCALL  F
                        P41IN
P41FJET1        DLOAD
```

```
                        FRCS2
                STORE   F

P41IN           CALL
                        S40.1           # BOTH
P41NORM         CALL
                        S40.2,3         # CALCULATE PREFERRED IMU ORIENTATION AND
                EXIT                    # SET PFRATFLG.

                INHINT
                TC      IBNKCALL
                CADR    ZATTEROR        # ZERO ATTITUDE ERRORS
                TC      IBNKCALL
                CADR    SETMINDB        # SET 0.3 DEGREE DEADBAND
                TC      P40SXT4

                TC      INTPRET
                VLOAD   CALL            # TRANSFORM VELOCITY-TO-BE-GAINED AT TIG
                        VGTIG           # FROM REFERENCE COORDINATES TO LM BODY-
                        S41.1           # AXIS COORDINATES FOR V16N85 DISPLAY.
                STORE   VGBODY          # (SCALED AT 2 (+7) METERS/CENTISECOND)
                EXIT

                CAF     V16N85B
                TC      BANKCALL
                CADR    GODSPRET

                CAF     PRIO5
                TS      DISPDEX         # FOR SAFETY ONLY
                TC      FINDVAC
                EBANK=  VGPREV
                2CADR   DYNMDISP

                TC      2PHSCHNG
                OCT     00076           # GROUP 6 RESTARTS AT REDO6.7
                OCT     04024           # GROUP 4 RESTARTS HERE

#       ********************************

                TCF     B*RNB*B*

#       ********************************

BLNKWAIT        CAF     1SEC
                TC      BANKCALL
                CADR    DELAYJOB

REDO6.7         CA      DISPDEX         # ON A RESTART, DO NOT PUT UP DISPLAY IF
                AD      TWO             # BLANKING (BETWEEN TIG-35 AND TIG-30)

#       ********************************
```

```
                EXTEND
                BZF     BLNKWAIT

                CAF     V16N85B
                TC      BANKCALL
                CADR    GODSPRET

                CAF     PRIO5
                TC      PRIOCHNG

DYNMDISP        CA      DISPDEX         # A NON-POSITIVE DISPDEX INDICATES PAST
                EXTEND                  # TIG-35, SO SERVICER WILL BE DOING THE
                BZMF    ENDOFJOB        # UPDATING OF NOUN 85.  STOP DYNMDISP.
                TC      INTPRET
                VLOAD   CALL
                        VGPREV
                        S41.1
                STORE   VGBODY
                EXIT
                CAF     1SEC
                TC      BANKCALL
                CADR    DELAYJOB
                TCF     DYNMDISP

CALCN85         TC      INTPRET
                CALL
                        UPDATEVG
                VLOAD   CALL
                        VGPREV
                        S41.1
                STORE   VGBODY
                EXIT
                TC      POSTJUMP
                CADR    SERVEXIT

                COUNT*  $$/P42
                EBANK=  WHICH

P42LM           TC      PHASCHNG
                OCT     04024

                CAF     P42ADRES        # INITIALIZATION FOR BURNBABY.
                TS      WHICH

                CS      FLGWRD10
                MASK    APSFLBIT
                CCS     A
                TC      P40ALM
P42STAGE        TC      BANKCALL
```

```
                CADR    R02BOTH
                CAF     THRESH2         # INITIALIZE DVMON
                TS      DVTHRUSH
                CAF     FOUR
                TS      DVCNTR

                TC      INTPRET
                SET     VLOAD           # LOAD FAPS, MDOTAPS, AND ATDECAY INTO
                        AVFLAG          # F, MDOT, AND TDECAY BY VECTOR LOAD.
                        FAPS
                STORE   F
                SLOAD   GOTO
                        APSVEX
                        P40IN

                EBANK=  WHICH

                COUNT*  $$/P47
P47LM           TC      BANKCALL
                CADR    R02BOTH
                TC      INTPRET
                CALRB
                        MIDTOAV2

                CA      MPAC +1
                TC      TWIDDLE
                ADRES   STARTP47

                TCF     ENDOFJOB

STARTP47        TC      PHASCHNG
                OCT     05014
                OCT     77777

                EXTEND
                DCA     ACADN83
                DXCH    AVEGEXIT
                CAF     PRIO20
                TC      FINDVAC
                EBANK=  DELVIMU
                2CADR   P47BODY

                TCF     REDO4.2         # CHECKS PHASE 5 AND GOES TO PREREAD
                                        # SEE TIG-30 IN BURNBABY

CALCN83         TC      INTPRET
                VLOAD   VAD
                        DELVCTL
                        DELVREF
                STORE   DELVSIN         # TEMP STORAGE FOR RESTARTS
```

```
                CALL
                        S41.1
                STORE   DELVIMU
                EXIT
                TC      PHASCHNG
                OCT     10035           # REREADAC AND HERE

                TC      INTPRET
                VLOAD
                        DELVSIN
                STORE   DELVCTL
                EXIT

                TC      POSTJUMP
                CADR    SERVEXIT

P47BOD          CAF     V1683
                TC      BANKCALL
                CADR    GOFLASHR
                TC      GOTOPOOH
                TC      GOTOPOOH

                TCF     P47BODY

                TCF     P40PHS1
P47BODY         TC      INTPRET
                VLOAD
                        HI6ZEROS
                STORE   DELVIMU
                STORE   DELVCTL
                EXIT
                TC      P47BOD

                COUNT*  $$/P40
IMPLBURN        CA      TGO     +1
                TC      GETDT
                TC      TWIDDLE
                ADRES   ENGOFTSK
                TC      DOWNFLAG        # TURN OFF IGNFLAG
                ADRES   IGNFLAG
                TC      DOWNFLAG        # TURN OFF ASTNFLG
                ADRES   ASTNFLAG
                TC      DOWNFLAG        # TURN OFF IMPULSW
                ADRES   IMPULSW
                TC      PHASCHNG        # RESTART PROTECT ENGOFTSK (ENGINOFF)
                OCT     40114

                TC      FIXDELAY        # WAIT HALF A SECOND
                DEC     50
```

```
                TC      NOULLAGE        # TURN OFF ULLAGE

                TC      TASKOVER

ENGOFTSK        TC      IBNKCALL        # THIS CODING ALLOWS ENGINOFF ET AL TO BE
                CADR    ENGINOFF        # USED BOTH BY WAITLIST AND BY TC IBNKCALL
                TC      TASKOVER

ENGINOFF        CAF     PRIO12          # MUST BE LOWER PRIO THAN CLOCKJOB
                TC      FINDVAC
                EBANK=  TRKMKCNT
                2CADR   POSTBURN

ENGINOF2        CAF     BIT1
                TC      WAITLIST
                EBANK=  OMEGAQ
                2CADR   COASTSET

ENGINOF1        CS      FLAGWRD7        # SET THE IDLE BIT.
                MASK    IDLEFBIT
                ADS     FLAGWRD7

                TC      NOULLAGE

ENGINOF4        EXTEND
                DCA     TIME2
                DXCH    TEVENT

ENGINOF3        CS      ENGONBIT        # INSURE ENGONFLG IS CLEAR.
                MASK    FLAGWRD5
                TS      FLAGWRD5
                CS      PRIO30          # ENGINOF3 IS USED AS A PRE-ENGINE ARM
                EXTEND                  # SUBROUTINE.
                RAND    DSALMOUT
                AD      PRIO20          # TURN OFF THE ENGINE -- DPS OR APS
                EXTEND
                WRITE   DSALMOUT

                CS      DAPBOOLS        # TURN OFF TRIM GIMBAL
                MASK    USEQRJTS
                ADS     DAPBOOLS

                CS      HIRTHROT        # ZERO AUTO-THROTTLE WHENEVER THE ENGINE
                TS      THRUST          # IS TURNED OFF.
                CAF     BIT4            # THE HARDWARE DOES SO ONLY WHEN THE
                EXTEND                  # ENGINE IS DISARMED.
                WOR     CHAN14

                TC      ISWRETRN
```

```
COASTSET        TC      IBNKCALL        # DO DAP COASTING INITIALIZATION
                CADR    ALLCOAST
                TC      TASKOVER

                EBANK=  OMEGAQ
UPDATEVG        STQ     CALL
                        QTEMP1
                        S40.8           # X-PRODUCT STEERING
                BON     BON
                        XDELVFLG
                        QTEMP1
                        NORMSW
                        180SETUP
                DLOAD   DSU
                        PIPTIME
                        TIGSAVE
                DSU     BMN
                        TNEWA
                        GETRANS
                DLOAD   DAD
                        TIGSAVE
                        TNEWA
                STORE   TIGSAVEP
180SETUP        EXIT
                CCS     PHASE2
                TCF     NO.9
                CAF     PRIO10
                INHINT
                TC      FINDVAC
                EBANK=  VG
                2CADR   S40.9           # LAMBERT VTOGAIN

                TC      2PHSCHNG
                OCT     00172           # 2.17SPOT FOR S40.9
                OCT     10035           # HERE AND REREADAC AFTER RESTART

ENDSTEER        TC      INTPRET
                DLOAD
                        TIGSAVEP
                STOVL   TIGSAVE
                        RN
                STOVL   RINIT
                        VN
                STORE   VINIT
GETRANS         DLOAD   DSU
                        TPASS4
                        PIPTIME
                STCALL  DELLT4
                        QTEMP1
```

```
NO.9            TC      INTPRET
                GOTO
                        QTEMP1
STEERING        TC      INTPRET

                CALL
                        UPDATEVG
                EXIT

                EBANK=  DVCNTR
NSTEER          INHINT
                CA      EBANK7
                TS      EBANK
                CS      FLAGWRD2        # CHECK IMPULSE SWITCH.  IT IS SET EITHER
                MASK    IMPULBIT        # BY S40.13 IF TBURN<6 SECS OR BY S40.8 IF
                CCS     A               # STEERING IS ALMOST DONE.

                TCF     +5              # IMPULSW = 0   EXIT
                CS      FLAGWRD7        # IMPULSW = 1   WHY?  CHECK IDLEFLAG
                MASK    IDLEFBIT        #       (IDLEFLAG = 0 --> DVMON ON)
                CCS     A
                TCF     +3              # DVMON ON --> THRUSTING --> IMPULSW VIA S40.8
                TC      POSTJUMP        # DVMON OFF --> IMPULSW ON VIA S40.13 --> EXIT
                CADR    SERVEXIT

                TC      IBNKCALL
                CADR    STOPRATE

                TC      DOWNFLAG        # TURN OFF IMPULSW
                ADRES   IMPULSW

                TC      UPFLAG
                ADRES   IDLEFLAG        # TURN OFF DVMON

                INHINT
                EXTEND
                DCA     TIG
                DXCH    MPAC
                EXTEND
                DCS     TIME2
                DAS     MPAC
                TC      TPAGREE
                CAE     MPAC    +1
                TC      GETDT
                TC      TWIDDLE
                ADRES   ENGOFTSK
                TC      2PHSCHNG
                OCT     40114           # ENGOFTSK (ENGINOFF)
                OCT     00035           # SERVICER -- REREADAC
```

```
                TCF     ENDOFJOB

GETDT           CCS     A
                TCF     +3
                TCF     +2
                CAF     ZERO
                AD      ONE
                XCH     L
                CAF     ZERO
                DXCH    TGO
                CA      TGO     +1
                TC      Q

# ***************************************

SEC15DP         OCT     00000           # DON'T SEPARATE
SEC15           DEC     1500            # DON'T SEPARATE
SEC30DP         2DEC    3000

SEC45DP         OCT     00000           # DON'T MOVE FROM JUST BEFORE SEC45
SEC45           DEC     4500
5SECDP          OCT     00000           # DON'T MOVE FROM JUST BEFORE 5SEC
5SEC            DEC     500
26SECS          DEC     2600
V16N40          VN      1640
V16N85B         VN      1685
V1683           VN      1683
SEC01           =       1SEC
ACADN85         =       P41TABLE +2

                EBANK=  DELVIMU
ACADN83         2CADR   CALCN83

# *******************************************
```

```
# PROGRAM DESCRIPTION: S40.1             DATE: 15 NOV 66
# MOD NO2                                LOG SECTION P40-P47
# MOD BY ZELDIN AND ADAPTED BY TALAYCO
#
# FUNCTIONAL DESCRIPTION
#       COMPUTE INITIAL THRUST DIRECTION(UT) AND INITIAL VALUE OF VG
#       VECTOR(VGTIG).
#
# CALLING SEQUENCE
#       L       CALL
#       L+1             S40.1
#
# NORMAL EXIT MODE
#       AT L+2 OF CALLING SEQUENCE (GOTO L+2) NORMAL RETURN OR
#       ERROR RETURN IF NOSOFLAG =1
#
# SUBROUTINES CALLED
#       LEMPREC
#       INITVEL
#       CALCGRAV
#       MIDGIM
#
# ALARM OR ABORT EXIT MODES
#       L+2 OF CALLING SEQUENCE, UNSOLVABLE CONIC IF NOSOFLAG=1
#
# ERASABLE INITIALIZATION REQUIRED
#       WEIGHT/G        ANTICIPATED VEHICLE MASS        DP B16 KGM
#       XDELVFLG        1=DELTA-V MANEUVER, 0=AIMPT STEER
#       F               THRUST FOR ENGINE USED
#   IF DELTA-V MANEUVER:
#       DELVSIN         SPECIFIED DELTA-V REQUIRED IN
#                       INERTIAL COORDS. OF ACTIVE VEHICLE
#                       AT TIME OF IGNITION             VECTOR B7 M/CS
#       DELVSAB         MAG. OF DELVSIN                 DP B7 M/CS
#       RTIG            POSITION AT TIME OF IGNITION    VECTOR B29 M
#       VTIG            VELOCITY AT TIME OF IGNITION    VECTOR B7 M/CS.
#   IF AIMPT STEER:
#       TIG             TIME OF IGNITION                DP B28 CS
#       RTARG           POSITION TARGET TIME            VECTOR B29 M
#       CSTEER          C FOR STEER LAW                 DP B2
#       DLTARG          TARGET TIME-IGNITION TIME       DP B28 CS
#
# OUTPUT
#       UT              DESIRED THRUST DIRECTION        VECT. B2 M/(CS.CS)
#       VGTIG           INITIAL VALUE OF VELOCITY
#                       TO BE GAINED (INERT. COORD.)    VECTOR B7 M/CS
#       DELVLVC         VGTIG IN LOC. VERT. COORDS.     B7 M/CS
#       BDT             V REQUIRED AT TIG -V REQUIRED AT (TIG-2SEC)
#           -GDT        FOR S40.13                      VECT B7 M/CS
#       RTIG            CALC IN S40.1B (AIMPT) FOR S40.2,3      VECTOR B27M
#                       POSITION AT TIME OF IGNITION
#
# DEBRIS        QTEMP1
#               MPAC, QPRET
#               PUSHLIST

                BANK    14
                SETLOC  P40S1
                BANK
```

1412THE

1

```
                COUNT*  $$/S40.1
S40.1           STQ     DLOAD
                        QTEMP
                        TIG
                STORE   TIGSAVE
DELVTEST        BOFF
                        XDELVFLG
                        S40.1B
CALCTHET        SETPD   VLOAD
                        0
                        VTIG
                STORE   VINIT
                VXV     UNIT
                        RTIG
                STOVL   UT              # UP IN UT
                        RTIG
                STORE   RINIT
                VSQ     PDDL
                        36D
                DMP     DDV
                        THETACON
                DMP     DMP
                        DELVSAB
                        WEIGHT/G
                DDV
                        F
                STOVL   14D
                        DELVSIN

                DOT     VXSC
                        UT
                        UT
                VSL2    PUSH            # (DELTAV.UP)UP SCALED AT 2(+7) P.D.L. 0
                BVSU    PDDL            # DELTA VP SCALED AT 2(+7) P.D.L. 6
                        DELVSIN
                        14D
                SIN     PDVL
                        6D
                VXV     UNIT
                        UT
                VXSC    STADR
                STOVL   VGTIG           # UNIT(VP X UP)SIN(THETAT/2) IN VGTIG.
                UNIT    PDDL            # UNIT(DELTA VP) IN P.D.L. 6
                        14D
                COS     VXSC
                VAD     VXSC
                        VGTIG
                        36D
                VSL2    VAD
                STADR
```

```
                STORE   VGTIG           # VG IGNITION SCALED AT 2(+7) M/CS

                UNIT
                STOVL   UT              # THRUST DIRECTION SCALED AT 2(+1)
                        VGTIG
                PUSH    CALL
                        GET.LVC         # VGTIG IN LV COOR AT 2(+7) M/CS IN DELVLVC
                GOTO
                        QTEMP
S40.1B          DLOAD
                        TIG
                STORE   TDEC1
                BDSU
                        TPASS4
                STCALL  DELLT4          # INTERCEPT TIME -- TIG.
                        LEMPREC
                VLOAD   SETPD           # LOAD STATE VECTOR AT TIG FOR INITVEL.
                        RATT
                        0
                STORE   RTIG
                STORE   RINIT
                UNIT
                STOVL   UNIT/R/
                        VATT
                STORE   VTIG
                STORE   VINIT
                DLOAD   PDDL            # NUMIT = 0
                        ZEROVECS
                        EPS1
                BOFF    DAD
                        NORMSW
                        SMALLEPS
                        EPS2            # EPSILON4 = 10 DEGREES OR 45 DEGREES.
SMALLEPS        PUSH    SXA,1
                        RTX1
                SXA,2   CALL
                        RTX2
                        INITVEL
                VLOAD   PUSH
                        DELVEET3        # VGTIG = VR - VN.
                STORE   VGTIG
                UNIT                    # UT = UNIT (VGTIG)
                STODL   UT
                        36D
                STCALL  VGDISP          # CONVERT VGTIG (IN PUSHLIST) TO LOCAL
                        GET.LVC         # VERTICAL COORDINATES.
                GOTO
                        QTEMP

EPS1            2DEC*   2.777777778 E-2*        # 10 DEGREES AT 1 REVOLUTION
```

```
EPS2            2DEC*    9.722222222 E-2*        # 35 DEGREES AT 1 REVOLUTION.
THETACON        2DEC     .31830989 B-8
```

```
# SUBROUTINE NAME:  S40.2,3             MOD. NO. 3, DATE APRIL 4, 1967
# MODIFICATION BY:  JONATHON D. ADDELSTON (ADAMS ASSOCIATES)
# MOD. NO. 4:  JULY 18, 1967: PETER ADLER (MIT/IL)
# MOD. NO. 5:  OCTOBER 18, 1967:  PETER ADLER (MIT/IL)
# ORIGINALLY BY:  SAYDEAN ZELDIN (MIT INSTRUMENTATION LAB) AND RICHARD TALAYCO (SYSTEM DELVELOPMENT CORP)
#
# S40.2,3 COMPUTES "POINTVSM" WHICH IS THE HALF-UNIT DESIRED THRUST VECTOR IN STABLE-MEMBER COORDINATES FROM "UT"
# WHICH IS THE SAME VECTOR IN REFERENCE COORDINATES.  IT DETERMINES THE CORRECT VALUES FOR "SCAXIS" USING THE +X
# AXIS FOR DPS, APS, AND RCS BURNS.  THE "WINGS-LEVEL HEADS-UP" LM ORIENTATION IS THEN COMPUTED IN REFERENCE
# COORDINATES.  THESE VECTORS ALSO DEFINE THE "PREFERRED IMU ORIENTATION".  UPON COMPLETION OF THIS CALCULATION,
# THE "PREFERRED ATTITUDE COMPUTED" FLAG IS SET (PFRATFLG).
#
# CALLING SEQUENCE:
#       L       CALL                    # INTERPRETIVE CALL.
#       L +1            S40.2,3
#       L +2    (RETURN)                # GIMBAL ANGLE VECTOR IN MPAC.
#
# SUBROUTINES CALLED:  NONE.
#
# NORMAL RETURN:  L +2 (SEE CALLING SEQUENCE ABOVE).
#
# ALARM/ABORT MODES:  NONE.
#
# INPUT:
#       1.      REFSMMAT        MATRIX FROM REFERENCE TO STABLE-MEMBER COORDINATES SCALED AT 2.
#       2.      UT              HALF-UNIT DESIRED THRUST DIRECTION.
#       3.      RTIG            POSITION AT TIG IN REFERENCE COORDINATES.
#
# OUTPUT:
#       1.      `XSCREF'        WINGS-LEVEL HEADS-UP LM ORIENTATION
#               `YSCREF'        IN REFERENCE COORDINATES
#               `ZSCREF'        (PREFERRED IMU ORIENTATION).
#       2.      POINTVSM        DESIRED THRUST DIRECTION IN STABLE-MEMBER COORDINATES.
#       3.      SCAXIS          HALF-UNIT OF AXIS TO ALIGN IN STABLE-MEMBER COORDINATES.
#       4.      PFRATFLG        INTERPRETIVE FLAG.  ON: PREFERRED ORIENTATION COMPUTED; OFF: NOT COMPUTED.
#
# DEBRIS:  NONE
```

```
                COUNT*   $$/S40.2
S40.2,3         VLOAD                      # UT:  DESIRED THRUST DIRECTION (HALF-UNIT)
                         UT                # (PUT INTO TOP OF PUSH-DOWN-LIST.)
                MXV      VSL1              # TRANSFORM THRUST DIRECTION TO STABLE-
                         REFSMMAT          # MEMBER FROM REFERENCE COORDS (RESCALE).
                STOVL    POINTVSM          # SAVE FOR "VECPOINT" ROUTINE (LEMMANU).
                         UNITX             # SCAXIS SET TO +X, FOR P40 AND P42 AND
                STOVL    SCAXIS            # FOR P41 IF RCS NOT -X,+Y,-Y,+Z,-Z.

                         UT                # ASSUME +X BURN ALWAYS, EVEN FOR RCS.
PLUSX           STORE    XSCREF            # XSCREF = UT (DESIRED THRUST DIRECTION)
                VXV      UNIT              # RTIG = POSITION AT TIME-OF-IGNITION.
                         RTIG              # YSCREF = UNIT(UT X RTIG)
                PDDL     BHIZ
                         36D               # TEST MAGNITUDE OF UT X RTIG
                         FIXY              # IF SMALL, USE UT X VTIG AS YSC
STORY           VLOAD    STADR
                STORE    YSCREF
                VXV      VSL1              # COMPUTE (YSCREF X XCREF), BUT FOR A
                         XSCREF            # RIGHT HANDED SYSTEM, NEED (X CROSS Y).
                VCOMP                      # ZSCREF = - (YSCREF X XSCREF)
                STORE    ZSCREF            #        = + (XSCREF X YSCREF)

                SET      RVQ
                         PFRATFLG
FIXY            VLOAD    VXV               # IN THIS CASE,
                         XSCREF            # YSCREF = UNIT(XSCREF X VTIG)
                         VTIG
                UNIT     PUSH
                GOTO
                         STORY
```

```
# SUBROUTINE S40.8
# MODIFIED APRIL 3, 1968 BY PETER ADLER, MIT/IL
#
# DESCRIPTION
#       S40.8 UPDATES THE VELOCITY-TO-BE-GAINED VECTOR, VG, (AND FOR LAMBERT TARGETTED BURNS ALSO EXTRAPOLATES VG
#       USING THE BDT VECTOR) COMPUTES THE TIME FOR ISSUING THE ENGINE OFF COMMAND, TGO, AND CALLS THE ROUTINE
#       "FINDCDUW", WHICH GENERATES STEERING COMMANDS FOR THE DAP.
#
# CALLING SEQUENCE
#       L-1     CALL
#       L               S40.8
#       L+1                     INTERPRETIVE RETURN
#
# ALARM
#       IF VG . DELVREF IS NEGATIVE (VG AND DELVREF OVER 90 DEGREES APART), BYPASS TGO AND STEERING COMPUTATIONS
#       AND SET ALARM 1407.  RETURN TO CALLER NORMALLY.
#
# INPUT AND INITIALIZATION
#       VGPREV          REFERENCE       2(7) M/CS
#       DELVREF         REFERENCE       2(7) M/CS
#       BDT             REFERENCE       2(7) M/CS
#       TDECAY          TAIL-OFF TIME   2(28) CS
#       XDELVFLG        1 = EXTERNAL DELTA-V; 0 = LAMBERT (AIMPOINT)
#       STEERSW         1 = DO STEERING AND TGO COMPUTATIONS; 0 = VG UPDATE ONLY
#       FIRSTFLG        1 = GONE TO LAMBERT AT LEAST ONCE; 0 = HAVEN'T GONE TO LAMBERT YET.
#
# NOTE:  VGTIG EQUALS VGPREV
#
# OUTPUT
#       STEERSW         SEE INPUT
#       INPULSW         1 = ENGINE OFF IN TGO CENTISECONDS; 0 = CONTINUE BURN
#       TGO             TIME TO CUT-OFF 2(28) CS
#       SEE FINDCDUW FOR STEERING OUTPUTS.
#
# SUBROUTINE CALLED
#       FINDCDUW
#
# DEBRIS
#       MPACS, PUSHLIST

                COUNT*  $$/S40.8
```

```
S40.8           BOF                             # GENERATE VR IF NOT EXTERNAL DELTA-V BURN
                        XDELVFLG
                        RASTEER1
                VLOAD   VSU
                        VGPREV
                        DELVREF
VGAIN*          STORE   VG                      # VELOCITY TO BE GAINED SCALED AT (7) M/CS
                MXV     VSL1
                        REFSMMAT
                STORE   UNFC/2
BDTOK           VLOAD   ABVAL
                        VG
                STORE   VGDISP
TGDCALC         SETPD   VLOAD
                        0
                        VG
                STOVL   VGPREV
                        DELVREF
                BOFF    VCOMP
                        STEERSW
                        QPRET
                UNIT
                DOT     PUSH
                        VG
                BPL     DDV
                        ALARMIT         # DELV IS MORE THAN 90 DEGREES FROM VG.
                        VEX
                DAD     DMP
                        DPHALF
                SR      DDV
                        10D
                        36D
                DMP     DAD
                        -FOURDT
                        TDECAY
                STORE   TGO
                DAD
                        PIPTIME
                STODL   TIG
                        TGO
                DSU     BPL
                        FOURSECS        # 400 CS
                        FINDCDUW -2
                SET     CLRGO
                        IMPULSW
                        STEERSW
                        QPRET

ALARMIT         EXIT
```

```
                TC      ALARM
                OCT     01407
                TC      INTPRET
                GOTO                    # SKIP TGO COMPUTATION BUT CALL FINDCDUW.
                        FINDCDUW -2     # FINDCDUW WILL EXIT TO UPDATEVG +3.

-FOURDT         2DEC    -800 B-18       # -4 (200 CS.) B(-18)
FOURSECS        2DEC    400             # 400 CS SCALED AT 2(+28) CS
2VEXHUST        =       VEX
```

```
# NAME:                 S40.13 -- TIMEBURN
#
# FUNCTION              (1) DETERMINE WHETHER A GIVEN COMBINATION OF VELOCITY TO
#                       BE GAINED AND ENGINE CHOICE RESULT IN A BURN TIME
#                       SUFFICIENT TO ALLOW STEERING AT THE VEHICLE DURING THE
#                       BURN
#                       (2) THE MAGNITUDE OF THE RESULTING BURN TIME -- IF IT
#                       IS SHORT -- AND THE ASSOCIATED TIME OF THE ENGINE OFF
#                       SIGNAL
#
# CALLING SEQUENCE      VIA FINDVAC AS A NEW JOB
#
# INPUT                 VGTIG -- VELOCITY TO BE GAINED VECTOR (METERS/CS) AT +7
#                       WEIGHT/G -- MASS OF VEHICLE IN KGM AT +16
#                       F -- APS ENGINE THRUST IN M.NEWTONS AT +7
#                               AND ALSO FOR RCS ENGINE
#                       MDOT -- RATE OF DECREASE OF VEHICLE MASS DURING ENGINE
#                               BURN IN KILOGRAMS/CS AT +3.  THIS SCALING MAY
#                               REQUIRE MODIFICATION FOR SATURN BURNS.
#                       ENG1FLAG -- SWITCH TO DECIDE WHETHER APS OR DPS ENGINE IS USED
#                               =0      DPS
#                               =1      APS
#
# OUTPUT                IMPULSW         ZERO FOR STEERING
#                                       ONE FOR ATTITUDE HOLD
#                       NOTHROTL        ZERO FOR THROTTLING
#                                       ONE TO INHIBIT THROTTLING
#                       TGO             TIME TO BURN IN CS
#
#                       THE QUANTITY M.NEWTON = 10000 NEWTONS WILL BE USED TO EXPRESS
#                       FORCE.

                EBANK=  TGO
                COUNT*  $$/40.13
S40.13          TC      INTPRET
                SETPD   CLEAR
                        OOD
                        IMPULSW         # ASSUME NO STEERING UNTIL FOUND OTHERWISE
                VLOAD   ABVAL
                        VGTIG           # VELOCITY TO BE GAINED AT +7
                PDDL    DMP             # OOD = MAG OF VGTIG AT +7
                        4SEC(17)        # CORRECT VG FOR 4 SECS OF 2 JET ULLAGE
                        FRCS2
                DDV     SL1             # SCALE
                        WEIGHT/G
                BDSU    PUSH
                BOFF    SET
                        APSFLAG
                        S40.13D         # FOR DPS ENGINE
                        NOTHROTL
                DLOAD   DDV             # OOD = MAG OF VGTIG CORRECTED
                        K1VAL           # M.NEWTONS-CS AT +24
                        WEIGHT/G
                BDSU    BMN
```

```
                                OOD
                                S40.131         # TGO LESS THAN 100 CS
                        PDDL    DMP             # 02D = TEMP1 AT +7
                                MDOT

# MDOT REPRESENTS THE RATE OF DECREASE OF VEHICLE MASS DURING ENGINE
# BURN IN KILOGRAMS/CS.  WHEN SATURN IS USED, THE SCALING MAY
# REQUIRE ADJUSTMENT.

                                3.5SEC          # 350 CS AT +14
                        BDSU    PDDL
                                WEIGHT/G
                                F
                        DMP     SR2             # SCALE
                                5SECS
                        DDV     PUSH            # 04D = TEMP2
                        BDSU    BPL
                                02D
                                S40.13D
                        DLOAD   BDDV
                        DMP     DAD
                                5SECS
                                1SEC2D          # 100 CS AT +14
                        GOTO
                                S40.132
S40.131                 DLOAD   DMP
                                WEIGHT/G
                        SR1     PUSH
                        DAD     DDV
                                K2VAL           # M.NEWTON CS AT +24
                                K3VAL           # M.NEWTON CS AT +10
S40.132                 SET     EXIT
                                IMPULSW
S40.132*                TC      TPAGREE
                        CA      MPAC
                        XCH     L
                        CA      ZERO
                        DXCH    TGO
                        TCF     S40.134

S40.13D                 DLOAD   DMP             # FOR DPS ENGINE
                                OOD
                                WEIGHT/G
                        PUSH    BON
                                APSFLAG
                                APSTGO
                        DDV     CLEAR
                                S40.136
                                NOTHROTL
                        BOV     PUSH
```

```
                                S40.130V
S40.127         DSU             BPL
                                6SEC            # 600.0 CS AT +14
                                S40.138
                DAD             GOTO
                                6SEC
                                S40.132
S40.133         EXIT
S40.134         TC              PHASCHNG
                OCT             00003
                TC              ENDOFJOB
S40.130V        DLOAD           SR4             # RECOMPUTED TGO IN TIMER UNITS
                DDV
                                S40.136_        # S40.136 SHIFTED LEFT 10
                STORE           TGO
                EXIT
                TCF             S40.134         # REJOIN COMMON CODING FOR RESTART PROTECT

S40.138         DSU             BPL
                                89SECS
                                STORETGO
                SET
                                NOTHROTL
STORETGO        DLOAD                           # LOAD TGO AT 2(14)
                EXIT
                TCF             S40.132*

APSTGO          DDV             SL2
                                FAPS
                GOTO
                                STORETGO +1
1SEC2D          2DEC            100.0 B-14      # 100.0 CS AT +14

3.5SEC          2DEC            350.0 B-13      # 350 CS AT +13

5SECS           2DEC            500.0 B-14      # 500.0 CS AT +14

6SEC            2DEC            600.0 B-14      # 600.0 CS AT +14

89SECS          2DEC            8900.0 B-14

# FUNCTION                      (1) GENERATES REQUIRED VELOCITY AND VELOCITY-TO-BE-GAINED
#                               VECTORS FOR USE DURING AIMPOINT MANEUVERS EVERY TWO
#                               COMPUTATION CYCLES (4 SECONDS).
#                               (2) UPDATES THE B VECTOR WHICH IS USED IN THE FINAL
#                               CALCULATION OF EXTRAPOLATING THE VELOCITY-TO-BE-GAINED
#                               THROUGH ONE 2-SECOND INTERVAL INTO THE FUTURE.
#
# CALLING SEQUENCE              VIA FINDVAC AS NEW JOB.
#
# INPUT                         RN      ACTIVE VEHICLE RADIUS VECTOR IN METERS AT +29
#                               VN      ACTIVE VEHICLE VELOCITY VECTOR IN METERS/CS AT +7
```

```
#                       VPREV   LAST COMPUTED VELOCITY REQUIRED VECTOR IN
#                               METERS/CS AT +7.
#                       TIG     TIME OF IGNITION IN CS AT +28.
#                       DLTARG  COMPUTATION CYCLE INTERVAL = 200 CS AT +28.
#                       PIPTIME TIME OF RN AND VN IN CS AT +28.
#                       GDT/2   HALF OF VELOCITY GAINED IN DELTA T TIME DUE TO
#                               ACCELERATION OF GRAVITY IN METERS/CS AT +7.
#                       DELVREF CHANGE IN VELOCITY DURING LAST 2 SEC IN
#                               METERS/CS AT +7.
#
# OUTPUT                VGPREV  VELOCITY TO BE GAINED VECTOR IN METERS/CS AT +7.
#                       VGDISP  MAG OF VGPREV FOR DISPLAY PURPOSES.
#                       VRPREV  VELOCTY REQUIRED VECTOR IN METERS/CS AT +7.
#                       BDT     B VECTOR IN METERS/CS AT +7.
#
# SUBROUTINES USED      INITVEL

                EBANK=  VGPREV
                COUNT*  $$/S40.9
S40.9           TC      INTPRET
                SETPD
                        00D
                SET     DLOAD
                        AVFLAG          # SET AVFLAG FOR LEM ACTIVE
                        HI6ZEROS
                PDDL
                        EPS1
                BOFF    DAD             # EPSILON4 = 10 OR 45 DEGREES.
                        NORMSW
                        EPSSMALL
                        EPS2
EPSSMALL        PUSH    CALL
                        HAVEGUES
ENDS40.9        EXIT
                TC      PHASCHNG
                OCT     2
                TCF     ENDOFJOB

RASTEER1        VLOAD   ABVAL
                        RN
                LXC,2   SL*
                        RTX2
                        0,2
                STOVL   RMAG
                        RTARG
                VSU     RTB
                        RN
                        NORMUNX1
                STODL   IC
                        36D             # C(36D) = ABVAL(C)
                XAD,2   SL*
                        X1
```

```
                0,2
        STORE   30D
        NORM    DMP
                X2
                RMAG
        NORM    XAD,2
                X1
                X1
        SXA,2
                MUSCALE
        STODL   R1C             #                       2(+56 -X)
                30D
        SR1     PDDL
                RMAG
        SR1     PDDL
                RTMAG
        SR1     DAD
        DAD     STADR
        STORE   SS              # SS = (R1 + R2 + C)/2
        DSU     DMP
                30D
                MU/A
        BDSU
                MUASTEER
        PDDL    DSU
                SS
                RMAG
        NORM    SR1
                X1
        DDV     DMP
                R1C
        XSU,2   SL*
                X1
                1,2
        LXA,2
                MUSCALE
        SQRT    SIGN
                GEOMSGN
        STORE   32D             # + OR - A
        DLOAD   DMP
                SS
                MU/A
        BDSU
                MUASTEER
        PDDL    DSU
                SS
                RTMAG
        NORM    SR1
                X1
        DDV     DMP
```

```
                        R1C
          XSU,2         SL*
                        X1
                        1,2
          SQRT          PDDL            # -B (NO SIGN)
                        SS
          DSU           DDV
                        30D
                        SS
          SQRT          PUSH
          SR1           ASIN
          DMP           PDDL
                        2PI+3
          PDDL          DDV
                        30D
                        SS
          BOV
                        +1
          SQRT          DMP
          SR3           BDSU
          SIGN          PDDL
                        GEOMSGN
                        2PI+3
          SR2           DSU
          DMP           PDDL
                        SS
                        SS
          SR3           SQRT
          DMP
          PDDL          SL3
                        MUASTEER
          SQRT          BDDV
          DSU           DAD
                        TPASS4
                        PIPTIME
          STODL         30D
          SIGN
                        30D             # B WITH SIGN
          STORE         30D
          BON           VLOAD
                        NORMSW
                        180MESS
                        IC
          VSU           UNIT
                        UNIT/R/
          VXSC          PDVL
                        30D
                        IC
          VAD           UNIT
                        UNIT/R/
```

```
GETVRVG1        VXSC    VAD
                        32D
GETVRVG2        LXC,2   VSR*
                        RTX2
                        0       -1,2
                STORE   VIPRIME
                GOTO
                        ASTREND -2
18OMESS         VLOAD   DOT
                        IC
                        UNIT/R/
                BMN     VLOAD
                        NEGPROD
                        IC
                VSR1    PDVL
                        UNIT/R/
                VSR1    VAD
                UNIT
                PUSH    VCOMP           # FOR A
                VXV     SIGN
                        UN
                        GEOMSGN
                UNIT    VXSC
                        30D
                PDVL                    # UNIT(IC-IR)   +-B
                GOTO
                        GETVRVG1
NEGPROD         VLOAD   VSR1
                        UNIT/R/
                PDVL    VSR1
                        IC
                VSU     UNIT
                PUSH
                VXV     SIGN
                        UN              # FOR B
                        GEOMSGN
                UNIT    VXSC
                        32D
                PDVL
                VXSC    VAD
                        30D
                GOTO
                        GETVRVG2
                VSU
                        VN1
ASTREND         STORE   DELVEET3
FIRSTTME        SLOAD   BZE
                        RTX2
                        GETGOBL
                VLOAD   GOTO            # NO OBLATENESS COMP IF IN MOON SPHERE
```

```
                        DELVEET3
                        NOGOBL
GETGOBL         VLOAD   UNIT            # CALCULATE OBLATENESS TERM.
                        RN
                DLOAD   DSU
                        PIPTIME         #              2
                        GOBLTIME        # G    = -(MU/R )(UNITGOBL)(T-TIG)
                DMP     DDV             #  OBL
                        EARTHMU
                        34D             # 34D = /RN/ (2) FROM UNIT OPERATION.
                VXSC    VAD
                        UNITGOBL
                        DELVEET3        # OUTPUT FROM INITVEL VG = VR - VN
NOGOBL          STORE   DELVEET3        # VG = VR + GOBL - VN
                GOTO
                        VGAIN*

2PI+3           2DEC    3.141592653 B-2
```

```
# TRIMGIMB      (FORMERLY S40.6)
# MOD 0         24 FEB 67       PETER ADLER
#
# FUNCTION:
#       TRIMS DPS ENGINE TO MINIMIZE THRUST/CG OFFSET.  ENGINE IS GIMBALLED TO FULL + PITCH AND + ROLL (TO LOCK)
#       FOR REFERENCE AND IS THEN BROUGHT BACK TO TRIM POSITION BY RUNNING FOR THE PROPER TIMES (TO BE
#       SPECIFIED BY GAEC) IN - PITCH AND - ROLL.
#
# CALLING SEQUENCE:
#       VIA WAITLIST FROM R03
#
# INPUT:
#       PITTIME         TIME TO RUN FROM FULL + PITCH TO TRIM (CS)
#       ROLLTIME        TIME TO RUN FROM FULL + ROLL TO TRIM (CS)
#
# SUBROUTINES USED:
#       WAITLIST, FIXDELAY, VARDELAY, FLAGUP, FLAGDOWN, NOVAC

                COUNT*  $$/S40.6
                EBANK=  ROLLTIME        # OCTAL MASKS: PRIO5=05000 EBANK5=02400

TRIMGIMB        TC      DOWNFLAG        # GMBDRVSW FLAG IS SET WHEN EITHER ROLL OR
                ADRES   GMBDRVSW        # PITCH IS COMPLETED, WHICHEVER IS FIRST.

                CS      PRIO5           # TURN OFF - PITCH, - ROLL, IF ON.
                EXTEND
                WAND    CHAN12
                CAF     EBANK5          # TURN ON + PITCH, + ROLL.
                EXTEND
                WOR     CHAN12
                TC      FIXDELAY        # WAIT ONE MINUTE TO MAKE SURE ENGINE IS
                DEC     6000            # AT FULL + PITCH AND FULL + ROLL
                CS      EBANK5          # TURN OFF + PITCH, + ROLL.
                EXTEND
                WAND    CHAN12
                CAF     PRIO5           # TURN ON - PITCH, - ROLL.
                EXTEND
                WOR     CHAN12
                CAE     PITTIME         # GET TIME TO SHUT OFF - PITCH AND SET UP
                TC      TWIDDLE         # TWIDDLE-TASK TO TURN IT OFF THEN
                ADRES   PITCHOFF

                CAE     ROLLTIME        # GET TIME TO SHUT OFF - ROLL AND GO AWAY
                TC      VARDELAY        # UNTIL THEN
                CS      BIT12
                EXTEND
                WAND    CHAN12          # SHUT OFF ROLL
ROLLOVER        CA      FLAGWRD6        # IF HERE INLINE (ROLL DONE) IS PITCH DONE
                MASK    GMBDRBIT        # IF HERE FROM PITCHOFF, IS ROLL DONE?
                EXTEND
                BZF     PITCHOFF +4     # NO.  SET FLAG, ROLL OR PITCH DONE.
                CAF     PRIO10          # RETURN TO R03.
                TC      NOVAC
                EBANK=  WHOCARES
```

```
                2CADR   TRIMDONE

                TC      TASKOVER
PITCHOFF        CS      BIT10
                EXTEND
                WAND    CHAN12          # SHUT OFF PITCH
                TCF     ROLLOVER        # SEE IF ROLL HAS FINISHED ALSO.
                TC      UPFLAG          # ROLL DONE; OR PITCH DONE; BUT NOT BOTH.
                ADRES   GMBDRVSW
                TC      TASKOVER
```

```
# SUBROUTINE NAME:  S41.1       MOD. NO. 0      DATE: FEBRUARY 28, 1967
# MOD. NO. 1    DATE: JANUARY 23, 1968, BY PETER ADLER (MIT/IL)
#
# AUTHOR: JONATHON D. ADDLESTON (ADAMS ASSOCIATES)
#
# S41.1 PERFORMS THE COORDINATE SYSTEM TRANSFORMATION FROM THE REFERENCE FRAME TO THE BODY OF THE LM.
# SPECIFICALLY, IT IS USED TO TRANSFORM A VELOCITY (SCALED AT 2(+7) METERS/CENTISECOND) FROM REFERENCE TO LM AXIS
# COORDINATES.  FIRST THE VECTOR IS TRANSFORMED TO THE STABLE MEMBER COORDINATES BY THE MATRIX REFSMMAT.  THIS
# LEAVES THE VECTOR IN MPAC, SCALED AT 2(+8) METERS/CENTISECOND.  THEN
# THE SUBROUTINE CDUTRIG IS CALLED TO SET UP THE DOUBLE-PRECISION CDU VECTOR ALONG WITH ITS SINES AND COSINES.
# THE VECTOR IS THEN TRANSFORMED FROM STABLE MEMBER COORDINATES TO SPACECRAFT (OR LM) COORDINATES BY THE
# SUBROUTINE *SMNB*.  FINALLY, THE VECTOR IS RESCALED TO 2(+7) METERS/CENTISECOND, AND CONTROL IS RETURNED BO THE
# CALLER WITH C(MPAC) = VELOCITY(LM).
#
# CALLING SEQUENCE:
#       L       VLOAD   CALL
#       L +1            VELOCITY(REF)           # SCALED AT 2(+7) M/CS IN REFERENCE COORDS.
#       L +2            S41.1
#       L +3    STORE   VELOCITY(LM)            # SCALED AT 2(+7) M/CS IN LM BODY AXIS SYS.
#
# SUBROUTINES CALLED:
#       1.      CDUTRIG,
#                       WHICH CALLS CDULOGIC.
#       2.      *SMNB*
#
# NORMAL RETURN:  L +3 (SEE CALLING SEQUENCE, ABOVE.)
#
# ALARM/ABORT MODES:  NONE.
#
# RESTART PROTECTION:  NONE.
#
```

```
# INPUT:
#       1.      REFSMMAT.
#       2.      CDUX, CDUY, CDUZ.
#       3.      VELOCITY (REF) IN MPAC.
#
# OUTPUT:
#       1.      CSUSPOT:        DOUBLE PRECISION CDU VECTOR, ORDERED Y,Z,X.
#       2.      SINCDU:         HALF SINES OF CDUSPOT COMPONENTS
#       3.      COSCDU:         HALF COSINES OF CDUSPOT COMPONENTS.
#       4.      MPAC:           VELOCITY(LM) (SCALED AT 2(+7) METERS/CENTISECOND)
#
# DEBRIS:  NONE.
#
# CHECKOUT STATUS:  CODED

                COUNT*  $$/S41.1
S41.1           MXV     VSL1            # CONVERT VECTOR IN MPAC FROM REF AT 2(+7)
                        REFSMMAT        # TO SM AND RESCALE DUE TO HALF-UNIT MATRIX
                GOTO                    # CONVERT TO BODY AT 2(+7) USING PRESENT
                        CDU*SMNB        # CDU ANGLES.  CDU*SMNB WILL RETURN
                                        # VIA RVQ TO THE CALLER OF S41.1.
```

```
                BANK    32
                SETLOC  F2DPS*32
                BANK

                EBANK=  E2DPS

#       ************************************
#       P63: THE LUNAR LANDING, BRAKING PHASE
#       ************************************

                COUNT*  $$/P63

P63LM           TC      PHASCHNG
                OCT     04024

                TC      BANKCALL        # DO IMU STATUS CHECK ROUTINE R02
                CADR    R02BOTH

                CAF     P63ADRES        # INITIALIZE WHICH FOR BURNBABY
                TS      WHICH

                CAF     DPSTHRSH        # INITIALIZE DVMON
                TS      DVTHRUSH
                CAF     FOUR
                TS      DVCNTR

                CS      ONE             # INITIALIZE WCHPHASE AND FLPASSO
                TS      WCHPHASE

                CA      ZERO
                TS      FLPASSO

                CS      BIT14
                EXTEND
                WAND    CHAN12          # REMOVE TRACK-ENABLE DISCRETE.

FLAGORGY        TC      INTPRET         # DIONYSIAN FLAG WAVING
                CLEAR   CLEAR
                        NOTHROTL
                        REDFLAG
                CLEAR   SET
                        LRBYPASS
                        MUNFLAG
                CLEAR   CLEAR
                        P25FLAG         # TERMINATE P25 IF IT IS RUNNING.
                        RNDVZFLG        # TERMINATE P20 IF IT IS RUNNING

                                        # ********************************

IGNALG          SETPD   VLOAD           # FIRST SET UP INPUTS FOR RP-TO-R:-
```

```
                                O               #   AT OD LANDING SITE IN MOON FIXED FRAME
                                RLS             #   AT 6D ESTIMATED TIME OF LANDING
                        PDDL    PUSH            #   MPAC NON-ZERO TO INDICATE LUNAR CASE
                                TLAND
                        STCALL  TPIP            # ALSO SET TPIP FOR FIRST GUIDANCE PASS
                                RP-TO-R
                        VSL4    MXV
                                REFSMMAT
                        STCALL  LAND
                                GUIDINIT        # GUIDINIT INITIALIZES WM AND /LAND/
                        DLOAD   DSU
                                TLAND
                                GUIDDURN
                        STCALL  TDEC1           # INTEGRATE STATE FORWARD TO THAT TIME
                                LEMPREC
                        SSP     VLOAD
                                NIGNLOOP
                                40D
                                UNITX
                        STOVL   CG
                                UNITY
                        STOVL   CG +6
                                UNITZ
                        STODL   CG +14
                                99999CON
                        STOVL   DELTAH          # INITIALIZE DELTAH FOR V16N68 DISPLAY
                                ZEROVECS
                        STODL   UNFC/2          # INITIALIZE TRIM VELOCITY CORRECTION TERM
                                HI6ZEROS
                        STORE   TTF/8

        IGNALOOP        DLOAD
                                TAT
                        STOVL   PIPTIME1
                                RATT1
                        VSL4    MXV
                                REFSMMAT
                        STCALL  R
                                MUNGRAV
                        STCALL  GDT/2
                                ?GUIDSUB        # WHICH DELIVERS N PASSES OF GUIDANCE

        # DDUMCALC IS PROGRAMMED AS FOLLOWS:-
        #                                             2                                  ___
        #              (RIGNZ - RGU )/16 + 16(RGU   )KIGNY/B8 + (RGU - RIGNX)KIGNX/B4 + (ABVAL(VGU) - VIGN)KIGNV/B4
        #                          2              1                  0
        #       DDUM = -----------------------------------------------------------------------------------------------
        #                                                    10
        #                                                   2    (VGU - 16 VGU KIGNX/B4)
        #                                                            2         0
```

```
# DISCONNECTED FROM THEIR RESPECTIVE VARIABLES
# THE NUMERATOR IS SCALED IN METERS AT 2(28).  THE DENOMINATOR IS A VELOCITY IN UNITS OF 2(10) M/CS.
# THE QUOTIENT IS THUS A TIME IN UNITS OF 2(18) CENTISECONDS.  THE FINAL SHIFT RESCALES TO UNITS OF 2(28) CS.
# THERE IS NO DAMPING FACTOR.  THE CONSTANTS KIGNX/B4, KIGNY/B8 AND KIGNV/B4 ARE ALL NEGATIVE IN SIGN.

DDUMCALC        TS      NIGNLOOP
                TC      INTPRET
                DLOAD   DMPR            # FORM DENOMINATOR FIRST
                        VGU
                        KIGNX/B4
                SL4R    BDSU
                        VGU +4
                PDDL    DSU
                        RIGNZ
                        RGU +4
                SR4R    PDDL
                        RGU +2
                DSQ     DMPR
                        KIGNY/B8
                SL4R    PDDL
                        RGU
                DSU     DMPR
                        RIGNX
                        KIGNX/B4
                PDVL    ABVAL
                        VGU
                DSU     DMPR
                        VIGN
                        KIGNV/B4
                DAD     DAD
                DAD     DDV
                SRR
                        10D

                PUSH    DAD
                        PIPTIME1
                STODL   TDEC1           # STORE NEW GUESS FOR NEXT INTEGRATION
                ABS     DSU
                        DDUMCRIT
                BMN     CALL
                        DDUMGOOD
                        INTSTALL
                SET     SET
                        INTYPFLG
                        MOONFLAG
                DLOAD

                        PIPTIME1
                STOVL   TET             # HOPEFULLY ?GUIDSUB DID NOT
                        RATT1           #   CLOBBER RATT1 AND VATT1
```

```
                STOVL   RCV
                        VATT1
                STCALL  VCV
                        INTEGRVS
                GOTO
                        IGNALOOP

DDUMGOOD        SLOAD   SR
                        ZOOMTIME
                        14D
                BDSU
                        TDEC1
                STOVL   TIG             # COMPUTE DISTANCE LANDING SITE WILL BE
                        V               #   OUT OF LM'S ORBITAL PLANE AT IGNITION:
                VXV     UNIT            #   SIGN IS + IF LANDING SITE IS TO THE
                        R               #   RIGHT, NORTH; - IF TO THE LEFT, SOUTH.
                DOT     SL1
                        LAND
R60INIT         STOVL   OUTOFPLN        # INITIALIZATION FOR CALCMANU
                        UNFC/2
                STORE   R60VSAVE        # STORE UNFC/2 TEMPORARILY IN R60SAVE
                EXIT
                                        # ********************************************

IGNALGRT        TC      PHASCHNG        # PREVENT REPEATING IGNALG
                OCT     04024

ASTNCLOK        CS      ASTNDEX
                TC      BANKCALL
                CADR    STCLOK2
                TCF     ENDOFJOB        # RETURN IN NEW JOB AND IN EBANK FIVE

ASTNRET         TC      INTPRET
                SSP     RTB             # GO PICK UP DISPLAY AT END OF R51:
                        QMAJ            #   "PROCEED" WILL DO A FINE ALIGNMENT
                FCADR   P63SPOT2        #   " ENTER " WILL RETURN TO P63SPOT2
                        R51P63
P63SPOT2        VLOAD   UNIT            # INITIALIZE KALCMANU FOR BURN ATTITUDE
                        R60VSAVE
                STOVL   POINTVSM
                        UNITX
                STORE   SCAXIS
                EXIT

                CAF     EBANK7
                TS      EBANK

                INHINT
                TC      IBNKCALL
                CADR    PFLITEDB
```

```
                RELINT

                TC      BANKCALL
                CADR    R60LEM

                TC      PHASCHNG        # PREVENT RECALLING R60
                OCT     04024

P63SPOT3        CA      BIT6            # IS THE LR ANTENNA IN POSITION 1 YET
                EXTEND
                RAND    CHAN33
                EXTEND
                BZF     P63SPOT4        # BRANCH IF ANTENNA ALREADY IN POSITION 1

                CAF     CODE500         # ASTRONAUT:    PLEASE CRANK THE
                TC      BANKCALL        #               SILLY THING AROUND
                CADR    GOPERF1
                TCF     GOTOPOOH        # TERMINATE
                TCF     P63SPOT3        # PROCEED       SEE IF HE'S LYING

P63SPOT4        TC      BANKCALL        # ENTER         INITIALIZE LANDING RADAR
                CADR    SETPOS1

                TC      POSTJUMP        # OFF TO SEE THE WIZARD...
                CADR    BURNBABY

#       ---------------------------------

#       CONSTANTS FOR P63LM AND IGNALG

P63ADRES        GENADR  P63TABLE

ASTNDEX         =       MD1             # OCT 25:  INDEX FOR CLOKTASK

CODE500         OCT     00500

99999CON        2DEC    30479.7 B-24

GUIDDURN        2DEC    +66440          # GUIDDURN      +6.64400314 E+2
DDUMCRIT        2DEC    +8 B-28         # CRITERION FOR IGNALG CONVERGENCE
```

```
#       --------------------------------
```

```
#       ************************
#       P68: LANDING CONFIRMATION
#       ************************

                BANK    31
                SETLOC  F2DPS*31
                BANK

                COUNT*  $$/P6567

LANDJUNK        TC      PHASCHNG
                OCT     04024

                INHINT
                TC      BANKCALL        # ZERO ATTITUDE ERROR
                CADR    ZATTEROR

                TC      BANKCALL        # SET 5 DEGREE DEADBAND
                CADR    SETMAXDB

                TC      INTPRET         # TO INTERPRETIVE AS TIME IS NOT CRITICAL
                SET     CLEAR
                        SURFFLAG
                        LETABORT
                SET     VLOAD
                        APSFLAG
                        RN
                STODL   ALPHAV
                        PIPTIME
                SET     CALL
                        LUNAFLAG
                        LAT-LONG
                SETPD   VLOAD           # COMPUTE RLS AND STORE IT AWAY
                        0
                        RN
                VSL2    PDDL
                        PIPTIME
                PUSH    CALL
                        R-TO-RP
                STORE   RLS
                EXIT
                CAF     V06N43*         # ASTRONAUT:  NOW LOOK WHERE YOU ENDED UP
                TC      BANKCALL
                CADR    GOFLASH
                TCF     GOTOPOOH        # TERMINATE
                TCF     +2              # PROCEED
                TCF     -5              # RECYCLE

                TC      INTPRET
```

```
                VLOAD                   # INITIALIZE GSAV AND (USING REFMF)
                        UNITX           #   YNBSAV, ZNBSAV AND ATTFLAG FOR P57
                STCALL  GSAV
                        REFMF
                EXIT

                TCF     GOTOPOOH        # ASTRONAUT:  PLEASE SELECT P57

V06N43*         VN      0643
```

```
                BANK    31
                SETLOC  FTHROT
                BANK
                EBANK=  PIF
                COUNT*  $$/THROT

# * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
# HERE FC, DESIRED THRUST, AND FP, PRESENT THRUST, UNWEIGHTED, ARE COMPUTED.

THROTTLE        CA      ABDELV          # COMPUTE PRESENT ACCELERATION IN UNITS OF
                EXTEND                  # 2(-4) M/CS/CS, SAVING SERVICER TROUBLE
                MP      /AF/CNST
 +3             EXTEND
                QXCH    RTNHOLD
AFDUMP          TC      MASSMULT
                DXCH    FP              # FP = PRESENT THRUST
                EXTEND
                DCA     /AFC/
                TC      MASSMULT
                TS      FC              # FC = THRUST DESIRED BY GUIDANCE
                DXCH    FCODD           # FCODD = WHAT IT IS GOING TO GET

# IF IT HAS BEEN LESS THAN 3 SECONDS SINCE THE LAST THROTTLING, AUGMENT FP USING THE FWEIGHT CALCULATED THEN.

                CS      TTHROT          # THIS CODING ASSUMES A FLATOUT WITHIN
                AD      TIME1           #   80 SECONDS BEFORE FIRST THROTTLE CALL
                MASK    POSMAX
                COM
                AD      3SECS
                EXTEND
                BZMF    WHERETO         # BRANCH IF (TIME1-TTHROT +1) > 3 SECONDS
                EXTEND
                DCA     FWEIGHT
                DAS     FP

# THIS LOGIC DETERMINES THE THROTTLING IN THE REGION 10% - 94%.  THE MANUAL THROTTLE, NOMINALLY SET AT
# MINIMUM BY ASTRONAUT OR MISSION CONTROL PROGRAMS, PROVIDES THE LOWER BOUND.  A STOP IN THE THROTTLE HARDWARE
# PROVIDES THE UPPER.

WHERETO         CA      EBANK5          # INITIALIZE L*WCR*T AND H*GHCR*T FROM
                TS      EBANK           #   PAD LOADED ERASABLES IN W-MATRIX
```

```
                EBANK=  LOWCRIT
                EXTEND
                DCA     LOWCRIT
                DXCH    L*WCR*T
                CA      EBANK7
                TS      EBANK
                EBANK=  PIF
                CS      ZERO            # INITIALIZE PIFPSET
                TS      PIFPSET
                CS      H*GHCR*T
                AD      FCOLD
                EXTEND
                BZMF    LOWFCOLD        # BRANCH IF FCOLD < OR = HIGHCRIT
                CS      L*WCR*T
                AD      FCODD
                EXTEND
                BZMF    FCOMPSET        # BRANCH IF FC < OR = LOWCRIT
                CA      FP              # SEE NOTE 1
                TCF     FLATOUT1

FCOMPSET        CS      FMAXODD         # SEE NOTE 2
                AD      FP
                TCF     FLATOUT2

LOWFCOLD        CS      H*GHCR*T
                AD      FCODD
                EXTEND
                BZMF    DOPIF           # BRANCH IF FC < OR = HIGHCRIT

                CA      FMAXPOS         # NO:  THROTTLE-UP
FLATOUT1        DXCH    FCODD
                CA      FEXTRA
FLATOUT2        TS      PIFPSET

# NOTE 1        FC IS SET EQUAL TO FP SO PIF WILL BE ZERO.  THIS IS DESIRABLE
#               AS THERE IS ACTUALLY NO THROTTLE CHANGE.
#
# NOTE 2        HERE, SINCE WE ARE ABOUT TO RETURN TO THE THROTTLEABLE REGION
#               (BELOW 55%) THE QUANTITY -(FMAXODD - FP) IS COMPUTED AND PUT
#               INTO PIFPSET TO COMPENSATE FOR THE DIFFERENCE BETWEEN THE
#               NUMBER OF BITS CORRESPONDING TO FULL THROTTLE (FMAXODD) AND THE
#               NUMBER CORRESPONDING TO ACTUAL THRUST (FP).  THUS THE TOTAL
#               THROTTLE COMMAND PIF = FC - FP - (FMAXODD - FP) = FC - FMAXODD.

DOPIF           TC      FASTCHNG
                EXTEND
                DCA     FCODD
                TS      FCOLD
                DXCH    PIF
                EXTEND
```

```
                DCS     FP
                DAS     PIF             # PIF = FC - FP, NEVER EQUALS +0

DOIT            CA      PIF
                AD      PIFPSET         # ADD IN PIFPSET, WITHOUT CHANGING PIF
                TS      PSEUDO55
                TS      THRUST
                CAF     BIT4
                EXTEND
                WOR     CHAN14
                CA      TIME1
                TS      TTHROT

# SINCE /AF/ IS NOT AN INSTANTANEOUS ACCELERATION, BUT RATHER AN "AVERAGE" OF THE ACCELERATION LEVELS DURING
# THE PRECEEDING PIPA INTERVAL, AND SINCE FP IS COMPUTED DIRECTLY FROM /AF/, FP IN ORDER TO CORRESPOND TO THE
# ACTUAL THRUST LEVEL AT THE END OF THE INTERVAL MUST BE WEIGHTED BY
#
#                    PIF(PPROCESS + TL)      PIF /PIF/
#       FWEIGHT = ------------------ + -------------
#                         PGUID            2 PGUID FRATE
#
# WHERE PPROCESS IS THE TIME BETWEEN PIPA READING AND THE START OF THROTTLING, PGUID IS THE GUIDANCE PERIOD, AND
# FRATE IS THE THROTTLING RATE (32 UNITS PER CENTISECOND).  PGUID IS EITHER 1 OR 2 SECONDS.  THE "TL" IN THE
# FIRST TERM REPRESENTS THE ENGINE'S RESPONSE LAG.  HERE FWEIGHT IS COMPUTED FOR USE NEXT PASS.

                CA      THISTPIP +1             # INITIALIZE FWEIGHT COMP AS IF FOR P66
                TS      BUF

                CS      MODREG                  # ARE WE IN FACT IN P66?
                AD      DEC66
                EXTEND
                BZF     FWCOMP                  # YES

                CA      PIPTIME +1              # NO:  INITIALIZE FOR TWO SECOND PERIOD
                TS      BUF
                CAF     4SECS
                TCF     FWCOMP +1

FWCOMP          CAF     2SECS
 +1             TS      Q
                EXTEND
                MP      BIT6
                LXCH    BUF +1
                CS      BUF             # TIME OF LAST PIPA READING.
                AD      TIME1
                AD      THROTLAG        # COMPENSATE FOR ENGINE RESPONSE LAG
                MASK    LOW8            # MAKE SURE SMALL AND POSITIVE
                ZL
                EXTEND
```

```
                DV      Q
                EXTEND
                MP      PIF
                DOUBLE
                DXCH    FWEIGHT
                CCS     PIF
                AD      ONE
                TCF     +2
                AD      ONE
                EXTEND
                MP      PIF
                EXTEND
                DV      BUF +1
                ZL
                DAS     FWEIGHT

THDUMP          TC      RTNHOLD

# FLATOUT THROTTLES UP THE DESCENT ENGINE, AND IS CALLED AS A BASIC SUBROUTINE.

FLATOUT         CAF     BIT13           # 4096 PULSES
WHATOUT         TS      PIFPSET         # USE PIFPSET SO FWEIGHT WILL BE ZERO
                CS      ZERO
                TS      FCOLD
                TS      PIF
                EXTEND
                QXCH    RTNHOLD
                TCF     DOIT

# MASSMULT SCALES ACCELERATION, ARRIVING IN A AND L IN UNITS OF 2(-4) M/CS/CS, TO FORCE IN PULSE UNITS.

MASSMULT        EXTEND
                QXCH    BUF
                DXCH    MPAC
                TC      DMP
                ADRES   MASS
                TC      DMP             # LEAVES PROPERLY SCALED FORCE IN MPAC
                ADRES   SCALEFAC
                TC      TPAGREE
                CA      MPAC
                EXTEND
                BZF     +3
                CAF     POSMAX
                TC      BUF
                DXCH    MPAC +1
                TC      BUF
```

```
# CONSTANTS:-

FEXTRA          =       BIT13           # FEXT +5.13309020E+ 4

/AF/CNST        DEC     .13107

# * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
```

```
                EBANK=  E2DPS
                COUNT*  $$/F2DPS

# ***********************************************************
# LUNAR LANDING FLIGHT SEQUENCE TABLES
# ***********************************************************

# FLIGHT SEQUENCE TABLES ARE ARRANGED BY FUNCTION.  THEY ARE REFERENCED USING AS AN INDEX THE REGISTER WCHPHASE:
#       WCHPHASE = -1 ---> IGNALG
#       WCHPHASE =  0 ---> BRAKQUAD
#       WCHPHASE =  1 ---> APPRQUAD
#       WCHPHASE =  2 ---> VERTICAL

#***********************************************************

# ROUTINES FOR STARTING NEW GUIDANCE PHASES:

                TCF     TTFINCR         # IGNALG
NEWPHASE        TCF     TTFINCR         # BRAKQUAD
                TCF     STARTP64        # APPRQUAD
                TCF     P65START        # VERTICAL

# PRE-GUIDANCE COMPUTATIONS:

                TCF     CALCRGVG        # IGNALG
PREGUIDE        TCF     RGVGCALC        # BRAKQUAD
                TCF     REDESIG         # APPRQUAD
                TCF     RGVGCALC        # VERTICAL

# GUIDANCE EQUATIONS:

                TCF     TTF/8CL         # IGNALG
WHATGUID        TCF     TTF/8CL         # BRAKQUAD
                TCF     TTF/8CL         # APPRQUAD
                TCF     VERTGUID        # VERTICAL

# POST GUIDANCE EQUATION COMPUTATIONS:

                TCF     CGCALC          # IGNALG
AFTRGUID        TCF     CGCALC          # BRAKQUAD
                TCF     CGCALC          # APPRQUAD
                TCF     STEER?          # VERTICAL
```

```
# WINDOW VECTOR COMPUTATIONS:

                TCF     EXGSUB          # IGNALG
WHATEXIT        TCF     EXBRAK          # BRAKQUAD
                TCF     EXNORM          # APPRQUAD

# DISPLAY ROUTINES:

WHATDISP        TCF     P63DISPS        # BRAKQUAD
                TCF     P64DISPS        # APPRQUAD
                TCF     VERTDISP        # VERTICAL

# ALARM ROUTINE FOR TTF COMPUTATION:

                TCF     1406P00         # IGNALG
WHATALM         TCF     1406ALM         # BRAKQUAD
                TCF     1406ALM         # APPRQUAD

# INDICES FOR REFERENCING TARGET PARAMETERS

                OCT     0               # IGNALG
TARGTDEX        OCT     0               # BRAKQUAD
                OCT     34              # APPRQUAD

#*************************************************************************
# ENTRY POINTS:  ?GUIDSUB FOR THE IGNITION ALGORITHM, LUNLAND FOR SERVOUT
#*************************************************************************

# IGNITION ALGORITHM ENTRY:  DELIVERS N PASSES OF QUADRATIC GUIDANCE

?GUIDSUB        EXIT
                CAF     TWO             # N = 3
                TS      NGUIDSUB
                TCF     GUILDRET +2

GUIDSUB         TS      NGUIDSUB        # ON SUCCEEDING PASSES SKIP TTFINCR
                TCF     CALCRGVG

# NORMAL ENTRY:  CONTROL COMES HERE FROM SERVOUT

LUNLAND         TC      PHASCHNG
                OCT     00035           # GROUP 5:  RETAIN ONLY PIPA TASK
                TC      PHASCHNG
                OCT     05023           # GROUP 3:  PROTECT GUIDANCE WITH PRIO 21
                OCT     21000           #       JUST HIGHER THAN SERVICER'S PRIORITY
```

```
#*******************************************************************
# GUILDENSTERN:  AUTO-MODES MONITOR (R13)
#*******************************************************************

                COUNT*  $$/R13

# HERE IS THE PHILOSOPHY OF GUILDENSTERN:       ON EVERY APPEARANCE OR DISAPPEARANCE OF THE MANUAL THROTTLE
# DISCRETE TO SELECT P67 OR P66 RESPECTIVELY:   ON EVERY APPEARANCE OF THE ATTITUDE-HOLD DISCRETE TO SELECT P66
# UNLESS THE CURRENT PROGRAM IS P67 IN WHICH CASE THERE IS NO CHANGE

GUILDEN         EXTEND                          # IS UN-AUTO-THROTTLE DISCRETE PRESENT?
# STERN                                         # RSB 2009: NOT ORIGINALLY A COMMENT.
                READ CHAN30
                MASK    BIT5
                CCS     A
                TCF     STARTP67        # YES
P67NOW?         TC      CHECKMM         # NO:  ARE WE IN P67 NOW?
                DEC     67
                TCF     STABL?          # NO
STARTP66        TC      FASTCHNG        # YES
                TC      NEWMODEX
DEC66           DEC     66
                EXTEND
                DCA     HDOTDISP        # SET DESIRED ALTITUDE RATE = CURRENT
                DXCH    VDGVERT         #       ALTITUDE RATE.
STRTP66A        TC      INTPRET
                SLOAD   PUSH
                        PBIASZ
                SLOAD   PUSH
                        PBIASY
                SLOAD   VDEF
                        PBIASX
                VXSC    SET
                        BIASFACT
                        RODFLAG
                STOVL   VBIAS
                        TEMX
                VCOMP
                STOVL   OLDPIPAX
                        ZEROVECS
                STODL   DELVROD
                        RODSCALE
                STODL   RODSCAL1
                        PIPTIME
                STORE   LASTTPIP
                EXIT
                CAF     ZERO
                TS      FCOLD
                TS      FWEIGHT
                TS      FWEIGHT +1
VRTSTART        TS      WCHVERT
```

```
                CAF     TWO             # WCHPHASE = 2 ---> VERTICAL: P65,P66,P67
                TS      WCHPHOLD
                TS      WCHPHASE
                TC      BANKCALL        # TEMPORARY, I HOPE HOPE HOPE
                CADR    STOPRATE        # TEMPORARY, I HOPE HOPE HOPE
                TC      DOWNFLAG        # PERMIT X-AXIS OVERRIDE
                ADRES   XOVINFLG
                TC      DOWNFLAG
                ADRES   REDFLAG
                TCF     VERTGUID

STARTP67        TC      NEWMODEX        # NO HARM IN "STARTING" P67 OVER AND OVER
                DEC     67              # SO NO NEED FOR A FASTCHNG AND NO NEED
                CAF     ZERO            # TO SEE IF ALREADY IN P67.
                TS      RODCOUNT
                CAF     TEN
                TCF     VRTSTART

STABL?          CAF     BIT13           # IS UN-ATTITUDE-HOLD DISCRETE PRESENT?
                EXTEND
                RAND    CHAN31
                CCS     A
                TCF     GUILDRET        # YES ALL'S WELL

P66NOW?         CS      MODREG
                AD      DEC66
                EXTEND
                BZF     RESTART?

                CA      RODCOUNT        # NO. HAS THE ROD SWITCH BEEN "CLICKED"?
                EXTEND
                BZF     GUILDRET        # NO. CONTINUE WITH AUTOMATIC LANDING
                TCF     STARTP66        # YES. SWITCH INTO THE ROD MODE.

RESTART?        CA      FLAGWRD1        # HAS THERE BEEN A RESTART?
                MASK    RODFLBIT
                EXTEND
                BZF     STRTP66A        # YES.  REINITIALIZE BUT LEAVE VDGVERT AS
                                        #       IS.

                TCF     VERTGUID        # NO: CONTINUE WITH R.O.D.

# ****************************************************************************
# INITIALIZATION FOR THIS PASS
# ****************************************************************************

                COUNT*  $$/F2DPS

GUILDRET        CAF     ZERO
                TS      RODCOUNT
```

```
 +2             EXTEND
                DCA     TPIP
                DXCH    TPIPOLD

                TC      FASTCHNG

                EXTEND
                DCA     PIPTIME1
                DXCH    TPIP

                EXTEND
                DCA     TTF/8
                DXCH    TTF/8TMP

                CCS     FLPASS0
                TCF     TTFINCR

BRSPOT1         INDEX   WCHPHASE
                TCF     NEWPHASE

# *****************************************************************
# ROUTINES TO START NEW PHASES
# *****************************************************************

P65START        TC      NEWMODEX
                DEC     65
                CS      TWO
                TS      WCHVERT
                TC      DOWNFLAG        # PERMIT X-AXIS OVERRIDE
                ADRES   XOVINFLG
                TCF     TTFINCR

STARTP64        TC      NEWMODEX
                DEC     64
                CA      DELTTFAP        # AUGMENT TTF/8
                ADS     TTF/8TMP
                CA      BIT12           # ENABLE RUPT10
                EXTEND
                WOR     CHAN13
                TC      DOWNFLAG        # INITIALIZE REDESIGNATION FLAG
                ADRES   REDFLAG


#               (CONTINUE TO TTFINCR)

# ********************************************************************************
# INCREMENT TTF/8, UPDATE LAND FOR LUNAR ROTATION, DO OTHER USEFUL THINGS
# ********************************************************************************
#
#       TTFINCR COMPUTATIONS ARE AS FOLLOWS --
```

```
#               TTF/8 UPDATED FOR TIME SINCE LAST PASS:
#                       TTF/8 = TTF/8 + (TPIP - TPIPOLD)/8
#               LANDING SITE VECTOR UPDATED FOR LUNAR ROTATION:
#                       ____           ____   ____                    __
#                       LAND = /LAND/ UNIT(LAND - LAND(TPIP - TPIPOLD) * WM)
#               SLANT RANGE TO LANDING SITE, FOR DISPLAY:
#                                        ____   _
#                       RANGEDSP = ABVAL(LAND - R)

TTFINCR         TC      INTPRET
                DLOAD   DSU
                        TPIP
                        TPIPOLD
                SLR     PUSH            # SHIFT SCALES DELTA TIME TO 2(17) CSECS
                        11D
                VXSC    VXV
                        LAND
                        WM
                BVSU    RTB
                        LAND
                        NORMUNIT
                VXSC    VSL1
                        /LAND/
                STODL   LANDTEMP
                EXIT

                DXCH    MPAC
                DAS     TTF/8TMP        # NOW HAVE INCREMENTED TTF/8 IN TTF/8TMP

                TC      FASTCHNG

                EXTEND
                DCA     TTF/8TMP
                DXCH    TTF/8

                EXTEND
                DCA     LANDTEMP
                DXCH    LAND
                EXTEND
                DCA     LANDTEMP +2
                DXCH    LAND     +2
                EXTEND
                DCA     LANDTEMP +4
                DXCH    LAND     +4
```

```
                TC      TDISPSET
                TC      FASTCHNG        # SINCE REDESIG MAY CHANGE LANDTEMP

BRSPOT2         INDEX   WCHPHASE
                TCF     PREGUIDE

# **********************************************************************
# LANDING SITE PERTURBATION EQUATIONS
# **********************************************************************

REDESIG         CA      FLAGWRD6        # IS REDFLAG SET?
                MASK    REDFLBIT
                EXTEND
                BZF     RGVGCALC        # NO:  SKIP REDESIGNATION LOGIC

                CA      TREDES          # YES:  HAS TREDES REACHED ZERO?
                EXTEND
                BZF     RGVGCALC        # YES:  SKIP REDESIGNATION LOGIC

                INHINT
                CA      ELINCR1
                TS      ELINCR
                CA      AZINCR1
                TS      AZINCR
                TC      FASTCHNG

                CA      ZERO
                TS      ELINCR1
                TS      AZINCR1
                TS      ELINCR  +1
                TS      AZINCR  +1

                CA      FIXLOC          # SET PD TO 0
                TS      PUSHLOC

                TC      INTPRET
                VLOAD   VSU
                        LAND
                        R               #
                RTB     PUSH            # PUSH DOWN UNIT (LAND - R)
                        NORMUNIT
                VXV     VSL1
                        YNBPIP          #
                VXSC    PDDL            # PUSH DOWN - ELINCR(YNB * UNIT(LAND - R))
                        ELINCR
                        AZINCR
                VXSC    VSU
                        YNBPIP
                VAD     PUSH            # RESULTING VECTOR IS 1/2 REAL SIZE
```

```

                DLOAD   DSU             # MAKE SURE REDESIGNATION IS NOT
                        0               #       TOO CLOSE TO THE HORIZON.
                        DEPRCRIT
                BMN     DLOAD
                        REDES1
                        DEPRCRIT
                STORE   0
REDES1          DLOAD   DSU
                        LAND
                        R
                DDV     VXSC
                        0
                VAD     UNIT
                        R
                VXSC    VSL1
                        /LAND/
                STORE   LANDTEMP
                EXIT                    # LOOKANGL WILL BE COMPUTED AT RGVGCALC

                TC      FASTCHNG

                EXTEND
                DCA     LANDTEMP
                DXCH    LAND
                EXTEND
                DCA     LANDTEMP +2
                DXCH    LAND +2
                EXTEND
                DCA     LANDTEMP +4
                DXCH    LAND +4

                TCF     RGVGCALC

# *********************************************************************
# COMPUTE STATE IN GUIDANCE COORDINATES
# *********************************************************************
#
#       RGVGCALC COMPUTATIONS ARE AS FOLLOWS:--
#       VELOCITY RELATIVE TO THE SURFACE:
#               _______   _   _   __
#               ANGTERM = V + R * WM
#       STATE IN GUIDANCE COORDINATES:
#               ___   *   _   ____
#               RGU = CG (R - LAND)
#               ___   *   _   __  _
#               VGU = CG (V - WM * R)
```

```
#
#       HORIZONTAL VELOCITY FOR DISPLAY
#
#               VHORIZ = 8 ABVAL (0, VG , VG )
#                                      2    1
#       DEPRESSION ANGLE FOR DISPLAY:
#
#                                      _   ____  ______
#               LOOKANGL = ARCSIN(UNIT(R - LAND).XMBPIP)

CALCRGVG        TC      INTPRET         # IN IGNALG, COMPUTE V FROM INTEGRATION
                VLOAD   MXV             #       OUTPUT AND TRIM CORRECTION TERM
                        VATT1           #       COMPUTED LAST PASS AND LEFT IN UNFC/2
                        REFSMMAT
                VSR1    VAD
                        UNFC/2
                STORE   V
                EXIT

RGVGCALC        TC      INTPRET         # ENTER HERE TO RECOMPUTE RG AND VG
                VLOAD   VXV
                        R
                        WM
                VAD     VSR2            # RESCALE TO UNITS OF 2(9) M/CS
                        V
                STORE   ANGTERM
                MXV
                        CG              # NO SHIFT SINCE ANGTERM IS DOUBLE SIZED
                STORE   VGU
                PDDL    VDEF            # FORM (0,VG ,VG ) IN UNITS OF 2(10) M/CS
                        ZEROVECS        #            2   1
                ABVAL   SL3
                STOVL   VHORIZ          # VHORIZ FOR DISPLAY DURING P65.
                        R               #            _   ____
                VSU     PUSH            # PUSH DOWN R - LAND
                        LAND
                MXV     VSL1
                        CG
                STORE   RGU
                ABVAL
                STOVL   RANGEDSP
                RTB     DOT             # NOW IN MPAC IS SINE(LOOKANGL)/4
                        NORMUNIT
                        XNBPIP
                EXIT

                CA      FIXLOC          # RESET PUSH DOWN POINTER
                TS      PUSHLOC
```

```
                CA      MPAC            # COMPUTE LOOKANGLE ITSELF
                DOUBLE
                TC      BANKCALL
                CADR    SPARCSIN -1
                AD      1/2DEG
                EXTEND
                MP      180DEGS
                TS      LOOKANGL        # LOOKANGL FOR DISPLAY DURING P64

BRSPOT3         INDEX   WCHPHASE
                TCF     WHATGUID

# ***************************************************************************
# TTF/8 COMPUTATION
# ***************************************************************************

TTF/8CL         TC      INTPRETX
                DLOAD*
                        JDG2TTF,1
                STODL*  TABLTTF +6      # A(3) = 8 JDG  TO TABLTTF
                        ADG2TTF,1       #              2
                STODL   TABLTTF +4      # A(2) = 6 ADG  TO TABLTTF
                        VGU     +4      #              2
                DMP     DAD*
                        3/4DP
                        VDG2TTF,1
                STODL*  TABLTTF +2      # A(1) = (6 VGU  + 18 VDG )/8 TO TABLTTF
                        RDG +4,1        #               2         2
                DSU     DMP
                        RGU +4
                        3/8DP
                STORE   TABLTTF         # A(0) = -24 (RGU  - RDG )/64 TO TABLTTF
                EXIT                    #                 2      2

                CA      BIT8
                TS      TABLTTF +10     # FRACTIONAL PRECISION FOR TTF TO TABLE

                EXTEND
                DCA     TTF/8
                DXCH    MPAC            # LOADS TTF/8 (INITIAL GUESS) INTO MPAC
                CAF     TWO             # DEGREE - ONE
                TS      L
                CAF     TABLTTFL
                TC      ROOTPSRS        # YIELDS TTF/8 IN MPAC
                INDEX   WCHPHASE
                TCF     WHATALM

                EXTEND                  # GOOD RETURN
                DCA     MPAC            # FETCH TTF/8 KEEPING IT IN MPAC
                DXCH    TTF/8           # CORRECTED TTF/8
```

```
                TC      TDISPSET

#               (CONTINUE TO QUADGUID)

# ********************************************************************************
# MAIN GUIDANCE EQUATION
# ********************************************************************************
#
#       AS PUBLISHED --
#
#                             ___   __     ___   __
#               ___   ___   6(VDG + VG)   12(RDG - RG)
#               ACG = ADG + ----------- + ------------
#                               TTF         (TTF)(TTF)
#       AS HERE PROGRAMMED --
#
#                     3 (1/4(RDG - RG)    ___   __)
#                     - (------------- + VDG + VG)
#               ___   4 (    TTF/8              )    ___
#               ACG = ---------------------------- + ADG
#                                 TTF/8

QUADGUID        CS      TTF/8
                AD      LEADTIME        # LEADTIME IS A NEGATIVE NUMBER
                AD      POSMAX          # SAFEGUARD THE COMPUTATIONS THAT FOLLOW
                TS      L               #       BY FORCING -TTF*LEADTIME > OR = ZERO
                CS      L
                AD      L
                ZL
                EXTEND
                DV      TTF/8
                TS      BUF             # - RATIO OF LAG-DIMINISHED TTF TO TTF
                EXTEND
                SQUARE
                TS      BUF +1
                AD      BUF
                XCH     BUF +1          # RATIO SQUARED - RATIO
                AD      BUF +1
                TS      MPAC            # COEFFICIENT FOR VGU TERM
                AD      BUF +1
                INDEX   FIXLOC
                TS      26D             # COEFFICIENT FOR RDG-RGU TERM
                AD      BUF +1
                INDEX   FIXLOC
                TS      28D             # COEFFICIENT FOR VDG TERM
                AD      BUF
                AD      POSMAX
```

```
                AD      BUF +1
                AD      BUF +1
                INDEX   FIXLOC
                TS      30D             # COEFFICIENT FOR ADG TERM

                CAF     ZERO
                TS      MODE

                TC      INTPRETX
                VXSC    PDDL
                        VGU
                        28D
                VXSC*   PDVL*
                        VDG,1
                        RDG,1
                VSU     V/SC
                        RGU
                        TTF/8
                VSR2    VXSC
                        26D
                VAD     VAD
                V/SC    VXSC
                        TTF/8
                        3/4DP
                PDDL    VXSC*
                        30D
                        ADG,1
                VAD
AFCCALC1        VXM     VSL1            # VERGUID COMES HERE
                        CG
                PDVL    V/SC
                        GDT/2
                        GSCALE
                BVSU    STADR
                STORE   UNFC/2          # UNFC/2 NEED NOT BE UNITIZED
                ABVAL
AFCCALC2        STODL   /AFC/           # MAGNITUDE OF AFC FOR THROTTLE
                        UNFC/2          # VERTICAL COMPONENT
                DSQ     PDDL
                        UNFC/2 +2       # OUT-OF-PLANE
                DSQ     PDDL
                        HIGHESTF
                DDV     DSQ
                        MASS            #                        2    2    2
                DSU     DSU             # AMAXHORIZ = SQRT(ATOTAL - A  - A  )
                BPL     DLOAD           #                            1    0
                        AFCCALC3
                        ZEROVECS
AFCCALC3        SQRT    DAD
                        UNFC/2 +4
```

```
                BPL     BDSU
                        AFCCLEND
                        UNFC/2 +4
                STORE   UNFC/2 +4
AFCCLEND        EXIT
                TC      FASTCHNG

                CA      WCHPHASE        # PREPARE FOR PHASE SWITCHING LOGIC
                TS      WCHPHOLD
                INCR    FLPASS0         # INCREMENT PASS COUNTER

BRSPOT4         INDEX   WCHPHASE
                TCF     AFTRGUID

# ************************************************************************
# ERECT GUIDANCE-STABLE MEMBER TRANSFORMATION MATRIX
# ************************************************************************

CGCALC          CAF     EBANK5
                TS      EBANK
                EBANK=  TCGIBRAK
                EXTEND
                INDEX   WCHPHASE
                INDEX   TARGTDEX
                DCA     TCGFBRAK
                INCR    BBANK
                INCR    BBANK
                EBANK=  TTF/8
                AD      TTF/8
                XCH     L
                AD      TTF/8
                CCS     A
                CCS     L
                TCF     EXTLOGIC
                TCF     EXTLOGIC
                NOOP

                TC      INTPRETX
                VLOAD   UNIT
                        LAND
                STODL   CG
                        TTF/8
                DMP*    VXSC
                        GAINBRAK,1      # NUMERO MYSTERIOSO
                        ANGTERM
                VAD
                        LAND
                VSU     RTB
                        R
                        NORMUNIT
```

```
                VXV     RTB
                        LAND
                        NORMUNIT
                STOVL   CG +6           # SECOND ROW
                        CG
                VXV     VSL1
                        CG +6
                STORE   CG +14
                EXIT

#               (CONTINUE TO EXTLOGIC)
#
# ***********************************************************************
# PREPARE TO EXIT
# ***********************************************************************
#
# DECIDE (1) HOW TO EXIT, AND (2) WHETHER TO SWITCH PHASES
#
EXTLOGIC        INDEX   WCHPHASE        # WCHPHASE = 1   APPRQUAD
                CA      TENDBRAK        # WCHPHASE = 0   BRAKQUAD
                AD      TTF/8

EXSPOT1         EXTEND
                INDEX   WCHPHASE
                BZMF    WHATEXIT

                TC      FASTCHNG

                CA      WCHPHOLD
                AD      ONE
                TS      WCHPHASE
                CA      ZERO
                TS      FLPASSO         # RESET FLPASSO

                INDEX   WCHPHOLD
                TCF     WHATEXIT

# ***********************************************************************
# ROUTINES FOR EXITING FROM LANDING GUIDANCE
# ***********************************************************************
#
# 1.    EXGSUB IS THE RETURN WHEN GUIDSUB IS CALLED BY THE IGNITION ALGORITHM.
# 2.    EXBRAK IN THE EXIT USED DURING THE BRAKING PHASE.  IN THIS CASE UNIT(R) IS THE WINDOW POINTING VECTOR.
# 3.    EXNORM IS THE EXIT USED AT OTHER TIMES DURING THE BURN.
# (EXOVFLOW IS A SUBROUTINE OF EXBRAK AND EXNORM CALLED WHEN OVERFLOW OCCURRED ANYWHERE IN GUIDANCE.)

EXGSUB          TC      INTPRET         # COMPUTE TRIM VELOCITY CORRECTION TERM.
```

```
                VLOAD   RTB
                        UNFC/2
                        NORMUNIT
                VXSC    VXSC
                        ZOOMTIME
                        TRIMACCL
                STORE   UNFC/2
                EXIT

                CCS     NGUIDSUB
                TCF     GUIDSUB
                CCS     NIGNLOOP
                TCF     +3
                TC      ALARM
                OCT     01412

 +3             TC      POSTJUMP
                CADR    DDUMCALC

EXBRAK          TC      INTPRET
                VLOAD
                        UNIT/R/
                STORE   UNWC/2
                EXIT
                TCF     STEER?

EXNORM          TC      INTPRET
                VLOAD   VSU
                        LAND
                        R
                RTB
                        NORMUNIT
                STORE   UNWC/2          # UNIT(LAND - R) IS TENTATIVE CHOICE
                VXV     DOT
                        XNBPIP
                        CG +6
                EXIT                    # WITH PROJ IN MPAC 1/8 REAL SIZE

                CS      MPAC            # GET COEFFICIENT FOR CG +14
                AD      PROJMAX
                AD      POSMAX
                TS      BUF
                CS      BUF
                ADS     BUF             # RESULT IS 0 IF PROJMAX - PROJ NEGATIVE

                CS      PROJMIN         # GET COEFFICIENT FOR UNIT(LAND - R)
                AD      MPAC
                AD      POSMAX
                TS      BUF +1
                CS      BUF +1
```

```
                ADS     BUF +1          # RESULT IS 0 IF PROJ - PROJMIN NEGATIVE

                CAF     FOUR
UNWCLOOP        MASK    SIX
                TS      Q
                CA      EBANK5
                TS      EBANK
                EBANK=  CG
                CA      BUF
                EXTEND
                INDEX   Q
                MP      CG +14
                INCR    BBANK
                EBANK=  UNWC/2
                INDEX   Q
                DXCH    UNWC/2
                EXTEND
                MP      BUF +1
                INDEX   Q
                DAS     UNWC/2
                CCS     Q
                TCF     UNWCLOOP

                INCR    BBANK
                EBANK=  PIF

STEER?          CA      FLAGWRD2        # IF STEERSW DOWN NO OUTPUTS
                MASK    STEERBIT
                EXTEND
                BZF     RATESTOP

EXVERT          CA      OVFIND          # IF OVERFLOW ANYWHERE IN GUIDANCE
                EXTEND                  #       DON'T CALL THROTTLE OR FINDCDUW
                BZF     +13

EXOVFLOW        TC      ALARM           # SOUND THE ALARM NON-ABORTIVELY
                OCT     01410

RATESTOP        CAF     BIT13           # ARE WE IN ATTITUDE-HOLD?
                EXTEND
                RAND    CHAN31
                EXTEND
                BZF     DISPEXIT        # YES

                TC      BANKCALL        # NO:  DO A STOPRATE
                CADR    STOPRATE

                TCF     DISPEXIT

GDUMP1          TC      THROTTLE
```

```
                TC      INTPRET
                CALL
                        FINDCDUW -2
                EXIT

#               (CONTINUE TO DISPEXIT)

# ************************************************************************
# GUIDANCE LOOP DISPLAYS
# ************************************************************************

DISPEXIT        EXTEND                  # KILL GROUP 3:  DISPLAYS WILL BE
                DCA     NEGO            #       RESTORED BY NEXT GUIDANCE CYCLE.
                DXCH    -PHASE3

 +3             CS      FLAGWRD8        # IF FLUNDISP IS SET, NO DISPLAY THIS PASS
                MASK    FLUNDBIT
                EXTEND
                BZF     ENDLLJOB        # TO PICK UP THE TAG

                INDEX   WCHPHOLD
                TCF     WHATDISP

-2              TC      PHASCHNG        # KILL GROUP 5
                OCT     00035

P63DISPS        CAF     V06N63
DISPCOMN        TC      BANKCALL
                CADR    REGODSPR

ENDLLJOB        TCF     ENDOFJOB

P64DISPS        CA      TREDES          # HAS TREDES REACHED ZERO?
                EXTEND
                BZF     RED-OVER        # YES: CLEAR REDESIGNATION FLAG

                CS      FLAGWRD6        # NO:  IS REDFLAG SET?
                MASK    REDFLBIT
                EXTEND
                BZF     REDES-OK        # YES:  DO STATIC DISPLAY

                CAF     V06N64          # OTHERWISE USE FLASHING DISPLAY
                TC      BANKCALL
                CADR    REFLASHR
                TCF     GOTOPOOH        # TERMINATE
                TCF     P64CEED         # PROCEED       PERMIT REDESIGNATIONS
                TCF     P64DISPS        # RECYCLE
```

```
                TCF     ENDLLJOB

P64CEED         CAF     ZERO
                TS      ELINCR1
                TS      AZINCR1

                TC      UPFLAG          # ENABLE REDESIGNATION LOGIC
                ADRES   REDFLAG

                TCF     ENDOFJOB

RED-OVER        TC      DOWNFLAG
                ADRES   REDFLAG
REDES-OK        CAF     V06N64
                TCF     DISPCOMN


VERTDISP        CAF     V06N60
                TCF     DISPCOMN


# ****************************************************************************
# GUIDANCE FOR P65
# ****************************************************************************

VERTGUID        CCS     WCHVERT
                TCF     P67VERT         # POSITIVE NON-ZERO ---> P67
                TCF     P66VERT         # +0
#
#       THE P65 GUIDANCE EQUATION IS AS FOLLOWS --
#
#                   V2FG - VGU
#               ACG = ----------
#                       TAUVERT

P65VERT         TC      INTPRET
                VLOAD   VSU
                        V2FG
                        VGU
                V/SC    GOTO
                        TAUVERT
                        AFCCALC1
```

```
# ***********************************************************
# GUIDANCE FOR P66
# ***********************************************************

P66VERT         TC      POSTJUMP
                CADR    P66VERTA

P67VERT         TC      PHASCHNG        # TERMINATE GROUP 3.
                OCT     00003

                TC      INTPRET
                VLOAD   GOTO
                        V
                        VHORCOMP

                SETLOC  P66LOC
                BANK
                COUNT*  $$/F2DPS

RODTASK         CAF     PRIO22
                TC      FINDVAC
                EBANK=  DVCNTR
                2CADR   RODCOMP

                TCF     TASKOVER

P66VERTA        TC      PHASCHNG        # TERMINATE GROUP 3.
                OCT     00003

                CAF     1SEC
                TC      TWIDDLE
                ADRES   RODTASK

RODCOMP         INHINT
                CAF     ZERO
                XCH     RODCOUNT
                EXTEND
                MP      RODSCAL1
                DAS     VDGVERT         # UPDATE DESIRED ALTITUDE RATE.

                EXTEND                  # SET OLDPIPAX,Y,Z = PIPAX,Y,Z
                DCA     PIPAX
                DXCH    OLDPIPAX
                DXCH    RUPTREG1        # SET RUPTREG1,2,3 = OLDPIPAX,Y,Z
                CA      PIPAZ
                XCH     OLDPIPAZ
                XCH     RUPTREG3

                EXTEND                  # SNAPSHOT TIME OF PIPA READING.
                DCA     TIME2
```

```
                DXCH    THISTPIP

                CA      OLDPIPAX
                AD      PIPATMPX
                TS      MPAC            # MPAC(X) = PIPAX + PIPATMPX
                CA      OLDPIPAY
                AD      PIPATMPY
                TS      MPAC +3         # MPAC(Y) = PIPAY + PIPATMPY
                CA      OLDPIPAZ
                AD      PIPATMPZ
                TS      MPAC +5         # MPAC(Z) = PIPAZ + PIPATMPZ

                CS      OLDPIPAX
                AD      TEMX
                AD      RUPTREG1
                TS      DELVROD
                CS      OLDPIPAY
                AD      TEMY
                AD      RUPTREG2
                TS      DELVROD +2
                CS      OLDPIPAZ
                AD      TEMZ
                AD      RUPTREG3
                TS      DELVROD +4

                CAF     ZERO
                TS      MPAC +1         # ZERO LO-ORDER MPAC COMPONENTS
                TS      MPAC +4
                TS      MPAC +6
                TS      TEMX            # ZERO TEMX, TEMY, AND TEMZ SO WE WILL
                TS      TEMY            #       KNOW WHEN READACCS CHANGES THEM.
                TS      TEMZ
                CS      ONE
                TS      MODE
                TC      INTPRET
ITRPNT1         VXSC    PDDL            # SCALE MPAC TO M/CS *2(-7) AND PUSH    (6)
                        KPIP1
                        THISTPIP
                DSU
                        PIPTIME
                STORE   30D             # 30-31D CONTAINS TIME IN CS SINCE PIPTIME
                DDV     PDVL            #                                       (8)
                        4SEC(28)
                        GDT/2
                VSU     VXSC            #                                       (6)
                        VBIAS
                VSL2    VAD
                        V
                VAD     STADR           #                                       (0)
                STOVL   24D             # STORE UPDATED VELOCITY IN 24-29D
```

```
                R
        UNIT
        STORE   14D
        DOT     SL1
                24D
        STODL   HDOTDISP        # UPDATE HDOTDISP RATE FOR NOUN 63.
                30D
        SL      DMP
                11D
                HDOTDISP
        DAD     DSU
                36D
                /LAND/
        STODL   HCALC1          # UPDATE HCALC1 FOR NOUN 63.
                HDOTDISP
        BDSU    DDV
                VDGVERT
                TAUROD
        PDVL    ABVAL           #                               (2)
                GDT/2
        DDV     SR2
                GSCALE
        STORE   20D
        DAD                     #                               (0)
        PDVL    CALL            #                               (2)
                UNITX
                CDU*NBSM
        DOT
                14D
        STORE   22D
        BDDV    STADR           #                               (0)
        STOVL   /AFC/
                DELVROD
        VXSC    VAD
                KPIP1
                VBIAS
        ABVAL   PDDL            #                               (2)
                THISTPIP
        DSU     PDDL            #                               (4)
                LASTTPIP
                THISTPIP
        STODL   LASTTPIP        #                               (2)
        DDV     BDDV            #                               (0)
                SHFTFACT
        PDDL    DMP             #                               (2)
                FWEIGHT
                BIT1H
        DDV     DDV
                MASS
                SCALEFAC
```

```
                DAD         PDDL            #                           (4)
                            0D
                            20D
                DDV         DSU             #                           (2)
                            22D
                DMP         DAD
                            LAG/TAU
                            /AFC/
                PDDL        DDV             #                           (4)
                            MAXFORCE
                            MASS
                PDDL        DDV             #                           (6)
                            MINFORCE
                            MASS
                PUSH        BDSU            #                           (8)
                            2D
                BMN         DLOAD           #                           (6)
                            AFCSPOT
                DLOAD       PUSH            #                           (6)
                BDSU        BPL
                            2D
                            AFCSPOT
                DLOAD                       #                           (4)
AFCSPOT         DLOAD                       #                           (2), (4), OR (6)
                SETPD                       #                           (2)
                            2D
                STODL       /AFC/           #                           (0)
ITRPNT2         EXIT
                DXCH        MPAC            # MPAC = MEASURED ACCELERATION.
                TC          BANKCALL
                CADR        THROTTLE +3
                TC          INTPRET
                VLOAD                       # PICK UP UPDATED VELOCITY VECTOR.
                            24D
VHORCOMP        VSL2        VAD
                            DELVS
                VSR2        PDVL
                            R
                UNIT        VXSC
                            HDOTDISP
                VSL1        BVSU
                ABVAL
                STORE       VHORIZ
                EXIT
                TC          BANKCALL        # PUT UP V06N60 DISPLAY BUT AVOID PHASCHNG
                CADR        DISPEXIT +3

BIT1H           OCT         00001
SHFTFACT        2DEC        1 B-17
```

```
BIASFACT        2DEC    655.36 B-28

# ************************************************************************************
# REDESIGNATOR TRAP
# ************************************************************************************

                BANK    11
                SETLOC  F2DPS*11
                BANK

                COUNT*  $$/F2DPS

PITFALL         XCH     BANKRUPT
                EXTEND
                QXCH    QRUPT

                TC      CHECKMM         # IF NOT IN P64, NO REASON TO CONTINUE
                DEC     64
                TCF     RESUME

                EXTEND
                READ    CHAN31
                COM
                MASK    ALL4BITS
                TS      ELVIRA
                CAF     TWO
                TS      ZERLINA
                CAF     FIVE
                TC      TWIDDLE
                ADRES   REDESMON
                TCF     RESUME

# REDESIGNATOR MONITOR (INITIATED BY PITFALL)

PREMON1         TS      ZERLINA
PREMON2         CAF     SEVEN
                TC      VARDELAY
REDESMON        EXTEND
                READ    31
                COM
                MASK    ALL4BITS
                XCH     ELVIRA
                TS      L
                CCS     ELVIRA          # DO ANY BITS APPEAR THIS PASS?
                TCF     PREMON2         # Y:    CONTINUE MONITOR

                CCS     L               # N:    ANY LAST PASS?
                TCF     COUNT'EM        #       Y:      COUNT 'EM, RESET RUPT, TERMINATE
```

```
                CCS     ZERLINA         #       N:      HAS ZERLINA REACHED ZERO YET?
                TCF     PREMON1         #               N:      DIMINISH ZERLINA, CONTINUE
RESETRPT        CAF     BIT12           #               Y:      RESET RUPT. TERMINATE
                EXTEND
                WOR     CHAN13
                TCF     TASKOVER

COUNT'EM        CAF     BIT13           # ARE WE IN ATTITUDE-HOLD?
                EXTEND
                RAND    CHAN31
                EXTEND
                BZF     RESETRPT        # YES: SKIP REDESIGNATION LOGIC.

                CA      L               # NO.
                MASK    -AZBIT
                CCS     A
-AZ             CS      AZEACH
                ADS     AZINCR1
                CA      L
                MASK    +AZBIT
                CCS     A
+AZ             CA      AZEACH
                ADS     AZINCR1
                CA      L
                MASK    -ELBIT
                CCS     A
-EL             CS      ELEACH
                ADS     ELINCR1
                CA      L
                MASK    +ELBIT
                CCS     A
+EL             CA      ELEACH
                ADS     ELINCR1
                TCF     RESETRPT

# THESE EQUIVALENCES ARE BASED ON GSOP CHAPTER 4, REVISION 16 OF P64LM

+ELBIT          =       BIT2            # -PITCH
-ELBIT          =       BIT1            # +PITCH
+AZBIT          =       BIT5
-AZBIT          =       BIT6
```

```
ALL4BITS        OCT     00063
AZEACH          DEC     .03491          # 2 DEGREES
ELEACH          DEC     .00873          # 1/2 DEGREE

# ****************************************************************
# R.O.D. TRAP
# ****************************************************************

                BANK    20
                SETLOC  RODTRAP
                BANK
                COUNT*  $$/F2DPS        # ************************

DESCBITS        MASK    BIT7            # COME HERE FROM MARKRUPT CODING WITH BIT
                CCS     A               #       7 OR 6 OF CHANNEL 16 IN A; BIT 7 MEANS
                CS      TWO             #       - RATE INCREMENT, BIT 6 + INCREMENT.
                AD      ONE
                ADS     RODCOUNT
                TCF     RESUME          # TRAP IS RESET WHEN SWITCH IS RELEASED

                BANK    31
                SETLOC  F2DPS*31
                BANK

                COUNT*  $$/F2DPS

# ***********************************************************************************
# DOUBLE PRECISION ROOT FINDER SUBROUTINE (BY ALLAN KLUMPP)
# ***********************************************************************************
#
#                                                         N        N-1
#       ROOTPSRS FINDS ONE ROOT OF THE POWER SERIES A X  + A    X     + ... + A X + A
#                                                   N      N-1                1     0
# USING NEWTON'S METHOD STARTING WITH AN INITIAL GUESS FOR THE ROOT.  THE ENTERING DATA MUST BE AS FOLLOWS:
#       A       SP      LOC-3           ADRES FOR REFERENCING PWR COF TABL
#       L       SP      N-1             N IS THE DEGREE OF THE POWER SERIES
#       MPAC    DP      X               INITIAL GUESS FOR ROOT
#
#       LOC-2N  DP      A(0)
#               ...
#       LOC     DP      A(N)
#       LOC+2   SP      PRECROOT         PREC RQD OF ROOT (AS FRACT OF 1ST GUESS)
#
```

```
# THE DP RESULT IS LEFT IN MPAC UPON EXIT, AND A SP COUNT OF THE ITERATIONS TO CONVERGENCE IS LEFT IN MPAC+2.
# RETURN IS NORMALLY TO LOC(TC ROOTPSRS)+3.  IF ROOTPSRS FAILS TO CONVERGE TO IN 8 PASSES, RETURN IS TO LOC+1 AND
# OUTPUTS ARE NOT TO BE TRUSTED.
#
# PRECAUTION:  ROOTPSRS MAKES NO CHECKS FOR OVERFLOW OR FOR IMPROPER USAGE.  IMPROPER USAGE COULD
# PRECLUDE CONVERGENCE OR REQUIRE EXCESSIVE ITERATIONS.  AS A SPECIFIC EXAMPLE, ROOTPSRS FORMS A DERIVATIVE
# COEFFICIENT TABLE BY MULTIPLYING EACH A(I) BY I, WHERE I RANGES FROM 1 TO N.  IF AN ELEMENT OF THE DERIVATIVE
# COEFFICIENT TABLE = 1 OR >1 IN MAGNITUDE, ONLY THE EXCESS IS RETAINED.  ROOTPSRS MAY CONVERGE ON THE CORRECT
# ROOT NONETHELESS, BUT IT MAY TAKE AN EXCESSIVE NUMBER OF ITERATIONS.  THEREFORE THE USER SHOULD RECOGNIZE:
#       1.  USER'S RESPONSIBILITY TO ASSUR THAT I X A(I) < 1 IN MAGNITUDE FOR ALL I.
#       2.  USER'S RESPONSIBILITY TO ASSURE OVERFLOW WILL NOT OCCUR IN EVALUATING EITHER THE RESIDUAL OR THE DERIVATIVE
#           POWER SERIES.  THIS OVERFLOW WOULD BE PRODUCED BY SUBROUTINE POWRSERS, CALLED BY ROOTPSRS, AND MIGHT NOT
#           PRECLUDE EVENTUAL CONVERGENCE.
#       3.  AT PRESENT, ERASABLE LOCATIONS ARE RESERVED ONLY FOR N UP TO 5.  AN N IN EXCESS OF 5 WILL PRODUCE CHAOS.
#           ALL ERASABLES USED BY ROOTPSRS ARE UNSWITCHED LOCATED IN THE REGION FROM MPAC-33 OCT TO MPAC+7.
#       4.  THE ITERATION COUNT RETURNED IN MPAC+2 MAY BE USED TO DETECT ABNORMAL PERFORMANCE.

                                        # STORE ENTERING DATA, INITIALIZE ERASABLES
ROOTPSRS        EXTEND
                QXCH    RETROOT         # RETURN ADRES
                TS      PWRPTR          # PWR TABLE POINTER
                DXCH    MPAC +3         # PWR TABLE ADRES, N-1
                CA      DERTABLL
                TS      DERPTR          # DER TABL POINTER
                TS      MPAC +5         # DER TABL ADRES
                CCS     MPAC +4         # NO POWER SERIES DEGREE 1 OR LESS
                TS      MPAC +6         # N-2
                CA      ZERO            # MODE USED AS ITERATION COUNTER.  MODE
                TS      MODE            # MUST BE POS SO ABS WON'T COMP MPAC+3 ETC.

                                        # COMPUTE CRITERION TO STOP ITERATING
                EXTEND
                DCA     MPAC            # FETCH ROOT GUESS, KEEPING IT IN MPAC
                DXCH    ROOTPS          # AND IN ROOTPS
                INDEX   MPAC +3         # PWR TABLE ADRES
                CA      5               # PRECROOT TO A
                TC      SHORTMP         # YIELDS DP PRODUCT IN MPAC
                TC      USPRCADR
                CADR    ABS             # YIELDS ABVAL OF CRITERION ON DX IN MPAC
                DXCH    MPAC
                DXCH    DXCRIT          # CRITERION

                                        # SET UP DER COF TABL
```

```
                EXTEND
                INDEX   PWRPTR
                DCA     3
                DXCH    MPAC            # A(N) TO MPAC

                CA      MPAC +4         # N-1 TO A

DERCLOOP        TS      PWRCNT          # LOOP COUNTER
                AD      ONE
                TC      DMPNSUB         # YIELDS DERCOF = I X A(I) IN MPAC
                EXTEND
                INDEX   PWRPTR
                DCA     1
                DXCH    MPAC            # (I-1) TO MPAC, FETCHING DERCOF
                INDEX   DERPTR
                DXCH    3               # DERCOF TO DER TABLE
                CS      TWO
                ADS     PWRPTR          # DECREMENT PWR POINTER
                CS      TWO
                ADS     DERPTR          # DECREMENT DER POINTER
                CCS     PWRCNT
                TCF     DERCLOOP

                                        # CONVERGE ON ROOT
ROOTLOOP        EXTEND
                DCA     ROOTPS          # FETCH CURRENT ROOT
                DXCH    MPAC            # LEAVE IN MPAC
                EXTEND
                DCA     MPAC +5         # LOAD A, L WITH DER TABL ADRES, N-2
                TC      POWRSERS        # YIELDS DERIVATIVE IN MPAC

                EXTEND
                DCA     ROOTPS
                DXCH    MPAC            # CURRENT ROOT TO MPAC, FETCHING DERIVATIVE
                DXCH    BUF             # LEAVE DERIVATIVE IN BUF AS DIVISOR
                EXTEND
                DCA     MPAC +3         # LOAD A, L WITH PWR TABL ADRES, N-1
                TC      POWRSERS        # YIELDS RESIDUAL IN MPAC

                TC      USPRCADR
                CADR    DDV/BDDV        # YIELDS -DX IN MPAC

                EXTEND
                DCS     MPAC            # FETCH DX, LEAVING -DX IN MPAC
                DAS     ROOTPS          # CORRECTED ROOT NOW IN ROOTPS

                TC      USPRCADR
                CADR    ABS             # YIELDS ABS(DX) IN MPAC
                EXTEND
```

```
                DCS     DXCRIT
                DAS     MPAC            # ABS(DX)-ABS(DXCRIT) IN MPAC

                CA      MODE
                MASK    BIT4            # KLUMPP SAYS GIVE UP AFTER EIGHT PASSES
                CCS     A
BADROOT         TC      RETROOT

                INCR    MODE            # INCREMENT ITERATION COUNTER
                CCS     MPAC            # TEST HI ORDER DX
                TCF     ROOTLOOP
                TCF     TESTLODX
                TCF     ROOTSTOR
TESTLODX        CCS     MPAC +1         # TEST LO ORDER DX
                TCF     ROOTLOOP
                TCF     ROOTSTOR
                TCF     ROOTSTOR
ROOTSTOR        DXCH    ROOTPS
                DXCH    MPAC
                CA      MODE
                TS      MPAC +2         # STORE SP ITERATION COUNT IN MPAC+2
                INDEX   RETROOT
                TCF     2

DERTABLL        ADRES   DERCOFN -3

# ****************************************************************************
# TRASHY LITTLE SUBROUTINES
# ****************************************************************************

INTPRETX        INDEX   WCHPHASE        # SET X1 ON THE WAY TO THE INTERPRETER
                CS      TARGTDEX
                INDEX   FIXLOC
                TS      X1
                TCF     INTPRET

TDISPSET        CA      TTF/8
                EXTEND
                MP      TSCALINV
                DXCH    TTFDISP

                CA      EBANK5          # TREDES BECOMES ZERO TWO PASSES
                TS      EBANK           #       BEFORE TCGFAPPR IS REACHED
                EBANK=  TCGFAPPR
                CA      TCGFAPPR
                INCR    BBANK
                INCR    BBANK
                EBANK=  TTF/8
```

```
                AD      TTF/8
                EXTEND
                MP      TREDESCL
                AD      -DEC103
                AD      NEGMAX
                TS      L
                CS      L
                AD      L
                AD      +DEC99
                AD      POSMAX
                TS      TREDES
                CS      TREDES
                ADS     TREDES
                TC      Q

1406P00         TC      POODOO
                OCT     01406
1406ALM         TC      ALARM
                OCT     01406
                TCF     RATESTOP

# ************************************************************************
# SPECIALIZED "PHASCHNG" SUBROUTINE
# ************************************************************************

                EBANK=  PHSNAME2
FASTCHNG        CA      EBANK3          # SPECIALIZED 'PHASCHNG' ROUTINE
                XCH     EBANK
                DXCH    L
                TS      PHSNAME3
                LXCH    EBANK
                EBANK=  E2DPS
                TC      A

# ****************************************************************************************
# PARAMETER TABLE INDIRECT ADDRESSES
# ****************************************************************************************

RDG             =       RBRFG
VDG             =       VBRFG
ADG             =       ABRFG
VDG2TTF         =       VBRFG*
ADG2TTF         =       ABRFG*
JDG2TTF         =       JBRFG*

# ****************************************************************************************
# LUNAR LANDING CONSTANTS
# ****************************************************************************************
```

```
TABLTTFL        ADRES    TABLTTF +3       # ADDRESS FOR REFERENCING TTF TABLE
TTFSCALE        =        BIT12
TSCALINV        =        BIT4
-DEC103         DEC      -103
+DEC99          DEC      +99
TREDESCL        DEC      -.08
180DEGS         DEC      +180
1/2DEG          DEC      +.00278
PROJMAX         DEC      .42262 B-3       # SIN(25')/8 TO COMPARE WITH PROJ
PROJMIN         DEC      .25882 B-3       # SIN(15')/8 TO COMPARE WITH PROJ
V06N63          VN       0663             # P63
V06N64          VN       0664             # P64
V06N60          VN       0660             # P65, P66, P67

                BANK     22
                SETLOC   LANDCNST
                BANK
                COUNT*   $$/F2DPS

HIGHESTF        2DEC     4.34546769 B-12
GSCALE          2DEC     100 B-11
3/8DP           2DEC     .375
3/4DP           2DEC     .750
DEPRCRIT        2DEC     -.02 B-1
```

```
# ***************************************************************************
# ***************************************************************************
```

```
                BANK    21
                SETLOC  R11
                BANK

                EBANK=  DVCNTR
                COUNT*  $$/R11

R10,R11         CS      FLAGWRD7        # IS SERVICER STILL RUNNING?
                MASK    AVEGFBIT
                CCS     A
                TCF     TASKOVER        # LET AVGEND TAKE CARE OF GROUP 2.
                CCS     PIPCTR
                TCF     +2
                TCF     LRHTASK         # LAST PASS. CALL LRHTASK.
 +2             TS      PIPCTR1

PIPCTR1         =       LADQSAVE
PIPCTR          =       PHSPRDT2
                CAF     OCT31
                TC      TWIDDLE
                ADRES   R10,R11
R10,R11A        CS      IMODES33        # IF LAMP TEST, DO NOT CHANGE LR LITES.
                MASK    BIT1
                EXTEND
                BZF     10,11

FLASHH?         MASK    FLGWRD11        # C(A) = 1 = HFLASH BIT
                EXTEND
                BZF     FLASHV?         # H FLASH OFF, SO LEAVE ALONE

                CA      HLITE
                TS      L
                TC      FLIP            # FLIP H LITE

FLASHV?         CA      VFLSHBIT        # VLASHBIT MUST BE BIT 2.
                MASK    FLGWRD11
                EXTEND
                BZF     10,11           # V FLASH OFF

                CA      VLITE
                TS      L
                TC      FLIP            # FLIP V LITE

10,11           CA      FLAGWRD9        # IS THE LETABORT FLAG SET ?
                MASK    LETABBIT
                EXTEND
                BZF     LANDISP         # NO.  PROCEED TO R10.

P71NOW?         CS      MODREG          # YES.  ARE WE IN P71 NOW?
```

```
                AD      1DEC71
                EXTEND
                BZF     LANDISP         # YES.  PROCEED TO R10.

                EXTEND                  # NO.  IS AN ABORT STAGE COMMANDED?
                READ    CHAN30
                COM
                TS      L
                MASK    BIT4
                CCS     A
                TCF     P71A            # YES.

P70NOW?         CS      MODREG          # NO.  ARE WE IN P70 NOW?
                AD      1DEC70
                EXTEND
                BZF     LANDISP         # YES.  PROCEED TO R10.

                CA      L               # NO.  IS AN ABORT COMMANDED?
                MASK    BIT1
                CCS     A
                TCF     P70A            # YES.
                TCF     LANDISP         # NO.  PROCEED TO R10.

                COUNT*  $$/P70

P70             TC      LEGAL?
P70A            CS      ZERO
                TCF     +3
P71             TC      LEGAL?
P71A            CAF     TWO
 +3             TS      Q
                INHINT
                EXTEND
                DCA     CNTABTAD
                DTCB

                EBANK=  DVCNTR
CNTABTAD        2CADR   CONTABRT

1DEC70          DEC     70
1DEC71          DEC     71

                BANK    05
                SETLOC  ABORTS1
                BANK
                COUNT*  $$/P70

CONTABRT        CAF     ABRTJADR
                TS      BRUPT
                RESUME
```

```
ABRTJADR        TCF     ABRTJASK

ABRTJASK        CAF     OCTAL27
                AD      Q
                TS      L
                COM
                DXCH    -PHASE4
                INDEX   Q
                CAF     MODE70
                TS      MODREG

                TS      DISPDEX         # INSURE DISPDEX IS POSITIVE.

                CCS     Q               # SET APSFLAG IF P71.
                CS      FLGWRD10        # SET APSFLAG PRIOR TO THE ENEMA.
                MASK    APSFLBIT
                ADS     FLGWRD10
                CS      DAPBITS         # DAPBITS = OCT 640 = BITS 6, 8, 9
                MASK    DAPBOOLS        # (TURN OFF: ULLAGE, DRIFT, AND XOVINHIB )
                TS      DAPBOOLS

                CS      FLAGWRD5        # SET ENGONFLG.
                MASK    ENGONBIT
                ADS     FLAGWRD5

                CS      PRIO30          # INSURE THAT THE ENGINE IS ON, IF ARMED.
                EXTEND
                RAND    DSALMOUT
                AD      BIT13
                EXTEND
                WRITE   DSALMOUT

                CAF     LRBYBIT         # TERMINATE R12.
                TS      FLGWRD11

                CS      FLAGWRD0        # SET R10FLAG TO SUPPRESS OUTPUTS TO THE
                MASK    R10FLBIT        # CROSS-POINTER DISPLAY.
                ADS     FLAGWRD0        # THE FOLLOWING ENEMA WILL REMOVE THE
                                        # DISPLAY INERTIAL DATA OUTBIT.

                TC      CLRADMOD        # INSURE RADMODES PROPERLY SET FOR R29.

                EXTEND                  # LOAD TEVENT FOR THE DOWNLINK.
                DCA     TIME2
                DXCH    TEVENT

                EXTEND
                DCA     SVEXITAD
                DXCH    AVGEXIT
```

```
                EXTEND
                DCA     NEGO
                DXCH    -PHASE1

                EXTEND
                DCA     NEGO
                DXCH    -PHASE3

                EXTEND
                DCA     NEGO
                DXCH    -PHASE6

                CAF     THREE           # SET UP 4.3SPOT FOR GOABORT
                TS      L
                COM
                DXCH    -PHASE4

                CAF     OCT37774        # SET T5RUPT TO CALL DAPIDLER IN
                TS      TIME5           # 40 MILLISECONDS.

                TC      POSTJUMP
                CADR    ENEMA

                EBANK=  DVCNTR
SVEXITAD        2CADR   SERVEXIT

MODE70          DEC     70
OCTAL27         OCT     27
MODE71          DEC     71

DAPBITS         OCT     00640

                BANK    32
                SETLOC  ABORTS
                BANK

                COUNT*  $$/P70

GOABORT         TC      INTPRET
                CALL
                        INITCDUW
                EXIT
                CAF     FOUR
                TS      DVCNTR

                CAF     WHICHADR
                TS      WHICH

                TC      DOWNFLAG
                ADRES   FLRCS
```

```
                TC      DOWNFLAG
                ADRES   FLUNDISP

                TC      DOWNFLAG
                ADRES   IDLEFLAG

                TC      UPFLAG          # INSURE 4-JET TRANSLATION CAPABILITY.
                ADRES   ACC4-2FL

                TC      CHECKMM
70DEC           DEC     70
                TCF     P71RET

P70INIT         TC      INTPRET
                CALL
                        TGOCOMP
                DLOAD   SL
                        MDOTDPS
                        4D
                BDDV
                        MASS
                STODL   TBUP
                        MASS
                DDV     SR1
                        K(1/DV)
                STORE   1/DV1
                STORE   1/DV2
                STORE   1/DV3
                BDDV
                        K(AT)
                STODL   AT
                        DTDECAY
                DCOMP   SL
                        11D
                STORE   TTO
                SLOAD   DCOMP
                        DPSVEX
                SR2
                STORE   VE              # INITIALIZE DPS EXHAUST VELOCITY
                SET     CALL
                        FLAP
                        COMMINIT
                AXC,1   GOTO            # RETURN HERE IN P70, SE X1 FOR DPS COEFF.
                        OD
                        BOTHPOLY
INJTARG         AXC,1                   # RETURN HERE IN P71, SET X1 FOR APS COEFF
                        8D
BOTHPOLY        DLOAD*  DMP             # TGO D
                        ABTCOF,1
                        TGO
```

```
                DAD*    DMP
                        ABTCOF +2,1     # TGO(C+TGO D)
                        TGO
                DAD*    DMP
                        ABTCOF +4,1     # TGO(B+TGO(C+TGO D))
                        TGO
                DAD*
                        ABTCOF +6,1     # A+TGO(B+TGO(C+TGO D))
                STORE   ZDOTD           # STORE TENTATIVELY IN ZDOTD
                DSU     BPL             # CHECK AGAINST MINIMUM
                        VMIN
                        UPRATE          # IF BIG ENOUGH, LEAVE ZDOTD AS IS .
                DLOAD
                        VMIN
                STORE   ZDOTD           # IF TOO SMALL, REPLACE WITH MINIMUM.
UPRATE          DLOAD
                        ABTRDOT
                STCALL  RDOTD           # INITIALZE RDOTD.
                        YCOMP           # COMPUTE Y
                ABS     DSU
                        YLIM            # /Y/-DYMAX
                BMN     SIGN            # IF <0, XR<.5DEG, LEAVE YCO AT 0
                        YOK             # IF >0, FIX SIGN OF DEFICIT, THIS IS YCO.
                        Y
                STORE   YCO
YOK             DLOAD   DSU
                        YCO
                        Y               # COMPUTE XRANGE IN CASE ASTRONAUT WANTS
                SR
                        5D
                STORE   XRANGE          # TO LOOK.
UPTHROT         SET     EXIT
                        FLVR

                TC      UPFLAG          # SET ROTFLAG
                ADRES   ROTFLAG

                TC      THROTUP

                TC      PHASCHNG
                OCT     04024

 -3             TC      BANKCALL        # VERIFY THAT THE PANEL SWITCHES
                CADR    P40AUTO         # ARE PROPERLY SET.

                TC      THROTUP

UPTHROT1        EXTEND                  # SET SERVICER TO CALL ASCENT GUIDANCE.
                DCA     ATMAGAD
                DXCH    AVGEXIT
```

```
GRP4OFF         TC      PHASCHNG        # TERMINATE USE OF GROUP 4.
                OCT     00004

                TCF     ENDOFJOB

P71RET          TC      DOWNFLAG
                ADRES   LETABORT

                CAF     THRESH2         # SET DVMON THRESHOLD TO THE ASCENT VALUE.
                TS      DVTHRUSH

                TC      INTPRET
                BON     CALL
                        FLAP
                        OLDTIME
                        TGOCOMP         # IF FLAP=0, TGO=T-TIG
                SSP     GOTO
                        QPRET
                CADR    INJTARG
                        P12INIT         # WILL EXIT P12INIT TO INJTARG
OLDTIME         DLOAD   SL1             # IF FLAP=1,GTO=2 TGO
                        TGO
                STCALL  TGO1
                        P12INIT
                EXIT
                TC      PHASCHNG
                OCT     04024

                EXTEND
                DCA     TGO1
                DXCH    TGO
                TCF     UPTHROT1 -3

TGO1            =       VGBODY
# ************************************************************************

                BANK    21
                SETLOC  R11
                BANK

                COUNT*  $$/P70

LEGAL?          CS      MMNUMBER        # IS THE DESIRED PGM ALREADY IN PROGRESS?
                AD      MODREG
                EXTEND
                BZF     ABORTALM

                CS      FLAGWRD9        # ARE THE ABORTS ENABLED?
                MASK    LETABBIT
                CCS     A
```

```
                TCF     ABORTALM

                CA      FLAGWRD7        # IS SERVICER ON THE AIR?
                MASK    AVEGFBIT
                CCS     A
                TC      Q               # YES.  ALL IS WELL.
ABORTALM        TC      FALTON
                TC      RELDSP
                TC      POSTJUMP
                CADR    PINBRNCH

                BANK    32
                SETLOC  ABORTS
                BANK

                COUNT*  $$/P70

# ************************************************************************

TGOCOMP         RTB     DSU
                        LOADTIME
                        TIG
                SL
                        11D
                STORE   TGO
                RVQ

# ************************************************************************

THROTUP         CAF     BIT13
                TS      THRUST
                CAF     BIT4
                EXTEND
                WOR     CHAN14
                TC      Q

# ************************************************************************

10SECS          2DEC    1000

HINJECT         2DEC    18288 B-24      # 60,000 FEET EXPRESSED IN METERS.

(TGO)A          2DEC    37000 B-17

K(AT)           2DEC    .02             # SCALING CONSTANT

WHICHADR        REMADR  ABRTABLE

# ************************************************************************
```

```
                EBANK=  DVCNTR
ATMAGAD         2CADR   ATMAG

ORBMANAD        ADRES   ORBMANUV
```

```
                BANK     24
                SETLOC   P12
                BANK

                EBANK=   DVCNTR
                COUNT*   $$/P12

P12LM           TC       PHASCHNG
                OCT      04024

                TC       BANKCALL
                CADR     R02BOTH          # CHECK THE STATUS OF THE IMU.

                TC       UPFLAG
                ADRES    MUNFLAG

                TC       UPFLAG           # INSURE 4-JET TRANSLATION CAPABILITY.
                ADRES    ACC4-2FL

                TC       UPFLAG           # PREVENT R10 FROM ISSUING CROSS-POINTER
                ADRES    R10FLAG          # OUTPUTS.

                TC       CLRADMOD         # INITIALIZE RADMODES FOR R29.

                TC       DOWNFLAG         # CLEAR RENDEZVOUS FLAG FOR P22
                ADRES    RNDVZFLG

                CAF      THRESH2          # INITIALIZE DVMON
                TS       DVTHRUSH
                CAF      FOUR
                TS       DVCNTR

                CA       ZERO
                TS       TRKMKCNT         # SHOW THAT R29 DOWNLINK DATA ISN'T READY.
                CAF      V06N33A
                TC       BANKCALL         # FLASH TIG
                CADR     GOFLASH
                TCF      GOTOPOOH
                TCF      +2               # PROCEED
                TCF      -5               # ENTER

                TC       PHASCHNG
                OCT      04024

                TC       INTPRET
                CALL                      # INITIALZE WM AND /LAND/
                         GUIDINIT
                SET      CALL
                         FLPI
                         P12INIT
```

```
P12LMB          DLOAD
                        (TGO)A          # SET TGO TO AN INITIAL NOMINAL VALUE.
                STODL   TGO
                        TIG
                STCALL  TDEC1
                        LEMPREC         # ROTATE THE STATE VECTORS TO THE
                VLOAD   MXV             # IGNITION TIME.
                        VATT
                        REFSMMAT
                VSL1
                STOVL   V1S             # COMPUTE V1S = VEL(TIG)*2(-7)M/CS.
                        RATT
                MXV     VSL6
                        REFSMMAT
                STCALL  R               # COMPUTE R = POS(TIG)*2(-24)M.
                        MUNGRAV         # COMPUTE GDT1/2(TIG)*2(-7)M/CS.
                VLOAD   UNIT
                        R
                STCALL  UNIT/R/         # COMPUTE UNIT/R/ FOR YCOMP.
                        YCOMP
                SR      DCOMP
                        5D
                STODL   XRANGE          # INITIALIZE XRANGE FOR NOUN 76.
                        VINJNOM
                STODL   ZDOTD
                        RDOTDNOM
                STORE   RDOTD
                EXIT

                TC      PHASCHNG
                OCT     04024

NEWLOAD         CAF     V06N76          # FLASH CROSS-RANGE, AND APOLUNE VALUES.
                TC      BANKCALL
                CADR    GOFLASH
                TCF     GOTOPOOH
                TCF     +2              # PROCEED
                TCF     NEWLOAD         # ENTER NEW DATA.

                CAF     P12ADRES
                TS      WHICH

                TC      PHASCHNG
                OCT     04024

                TC      INTPRET
                DLOAD   SL
                        XRANGE
                        5D
                DAD
```

```
                                Y
                STOVL           YCO
                                UNIT/R/
                VXSC            VAD
                                49FPS
                                V1S
                STORE           V               # V(TIPOVER) = V(IGN) + 57FPS (UNIT/R/)
                DOT             SL1
                                UNIT/R/
                STOVL           RDOT            # RDOT * 2(-7)
                                UNIT/R/
                VXV             UNIT
                                QAXIS
                STORE           ZAXIS1
                SETGO
                                FLVR
                                ASCENT
P12RET          DLOAD
                                ATP             # ATP(2)*2(18)
                DSQ             PDDL
                                ATY             # ATY(2)*2(18)
                DSQ             DAD
                BZE             SQRT
                                YAWDUN
                SL1             BDDV
                                ATY
                ARCSIN
YAWDUN          STOVL           YAW
                                UNFC/2
                UNIT            DOT
                                UNIT/R/
                SL1             ARCCOS
                DCOMP
                STORE           PITCH
                EXIT
                TC              PHASCHNG
                OCT             04024

                TC              DOWNFLAG
                ADRES           FLPI

                INHINT
                TC              IBNKCALL
                CADR            PFLITEDB
                RELINT

                TC              POSTJUMP
                CADR            BURNBABY

P12INIT         DLOAD                           # INITIALIZE ENGINE DATA.  USED FOR P12 AND
```

```
                                (1/DV)A         # P71.
                STORE           1/DV3
                STORE           1/DV2
                STODL           1/DV1
                                (AT)A
                STODL           AT
                                (TBUP)A
                STODL           TBUP
                                ATDECAY
                DCOMP           SL
                                11D
                STORE           TTO
                SLOAD           DCOMP
                                APSVEX
                SR2
                STORE           VE
                BOFF            RVQ
                                FLAP
                                COMMINIT
COMMINIT        DLOAD           DAD             # INITIALIZE TARGET DATA.  USED BY P12, P70
                                HINJECT         # AND P71 IF IT DOES NOT FOLLOW P70.
                                /LAND/
                STODL           RCO
                                HI6ZEROS
                STORE           TXO
                STORE           YCO
                STORE           RDOTD
                STOVL           YDOTD
                                VRECTCSM
                VXV             MXV
                                RRECTCSM
                                REFSMMAT
                UNIT
                STORE           QAXIS
                RVQ
P12ADRES        REMADR          P12TABLE

                SETLOC          P12A
                BANK
                COUNT*          $$/P12

GUIDINIT        STQ             SETPD
                                TEMPR60
                                OD
                VLOAD           PUSH
                                UNITZ
                RTB             PUSH
                                LOADTIME
                CALL
                                RP-TO-R
```

```
                MXV     VXSC
                        REFSMMAT
                        MOONRATE
                STOVL   WM
                        RLS
                ABVAL   SL3
                STCALL  /LAND/
                        TEMPR60

49FPS           2DEC    .149352 B-6     # EXPECTED RDOT AT TIPOVER
VINJNOM         2DEC    16.7924 B-7     # 5509.5 FPS(APO=30NM WITH RDOT=19.5FPS)
RDOTDNOM        2DEC    .059436 B-7     # 19.5 FPS
```

```
                BANK    34
                SETLOC  ASCFILT
                BANK

                EBANK=  DVCNTR

                COUNT*  $$/ASENT

ATMAG           TC      PHASCHNG
                OCT     00035
                TC      INTPRET
                BON
                        FLRCS
                        ASCENT
                DLOAD   DSU
                        ABDVCONV
                        MINABDV
                BMN     CLEAR
                        ASCTERM4
                        SURFFLAG
                CLEAR   SLOAD
                        RENDWFLG
                        BIT3H
                DDV     EXIT
                        ABDVCONV
                DXCH    MPAC
                DXCH    1/DV3
                DXCH    1/DV2
                DXCH    1/DV1
                DXCH    1/DV0
                TC      INTPRET
                DLOAD   DAD
                        1/DV0
                        1/DV1
                DAD     DAD
                        1/DV2
                        1/DV3
                DMP     DMP
                        VE
                        2SEC(9)
                SL3     PDDL
                        TBUP
                SR1     DAD
                DSU
                        6SEC(18)
                STODL   TBUP
                        VE
                SR1     DDV
                        TBUP
                STCALL  AT
```

```
                                 ASCENT
        BIT3H           OCT      4
```

```
                BANK    30
                SETLOC  ASENT
                BANK
                COUNT*  $$/ASENT

ASCENT          VLOAD   ABVAL
                        R
                STOVL   /R/MAG
                        ZAXIS1
                DOT     SL1
                        V               # Z.V = ZDOT*2(-8).
                STOVL   ZDOT            # ZDOT*2(-7)
                        ZAXIS1
                VXV     VSL1
                        UNIT/R/         # Z X UR = LAXIS*2(-2)
                STORE   LAXIS           # LAXIS*2(-1)
                DOT     SL1
                        V               # L.V = YDOT*2(-8).
                STCALL  YDOT            # YDOT * 2(-7)
                        YCOMP
                VLOAD
                        GDT1/2          # LOAD GDT1/2*2(-7) M/CS.
                V/SC    DOT
                        2SEC(18)
                        UNIT/R/         # G.UR*2(9) = GR*2(9).
                PDVL    VXV             # STORE IN PDL(0)
                        UNIT/R/         # LOAD UNIT/R/ *2(-1)
                        V               # UR*2(-1) X V*2(-7) = H/R*2(-8).
                VSQ     DDV             # H(2)/R(2)*2(-16).
                        /R/MAG          # H(2)/R(3)*2(9).
                SL1     DAD
                STADR
                STODL   GEFF            # GEFF*2(10)M/CS/CS.
                        ZDOTD
                DSU
                        ZDOT
                STORE   DZDOT           # DZDOT = (ZDOTD - ZDOT) * 2(7) M/CS.
                VXSC    PDDL
                        ZAXIS1
                        YDOTD
                DSU
                        YDOT
                STORE   DYDOT           # DYDOT = (YDOTD - YDOT) *2(7) M/CS.
                VXSC    PDDL
                        LAXIS
                        RDOTD
```

```
                DSU
                        RDOT
                STORE   DRDOT           # DRDOT = (RDOTD - RDOT) * 2(7) M/CS.
                VXSC    VAD
                        UNIT/R/
                VAD     VSL1
                STADR
                STORE   VGVECT          # VG = (DRDOT)R + (DVDOT)L + (DZDOT)Z.
                DLOAD   DMP             # LOAD TGO
                        TGO             # TGO GEFF
                        GEFF
                VXSC    VSL1
                        UNIT/R/         # TGO GEFF UR
                BVSU
                        VGVECT          # COMPENSATED FOR GEFF
                STORE   VGVECT          # STORE FOR DOWNLINK
                MXV     VSL1            # GET VGBODY FOR N85 DISPLAY
                        XNBPIP
                STOVL   VGBODY
                        VGVECT
                ABVAL   BOFF            # MAGNITUDE OF VGVECT
                        FLRCS           # IF FLRCS=0,DO NORMAL GUIDANCE
                        MAINENG
                DDV                     # USE TGO=VG/AT WITH RCS
                        AT/RCS
                STCALL  TGO             # THIS WILL BE USED ON NEXT CYCLE
                        ASCTERM2
MAINENG         DDV     PUSH            # VG/VE IN PDL(0)               (2)
                        VE
                DMP     BDSU            # 1 - KT VG/VE
                        KT1
                        NEARONE
                DMP     DMP             # TBUP VG(1-KT VG/VE)/VE        (0)
                        TBUP            #       = TGO
                DSU                     # COMPENSATE FOR TAILOFF
                        TTO
                STORE   TGO
                SR      DCOMP
                        11D
                STODL   TTOGO           # TGO *2(-28) CS
                        TGO
                BON     DSU
                        IDLEFLAG
                        T2TEST
                        4SEC(17)        # ( TGO - 4 )*2(-17) CS.
                BMN
                        ENGOFF
T2TEST          DLOAD
                        TGO
                DSU     BMN             # IF TGO - T2 NEG., GO TO CMPONENT
```

```
                        T2A
                        CMPONENT
                DLOAD   DSU
                        TBUP
                        TGO
                DDV     CALL            # 1- TGO/TBUP
                        TBUP
                        LOGSUB
                SL      PUSH            # -L IN PDL(0)                  (2)
                        5
                BDDV    BDSU            # -TGO/L*2(-17)
                        TGO
                        TBUP            # TBUP + TGO/L = D12*2(-17)
                PUSH    BON             # STORE IN PDL(2)               (4)
                        FLPC            # IF FLPC = 1, GO TO CONST
                        NORATES
                DLOAD   DSU
                        TGO
                        T3
                BPL     SET             # FLPC=1
                        RATES
                        FLPC
NORATES         DLOAD
                        HI6ZEROS
                STORE   PRATE           # B = 0
                STORE   YRATE           # D = 0
                GOTO
                        CONST           # GO TO CONST
RATES           DLOAD   DSU
                        TGO
                        02D             # TGO - D12 = D21*2(-17)
                PUSH    SL1             # IN PDL(4)                     (6)
                BDSU    SL3             # (1/2TGO - D21)*2(-13) = E * 2(-13)
                        TGO             #                               (8)
                PDDL    DMP             # IN PDL(6)
                        TGO
                        RDOT            # RDOT TGO * 2(-24)
                DAD     DSU             # R + RDOT TGO
                        /R/MAG          # R + RDOT TGO - RCO
                        RCO             # MPAC = -DR *2(-24).
                PDDL    DMP             # -DR IN PDL(8)                 (10)
                        DRDOT
                        04D             # D21 DRDOT*2(-24)
                DAD     SL2             # (D21 DRDOT-DR)*2(-22)         (8)
                DDV     DDV
                        06D             # (D21 DRDOT-DR)/E*2(-9)
                        TGO
                STORE   PRATE           # B * 2(8)
                BMN     DLOAD           # B>0 NOT PERMITTED
                        CHKBMAG
```

```
                          HI6ZEROS
                STCALL    PRATE
                          PROK
CHKBMAG         SR4       DDV               # B*2(4)
                          TBUP              # (B / TAU) * 2(21)
                DSU       BPL
                          PRLIMIT           # ( B / TAU ) = 2(21) MAX.
                          PROK
                DLOAD     DMP
                          PRLIMIT
                          TBUP              # B MAX. * 2(4)
                SL4                         # BMAX*2(8)
                STORE     PRATE
PROK            DLOAD
                          TGO
                DMP       DAD               # YDOT TGO
                          YDOT
                          Y                 # Y + YDOT TGO
                DSU       PDDL              # Y + YDOT TGO - YCO
                          YCO               # MPAC = - DY*(-24.) IN PDL(8)  (10)
                          DYDOT
                DMP       DAD               # D21 DYDOT - DY                (8)
                          04D
                SL2       DDV               # (D21 DYDOT - DY)/E*2(-9)
                DDV       SETPD             # (D21 DYDOT - DY)/E TGO*2(8)
                          TGO               #        = D*2(8)
                          04
                STORE     YRATE
CONST           DLOAD     DMP               # LOAD B*2(8)
                          PRATE             # B D12*2(-9)
                          02D
                PDDL      DDV               # D12 B IN PDL(4)       (6)
                          DRDOT             # LOAD DRDOT*2(-7)
                          00D               # -DRDOT/L*2(-7)
                SR2       DSU               # (-DRDOT/L-D12 B)=A*2(-9)      (4)
                STADR
                STODL     PCONS
                          YRATE             # D*2(8)
                DMP       PDDL              # D12 D,EXCH WITH -L IN PDL(0)  (2,2)
                BDDV      SR2               # -DYDOT/L*2(-9)
                          DYDOT
                DSU                         # (-DYDOT/L-D12 D)=C*2(-9)
                          00D
                STORE     YCONS
CMPONENT        SETPD     DLOAD
                          00D
                          100CS
                DMP
                          PRATE             # B(T-T0)*2(-9)
                DAD       DDV               # (A+B(T-T0))*2(-9)
```

```
                        PCONS           # (A+B(T-TO))/TBUP*2(8)
                        TBUP
                SL1     DSU
                        GEFF            # ATR*2(9)
                STODL   ATR
                        100CS
                DMP     DAD
                        YRATE
                        YCONS           # (C+D(T-TO))*2(-9)
                DDV     SL1
                        TBUP
                STORE   ATY             # ATY*2(9)
                VXSC    PDDL            # ATY UY*2(8)           (6)
                        LAXIS
                        ATR
                VXSC    VAD
                        UNIT/R/
                VSL1    PUSH            # AH*2(9) IN PDL(0)     (6)
                ABVAL   PDDL            # AH(2) IN PDL(34)
                        AT              # AHMAG IN PDL(6)       (8)
                DSQ     DSU             # (AT(2)-AH(2))*2(18)
                        34D             # =ATP2*2(18)
                PDDL    PUSH            #                       (12)
                        AT
                DSQ     DSU             # (AT(2)KR(2)-AH(2))*2(18)      (10)
                        34D             # =ATP3*2(18)
                BMN     DLOAD           # IF ATP3 NEG,GO TO NO-ATP
                        NO-ATP          # LOAD ATP2, IF ATP3 POS
                        8D
                SQRT    GOTO            # ATP*2(9)
                        AIMER
NO-ATP          DLOAD   BDDV            # KR AT/AH = KH         (8)
                        6D
                VXSC                    # KH AG*2(9)
                        00D
                STODL   00D             # STORE NEW AH IN PDL(0)
                        HI6ZEROS
AIMER           SIGN
                        DZDOT
                STORE   ATP
                VXSC
                        ZAXIS1          # ATP ZAXIS *2(8).
                VSL1    VAD             # AT*2(0)
                        00D
                STORE   UNFC/2          # WILL BE OVERWRITTEN IF IN VERT. RISE.
                SETPD   BON
                        00D
                        FLPI
                        P12RET
                BON
```

```
                                FLVR
                                CHECKALT
        MAINLINE        VLOAD   VCOMP
                                UNIT/R/
                        STODL   UNWC/2
                                TXO
                        DSU     BPL
                                PIPTIME
                                ASCTERM
                        BON
                                ROTFLAG
                                ANG1CHEK
        CLRXFLAG        CLEAR   CLEAR
                                NOR29FLG        # START R29 IN ASCENT PHASE.
                                XOVINFLG        # ALLOW X-AXIS OVERRIDE
        ASCTERM         EXIT
                        CA      FLAGWRD9
                        MASK    FLRCSBIT
                        CCS     A
                        TCF     ASCTERM3
                        TC      INTPRET
                        CALL
                                FINDCDUW -2
        ASCTERM1        EXIT
         +1             CA      FLAGWRD9        # INSURE THAT THE NOUN 63 DISPLAY IS
                        MASK    FLRCSBIT        # BYPASSED IF WE ARE IN THE RCS TRIMMING
                        CCS     A               # MODE OF OPERATION
                        TCF     ASCTERM3
                        CA      FLAGWRD8        # BYPASS DISPLAYS IF ENGINE FAILURE IS
                        MASK    FLUNDBIT        # INDICATED.
                        CCS     A
                        TCF     ASCTERM3
                        CAF     V06N63*
                        TC      BANKCALL
                        CADR    GODSPR
                        TCF     ASCTERM3
        ASCTERM2        EXIT
        ASCTERM3        TCF     ENDOFJOB
        ASCTERM4        EXIT
                        INHINT
                        TC      IBNKCALL        # NO GUIDANCE THIS CYCLE -- HENCE ZERO
                        CADR    ZATTEROR        # THE DAP COMMANDED ERRORS.
                        TCF     ASCTERM1 +1

        CHECKALT        DLOAD   DSU
                                /R/MAG
                                /LAND/
                        DSU     BMN             # IF H LT 25K CHECK Z AXIS ORIENTATION
                                25KFT
                                CHECKYAW
```

```
EXITVR          CLEAR   BON
                        FLVR
                        ROTFLAG
                        MAINLINE
                DLOAD   DAD
                        PIPTIME
                        10SECS
                STCALL  TXO
                        MAINLINE
EXITVR1         CLRGO
                        ROTFLAG
                        EXITVR

                SETLOC  ASENT1
                BANK
                COUNT*  $$/ASENT

ANG1CHEK        VLOAD   DOT
                        UNFC/2
                        XNBPIP
                DSU     BPL
                        COSTHET1
                        OFFROT
                VLOAD   DOT
                        XNBPIP
                        UNIT/R/
                DSU     BMN
                        COSTHET2
                        KEEPVR1
OFFROT          CLRGO
                        ROTFLAG
                        CLRXFLAG

                BANK    7
                SETLOC  ASENT2
                BANK
                COUNT*  $$/ASENT

SETXFLAG        =       CHECKYAW

CHECKYAW        SET
                        XOVINFLG        # PROHIBIT X-AXIS OVERRIDE
                DLOAD   VXSC
                        ATY
                        LAXIS
                PDDL    VXSC
                        ATP
                        ZAXIS1
                VAD     UNIT
                PUSH    DOT
```

```
                        YNBPIP
                ABS     DSU
                        SIN5DEG
                BPL     DLOAD
                        KEEPVR
                        RDOT
                DSU     BPL
                        40FPS
                        EXITVR1
                GOTO
                        KEEPVR

                BANK    5
                SETLOC  ASENT3
                BANK
                COUNT*  $$/ASENT

SIN5DEG         2DEC    0.08716 B-2
40FPS           2DEC    0.12192 B-7

                BANK    14
                SETLOC  ASENT4
                BANK
                COUNT*  $$/ASENT

KEEPVR          VLOAD   STADR           # RECALL LOSVEC FROM PUSHLIST
                STORE   UNWC/2
KEEPVR1         VLOAD
                        UNIT/R/
                STCALL  UNFC/2
                        ASCTERM

ENGOFF          RTB
                        LOADTIME
                DSU     DAD
                        PIPTIME
                        TTOGO
                DCOMP   EXIT
                TC      TPAGREE         # FORCE SIGN AGREEMENT ON MPAC, MPAC +1.
                CAF     EBANK7
                TS      EBANK
                EBANK=  TGO
                INHINT
                CCS     MPAC +1
                TCF     +3              # C(A) = DT - 1 BIT
                TCF     +2              # C(A) = 0
                CAF     ZERO            # C(A) = 0
                AD      BIT1            # C(A) = 1 BIT OR DT.
```

```
                TS      ENGOFFDT
                TC      TWIDDLE
                ADRES   ENGOFF1
                TC      PHASCHNG
                OCT     47014
                -GENADR ENGOFFDT
                EBANK=  TGO
                2CADR   ENGOFF1

                TC      INTPRET
                SET     GOTO
                        IDLEFLAG        # DISABLE DELTA-V MONITOR
                        T2TEST

ENGOFF1         TC      IBNKCALL        # SHUT OFF THE ENGINE.
                CADR    ENGINOF2

                CAF     PRIO17          # SET UP A JOB FOR THE ASCENT GUIDANCE
                TC      FINDVAC         # POSTBURN LOGIC.
                EBANK=  WHICH
                2CADR   CUTOFF

                TC      PHASCHNG
                OCT     07024
                OCT     17000
                EBANK=  TGO
                2CADR   CUTOFF

                TCF     TASKOVER

CUTOFF          TC      UPFLAG          # SET FLRCS FLAG.
                ADRES   FLRCS

 -5             CAF     V16N63
                TC      BANKCALL
                CADR    GOFLASH
                TCF     +3
                TCF     CUTOFF1
                TCF     -5

 +3             TC      POSTJUMP
                CADR    TERMASC

CUTOFF1         INHINT
                TC      IBNKCALL        # ZERO ATTITUDE ERRORS BEFORE REDUCINT DB.
                CADR    ZATTEROR
                TC      IBNKCALL
                CADR    SETMINDB
                TC      POSTJUMP
                CADR    CUTOFF2
```

```
V16N63          VN      1663
                BANK    30
                SETLOC  ASENT5
                BANK
                COUNT*  $$/ASENT

CUTOFF2         TC      PHASCHNG
                OCT     04024

                CAF     V16N85C
                TC      BANKCALL
                CADR    GOFLASH
                TCF     TERMASC
                TCF     +2              # PROCEED
                TCF     CUTOFF2

TERMASC         TC      PHASCHNG
                OCT     04024

                INHINT                  # RESTORE DEADBAND DESIRED BY ASTRONAUT.
                TC      IBNKCALL
                CADR    RESTORDB
                TC      DOWNFLAG        # DISALLOW ABORTS AT THIS TIME.
                ADRES   LETABORT
                TCF     GOTOPOOH

V16N85C         VN      1685

                BANK 27
                SETLOC  ASENT1
                BANK
                COUNT* $$/ASENT

YCOMP           VLOAD   DOT
                        UNIT/R/
                        QAXIS
                SL2     DMP
                        RCO
                STORE   Y
                RVQ

                BANK    30
                SETLOC  ASENT
                BANK
```

```
100CS           EQUALS  2SEC(18)
T2A             EQUALS  2SEC(17)
4SEC(17)        2DEC    400 B-17
2SEC(17)        2DEC    200 B-17
T3              2DEC    1000 B-17
6SEC(18)        2DEC    600 B-18
BIT4H           OCT     10
2SEC(9)         2DEC    200 B-9
V06N63*         VN      0663
V06N76          VN      0676
V06N33A         VN      0633

                BANK    33
                SETLOC  ASENT6
                BANK
                COUNT*  $$/ASENT

KT1             2DEC    0.5000
PRLIMIT         2DEC    -.0639          # (B/TBUP)MIN=-.1FT.SEC(-3)
MINABDV         2DEC    .0356 B-5       # 10 PERCENT BIGGER THAN GRAVITY
1/DV0           =       MASS1
```

```
# THE LOGARITHM SUBROUTINE

                BANK    24
                SETLOC  FLOGSUB
                BANK

# INPUT ..... X IN MPAC
# OUTPUT ..... -LOG(X) IN MPAC

LOGSUB          NORM    BDSU
                        MPAC +6
                        NEARONE
                EXIT
                TC      POLY
                DEC     6
                2DEC    .0000000060
                2DEC    -.0312514377
                2DEC    -.0155686771
                2DEC    -.0112502068
                2DEC    -.0018545108
                2DEC    -.0286607906
                2DEC    .0385598563
                2DEC    -.0419361902

                CAF     ZERO
                TS      MPAC +2
                EXTEND
                DCA     CLOG2/32
                DXCH    MPAC
                DXCH    BUF +1
                CA      MPAC +6
                TC      SHORTMP
                DXCH    MPAC +1
                DXCH    MPAC
                DXCH    BUF +1
                DAS     MPAC
                TC      INTPRET
                DCOMP   RVQ

CLOG2/32        2DEC    .0216608494
```

```
                BANK    37
                SETLOC  SERV1
                BANK

                EBANK=  DVCNTR

# ************* PREREAD *******************

                COUNT*  $$/SERV

PREREAD         CAF     SEVEN           # 5.7 SPOT TO SKIP LASTBIAS AFTER
                TC      GNUFAZE5        # RESTART.
                CAF     PRIO21
                TC      NOVAC
                EBANK=  NBDX
                2CADR   LASTBIAS        # DO LAST GYRO COMPENSATION IN FREE FALL

BIBIBIAS        TC      PIPASR +3       # CLEAR + READ PIPS LAST TIME IN FRE5+F133
                                        # DO NOT DESTROY VALUE OF PIPTIME1

                CS      FLAGWRD7
                MASK    SUPER011        # SET V37FLAG AND AVEGFLAG (BITS 5 AND 6
                ADS     FLAGWRD7        #       OF FLAGWRD7)

                CS      DRFTBIT
                MASK    FLAGWRD2        # RESET DRIFTFLAG
                TS      FLAGWRD2

                CAF     FOUR            # INITIALIZE DV MONITOR
                TS      PIPAGE

                CAF     ENDJBCAD        # POINT OUTROUTE TO END-OF-JOB.
                TS      OUTROUTE

                CAF     PRIO22
                TC      FINDVAC         # TO FIRST ENTRY TO AVERAGE G
                EBANK=  DVCNTR
                2CADR   NORMLIZE

                CA      TWO             # 5.2SPOT FOR REREADAC AND NORMLIZE
GOREADAX        TC      GNUTFAZ5
                CA      2SECS           # WAIT TWO SECONDS FOR READACCS
                TC      VARDELAY
```

```
# ************* READACCS ****************

READACCS        CS      OCT37771        # THIS PIECE OF CODING ATTEMPTS TO
                AD      TIME5           # SYNCHRONIZE READACCS WITH THE DIGITAL
                CCS     A               # AUTOPILOT SO THAT A PAXIS RUPT WILL
                CS      ONE             # OCCUR APPROXIMATELY 70 MILLISECONDS
                TCF     +2              # FOLLOWING THE READACCS RUPT.  THE 70 MS
                CA      ONE             # OFFSET WAS CHOSEN SO THAT THE PAXIS
 +2             ADS     TIME5           # RUPT WOULD NOT OCCUR SIMULTANEOUSLY
                                        # WITH ANY OF THE 8 SUBSEQUENT R10,R11
                                        # INTERRUPTS -- THUS MINIMIZING THE POSS-
                                        # IBILITY OF LOSING DOWNRUPTS.

                TC      PIPASR          # READ THE PIPAS.

PIPSDONE        CA      FIVE
                TC      GNUFAZE5
REDO5.5         CAF     ONE
                TS      PIPAGE

                CA      PRIO20
                TC      FINDVAC
                EBANK=  DVCNTR
                2CADR   SERVICER        # SET UP SERVICER JOB

                CA      BIT9
                EXTEND
                WOR     DSALMOUT        # TURN ON TEST CONNECTOR OUTBIT

                CA      FLAGWRD7
                MASK    AVEGFBIT
                EXTEND
                BZF     AVEGOUT         # AVEGFLAG DOWN -- SET UP FINAL EXIT

                CA      FLAGWRD6
                MASK    MUNFLBIT
                EXTEND
                BZF     MAKEACCS        # MUNFLAG CLEAR -- BYPASS LR AND DISP.

                CCS     PHASE2
                TCF     MAKEACCS        # PHASE 2 ACTIVATED -- AVOID MULTIPLE R10.

                CAF     SEVEN           # SET PIPCTR FOR 4X/SEC RATE.
                TS      PIPCTR

                CS      TIME1           # SET TBASE2 .05 SECONDS IN THE PAST.
                AD      FIVE
                AD      NEG1/2
                AD      NEG1/2
                XCH     TBASE2
```

```
                CAF     DEC17           # 2.21SPOT FOR R10,R11
                TS      L
                COM
                DXCH    -PHASE2

                CAF     OCT24           # FIRST R10,R11 IN .200 SECONDS
                TC      WAITLIST
                EBANK=  UNIT/R/
                2CADR   R10,R11

MAKEACCS        CA      FOUR
                TCF     GOREADAX        # DO PHASE CHANGE AND RECALL READACCS

AVEGOUT         EXTEND
                DCA     AVOUTCAD        # SET UP FINAL SERVICER EXIT
                DXCH    AVGEXIT

                CA      FOUR            # SET 5.4 SPOT FOR REREADAC AND SERVICER
                TC      GNUTFAZ5        # IF REREADAC IS CALLED, IT WILL EXIT
                TC      TASKOVER        # END TASK WITHOUT CALLING READACCS

GNUTFAZ5        TS      L               # SAVE INPUT IN L
                CS      TIME1
                TS      TBASE5          # SET TBASE5
                TCF     +2

GNUFAZE5        TS      L               # SAVE INPUT IN L
                CS      L               # -PHASE IN A, PHASE IN L.
                DXCH    -PHASE5         # SET -PHASE5,PHASE5
                TC      Q

                EBANK=  DVCNTR
AVOUTCAD        2CADR   AVGEND

ENDJBCAD        CADR    SERVEXIT +2

OCT37771        OCT     37771

                BANK    33
                SETLOC  SERVICES
                BANK

                COUNT*  $$/SERV
```

```
# ************* SERVICER ****************

SERVICER        TC      PHASCHNG        # RESTART REREADAC + SERVICER
                OCT     16035
                OCT     20000
                EBANK=  DVCNTR
                2CADR   GETABVAL

                CAF     PRIO31          # INITIALIZE 1/PIPADT IN CASE RESTART HAS
                TS      1/PIPADT        # CAUSED LASTBIAS TO BE SKIPPED.

                TC      BANKCALL        # PIPA COMPENSATION CALL
                CADR    1/PIPA

GETABVAL        TC      INTPRET
                VLOAD   ABVAL
                        DELV
                EXIT
                CA      MPAC
                TS      ABDELV          # ABDELV = CM/SEC*2(-14).
                EXTEND
                MP      KPIP
                DXCH    ABDVCONV        # ABDVCONV = M/CS * 2(-5).
                EXTEND
                DCA     MASS
                DXCH    MASS1           # INITIALIZE MASS1 IN CASE WE SKIP MASSMON
MASSMON         CS      FLAGWRD8        # ARE WE ON THE SURFACE?
                MASK    SURFFBIT
                EXTEND
                BZF     MOONSPOT        # YES:  BYPASS MASS MESS

                CA      FLGWRD10        # NO:  WHICH VEX SHOULD BE USED?
                MASK    APSFLBIT
                CCS     A
                EXTEND                  # IF EXTEND IS EXECUTED, APSVEX --> A,
                DCA     APSVEX          #       OTHERWISE DPSVEX --> A
                TS      Q

                EXTEND
                DCA     ABDVCONV
                EXTEND
OCT10002        DV      Q               # WHERE APPROPRIATE VEX RESIDES
                EXTEND
                MP      MASS
                DAS     MASS1

MOONSPOT        CA      KPIP1           # TP MPAC = ABDELV AT 2(14) CM/SEC
                TC      SHORTMP         # MULTIPLY BY KPIP1 TO GET
```

```
                DXCH    MPAC            # ABDELV AT 2(7) M/CS
                DAS     DVTOTAL         # UPDATE DVTOTAL FOR DISPLAY

                TC      TMPTOSPT

                TC      BANKCALL
                CADR    QUICTRIG

                CAF     XNBPIPAD
                TC      BANKCALL
                CADR    FLESHPOT
                TC      INTPRET
AVERAGEG        BON     CALL
                        MUNFLAG
                        RVBOTH
                        CALCRVG
                EXIT
GOSERV          TC      QUIKFAZ5

COPYCYCL        TC      COPYCYC

#               CA      ZERO            # A IS ZERO ON RETURN FROM COPYCYC
                TS      PIPATMPX
                TS      PIPATMPY
                TS      PIPATMPZ

                CS      STEERBIT        # CLEAR STEERSW PRIOR TO DVMON.
                MASK    FLAGWRD2
                TS      FLAGWRD2

                CAF     IDLEFBIT        # IS THE IDLE FLAG SET?
                MASK    FLAGWRD7
                CCS     A
                TCF     NODVMON1        # IDLEFLAG = 1, HENCE SET AUXFLAG TO 0.

                CS      FLAGWRD6
                MASK    AUXFLBIT
                CCS     A
                TCF     NODVMON2        # AUXFLAG = 0, HENCE SET AUXFLAG TO 1.

DVMON           CS      DVTHRUSH
                AD      ABDELV
                EXTEND
                BZMF    LOTHRUST

                CS      FLAGWRD2        # SET STEERSW.
                MASK    STEERBIT
                ADS     FLAGWRD2

DVCNTSET        CAF     ONE             # ALLOW TWO PASSES MAXIMUM NOW THAT
```

```
                TS      DVCNTR          # THRUST HAS BEEN DETECTED.

                CA      FLGWRD10        # BRANCH IF APSFLAG IS SET.
                MASK    APSFLBIT
                CCS     A
                TCF     USEJETS

                CA      BIT9            # CHECK GIMBAL FAIL BIT
                EXTEND
                RAND    CHAN32
                EXTEND
                BZF     USEJETS

USEGTS          CS      USEQRJTS
                MASK    DAPBOOLS
                TS      DAPBOOLS
                TCF     SERVOUT

NODVMON1        CS      AUXFLBIT        # SET AUXFLAG TO 0.
                MASK    FLAGWRD6
                TS      FLAGWRD6
                TCF     USEJETS
NODVMON2        CS      FLAGWRD6        # SET AUXFLAG TO 1.
                MASK    AUXFLBIT
                ADS     FLAGWRD6
                TCF     USEJETS

LOTHRUST        TC      QUIKFAZ5
                CCS     DVCNTR
                TCF     DECCNTR

                CCS     PHASE4          # COMFAIL JOB ACTIVE?
                TCF     SERVOUT         # YES:  WON'T NEED ANOTHER.

                TC      PHASCHNG        # 4.37SPOT FOR COMFAIL.
                OCT     00374

                CAF     PRIO25
                TC      NOVAC
                EBANK=  WHICH
                2CADR   COMFAIL

                TCF     SERVOUT

DECCNTR         TS      DVCNTR1
                TC      QUIKFAZ5
                CA      DVCNTR1
                TS      DVCNTR
                INHINT
                TC      IBNKCALL        # IF THRUST IS LOW, NO STEERING IS DONE
```

```
                CADR    STOPRATE        # AND THE DESIRED RATES ARE SET TO ZERO.
USEJETS         CS      DAPBOOLS
                MASK    USEQRJTS
                ADS     DAPBOOLS
SERVOUT         RELINT
                TC      BANKCALL
                CADR    1/ACCS

                CA      PRIORITY
                MASK    LOW9
                TS      PUSHLOC
                ZL
                DXCH    FIXLOC          # FIXLOC AND DVFIND

                TC      QUIKFAZ5
                EXTEND                  # EXIT TO SELECTED ROUTINE WHETHER THERE
                DCA     AVGEXIT         # IS THRUST OR NOT.  THE STATE OF STEERSW
                DXCH    Z               # WILL CONVEY THIS INFORMATION.

XNBPIPAD        ECADR   XNBPIP

                BANK    32
                SETLOC  SERV2
                BANK
                COUNT*  $$/SERV

AVGEND          CA      PIPTIME +1      # FINAL AVERAGE G EXIT
                TS      1/PIPADT        # SET UP FREE FALL GYRO COMPENSATION.

                TC      UPFLAG          # SET DRIFT FLAG.
                ADRES   DRIFTFLG

                TC      BANKCALL
                CADR    PIPFREE

                CS      BIT9
                EXTEND
                WAND    DSALMOUT

                TC      2PHSCHNG
                OCT     5               # GROUP 5 OFF
                OCT     05022           # GROUP 2 ON
                OCT     20000

                TC      INTPRET
                SET     CLEAR
                        NOR29FLG        # SHUT OFF R29 WHEN SERVICER ENDS.
                        SWANDISP        # SHUT OFF R10 WHEN SERVICER ENDS.
                CLEAR   CALL            # RESET MUNFLAG.
                        MUNFLAG
```

```
                                AVETOMID
                        CLEAR   EXIT
                                V37FLAG
AVERTRN                 CA      OUTROUTE        # RETURN TO DESIRED POINT.
                        TC      BANKJUMP

OUTGOAVE                =       AVERTRN
DVCNTR1                 =       MASS1
```

```
                SETLOC  SERV3
                BANK
                COUNT*  $$/SERV

SERVIDLE        EXTEND                  # DISCONNECT SERVICER FROM ALL GUIDANCE
                DCA     SVEXTADR
                DXCH    AVGEXIT

                CS      FLAGWRD7        # DISCONNECT THE DELTA-V MONITOR
                MASK    IDLEFBIT
                ADS     FLAGWRD7

                CAF     LRBYBIT         # TERMINATE R12 IS RUNNING.
                TS      FLGWRD11

                EXTEND
                DCA     NEG0
                DXCH    -PHASE1

                CA      FLAGWRD6        # DO NOT TURN OFF PHASE 2 IF MUNFLAG SET.
                MASK    MUNFLBIT
                CCS     A
                TCF     +4

                EXTEND
                DCA     NEG0
                DXCH    -PHASE2

 +4             EXTEND
                DCA     NEG0
                DXCH    -PHASE3

                EXTEND
                DCA     NEG0
                DXCH    -PHASE6

                CAF     OCT33           # 4.33SPOT FOR GOPOOFIX
                TS      L
                COM
                DXCH    -PHASE4

                TCF     WHIMPER         # PERFORM A SOFTWARE RESTART AND PROCEED
                                        # TO GOTOPOOH WHILE SERVICER CONTINUES TO
                                        # RUN, ALBEIT IN A GROUND STATE WHERE
                                        # ONLY STATE-VECTOR DEPENDENT FUNCTIONS
                                        # ARE MAINTAINED.

                EBANK=  DVCNTR
```

```
SVEXTADR        2CADR   SERVEXIT

                BANK    32
                SETLOC  SERV
                BANK
                COUNT*  $$/SERV

SERVEXIT        TC      PHASCHNG
                OCT     00035

+2              TCF     ENDOFJOB

                BANK    23
                SETLOC  NORMLIZ
                BANK

                COUNT*  $$/SERV
```

```
NORMLIZE        TC      INTPRET
                VLOAD   BOFF
                        RN1
                        MUNFLAG
                        NORMLIZ1
                VSL6    MXV
                        REFSMMAT
                STCALL  R
                        MUNGRAV
                VLOAD   VSL1
                        VN1
                MXV
                        REFSMMAT
                STOVL   V
                        V(CSM)
                VXV     UNIT
                        R(CSM)
                STORE   UHYP
ASCSPOT         EXIT
                EXTEND                  # MAKE SURE GROUP 2 IS OFF
                DCA     NEGO
                DXCH    -PHASE2

                TC      POSTJUMP
                CADR    NORMLIZ2

                BANK    33
                SETLOC  SERVICES
                BANK
                COUNT*  $$/SERV

NORMLIZ1        CALL
                        CALCGRAV
                EXIT

NORMLIZ2        CA      EIGHTEEN
                TC      COPYCYC +1      # DO NOT COPY MASS IN NORMLIZE
                TC      ENDOFJOB

COPYCYC         CA      OCT24           # DEC 20
 +1             INHINT
 +2             MASK    NEG1            # REDUCE BY 1 IF ODD
                TS      ITEMP1
                EXTEND
                INDEX   ITEMP1
                DCA     RN1
                INDEX   ITEMP1
```

```
                DXCH    RN
                CCS     ITEMP1
                TCF     COPYCYC +2
                TC      Q               # RETURN UNDER INHINT

EIGHTEEN        DEC     18
```

```
# ************* PIPA READER *****************
# MOD NO. 00 BY D. LICKLY, DEC. 9 1966
#
# FUNCTIONAL DESCRIPTION
#       SUBROUTINE TO READ PIPA COUNTERS, TRYING TO BE VERY CAREFUL SO THAT WILL BE RESTARTABLE.
#       PIPA READINGS ARE STORED IN THE VECTOR DELV.  THE HIGH ORDER PART OF EACH COMPONENT CONTAINS THE PIPA READING,
#       RESTARTS BEGIN AT REREADAC.
#
#       AT THE END OF THE PIPA READER THE CDUS ARE READ AND STORED AS A
#       VECTOR IN CDUTEMP.  THE HIGH ORDER PART OF EACH COMPONENT CONTAINS
#       THE CDU READING IN 2S COMP IN THE ORDER CDUX,Y,Z.  THE THRUST
#       VECTOR ESTIMATOR IN FINDCDUD REQUIRES THE CDUS BE READ AT PIPTIME.
#
# CALLING SEQUENCE AND EXIT
#       CALL VIA TC, ISWCALL, ETC.
#       EXIT IS VIA Q.
#
# INPUT
#       INPUT IS THROUGH THE COUNTERS PIPAX, PIPAY, PIPAZ, AND TIME2.
#
# OUTPUT
#       HIGH ORDER COMPONENTS OF THE VECTOR DELV CONTAIN THE PIPA READINGS.
#       PIPTIME CONTAINS TIME OF PIPA READING.
#
# DEBRIS (ERASABLE LOCATIONS DESTROYED BY PROGRAM)
#       TEMX, TEMY, TEMZ, PIPAGE

                BANK    37
                SETLOC  SERV1
                BANK

                COUNT*  $$/SERV

PIPASR          EXTEND
```

```
                DCA     TIME2
                DXCH    PIPTIME1        # CURRENT TIME POSITIVE VALUE
 +3             CS      ZERO            # INITIALIZE THESE AT NEG. ZERO.
                TS      TEMX
                TS      TEMY
                TS      TEMZ

                CA      ZERO
                TS      DELVZ
                TS      DELVZ +1
                TS      DELVY
                TS      DELVY +1
                TS      DELVX +1
                TS      PIPAGE          # SHOW PIPA READING IN PROGRESS

REPIP1          EXTEND
                DCS     PIPAX           # X AND Y PIPS READ
                DXCH    TEMX
                DXCH    PIPAX           # PIPAS SET TO NEG ZERO AS READ.
                TS      DELVX
                LXCH    DELVY

REPIP3          CS      PIPAZ           # REPEAT PROCESS FOR Z PIP
                XCH     TEMZ
                XCH     PIPAZ
DODELVZ         TS      DELVZ

REPIP4          EXTEND                  # COMPUTE GUIDANCE PERIOD
                DCA     PIPTIME1
                DXCH    PGUIDE
                EXTEND
                DCS     PIPTIME
                DAS     PGUIDE

                CA      CDUX            # READ CDUS INTO HIGH ORDER CDUTEMPS
                TS      CDUTEMPX
                CA      CDUY
                TS      CDUTEMPY
                CA      CDUZ
                TS      CDUTEMPZ
                CA      DELVX
                TS      PIPATMPX
                CA      DELVY
                TS      PIPATMPY
                CA      DELVZ
                TS      PIPATMPZ

                TC      Q
```

```
REREADAC        CCS     PIPAGE
                TCF     READACCS        # PIP READING NOT STARTED.  GO TO BEGINNING

                CAF     DONEADR         # SET UP RETURN FROM PIPASR
                TS      Q

                CCS     DELVZ
                TCF     REPIP4          # Z DONE, GO DO CDUS
                TCF     +3              # Z NOT DONE, CHECK Y.
                TCF     REPIP4
                TCF     REPIP4

                ZL
                CCS     DELVY
                TCF     +3
                TCF     CHKTEMX         # Y NOT DONE, CHECK X.
                TCF     +1
                LXCH    PIPAZ           # Y DONE, ZERO Z PIP.

                CCS     TEMZ
                CS      TEMZ            # TEMZ NOT = -0, CONTAINS -PIPAZ VALUE.
                TCF     DODELVZ
                TCF     -2
                LXCH    DELVZ           # TEMZ = -0, L HAS ZPIP VALUE.
                TCF     REPIP4

CHKTEMX         CCS     TEMX            # HAS THIS CHANGED
                CS      TEMX            # YES
                TCF     +3              # YES
                TCF     -2              # YES
                TCF     REPIP1          # NO
                TS      DELVX

                CS      TEMY
                TS      DELVY

                CS      ZERO            # ZERO X AND Y PIPS
                DXCH    PIPAX           # L STILL ZERO FROM ABOVE

                TCF     REPIP3

DONEADR         GENADR  PIPSDONE
```

```
                BANK    33
                SETLOC  SERVICES
                BANK

                COUNT*  $$/SERV

TMPTOSPT        CA      CDUTEMPY        # THIS SUBROUTINE, CALLED BY AN RTB FROM
                TS      CDUSPOTY        # INTERPRETIVE, LOADS THE CDUS CORRESPON-
                CA      CDUTEMPZ        # DING TO PIPTIME INTO THE CDUSPOT VECTOR.
                TS      CDUSPOTZ
                CA      CDUTEMPX
                TS      CDUSPOTX
                TC      Q

# LRHTASK IS A WAITLIST TASK SET BY READACCS DURING THE DESCENT BRAKING
# PHASE WHEN THE ALT TO THE LUNAR SURFACE IS LESS THAN 25,000 FT.  THIS
# TASK CLEARS THE ALTITUDE MEASUREMENT MADE DISCRETE AND INITIATES THE
# LANDING RADAR MEASUREMENT JOB (LRHJOB) TO TAKE A ALTITUDE MEASUREMENT
# 50 MS PRIOR TO THE NEXT READACCS TASK.

                BANK    21
                SETLOC  R10
                BANK

                COUNT*  $$/SERV

LRHTASK         CS      FLGWRD11
                MASK    LRBYBIT
                EXTEND
                BZF     GRP2OFF         # LR BYPASS SET -- BYPASS ALL LR READING.

                CA      READLBIT
                MASK    FLGWRD11        # IS READLR FLAG SET?
                EXTEND
                BZF     GRP2OFF         # NO.  BYPASS LR READ.

                CS      FLGWRD11
                MASK    NOLRRBIT        # IS LR READ INHIBITED?
                EXTEND
                BZF     GRP2OFF         # YES.  BYPASS LR READ.

                CA      PRIO32          # LR READ OK.  SET JOB TO DO IT
                TC      NOVAC           # ABOUT 50 MS. PRIOR TO PIPA READ.
                EBANK=  HMEAS
                2CADR   LRHJOB

GRP2OFF         EXTEND
                DCA     NEG0
                DXCH    -PHASE2
                TCF     R10,R11A

                BANK    33
                SETLOC  SERVICES
                BANK
```

```
                COUNT*  $$/SERV

# HIGATASK IS ENTERED APPROXIMATELY 6 SECS PRIOR TO HIGATE DURING THE
# DESCENT PHASE.  HIGATASK SETS THE HIGATE FLAG (BIT11) AND THE LR INHIBIT
# FLAG (BIT10) IN LRSTAT.  THE HIGATJOB IS SET UP TO REPOSITION THE LR
# ANTENNA FROM POSITION 1 TO POSITION 2.  IF THE REPOSITIONING IS
# SUCCESSFUL THE ALT BEAM AND VELOCITY BEAMS ARE TRANSFORMED TO THE NEW
# ORIENTATION IN NB COORDINATES AND STORED IN ERASABLE.

HIGATASK        INHINT
                CS      PRIO3           # SET HIGATE AND LR INHIBIT FLAGS
                MASK    FLGWRD11
                AD      PRIO3
                TS      FLGWRD11
                CAF     PRIO32
                TC      FINDVAC         # SET LR POSITIONING JOB (POS2)
                EBANK=  HMEAS
                2CADR   HIGATJOB

                TCF     CONTSERV        # CONTINUE SERVICER
```

```
# MUNRETRN IS THE RETURN LOC FROM SPECIAL AVE G ROUTINE (MUNRVG)

MUNRETRN        EXIT

                CS      FLGWRD11
                MASK    LRBYBIT
                EXTEND
                BZF     COPYCYC1        # BYPASS LR LOGIC IF BIT15 IS SET.

                CA      READLBIT        # SEE IF ALT < 35000 FT LAST CYCLE
                MASK    FLGWRD11
                EXTEND
                BZF     35KCHK          # ALT WAS > 35000 FT LAST CYCLE   CHK NOW

                CAF     XORFLBIT        # WERE WE BELOW 30000 FT LAST PASS?
                MASK    FLGWRD11
                EXTEND
                BZF     XORCHK          # NO -- TEST THIS PASS
HITEST          CAF     PSTHIBIT        # CHECK FOR HIGATE
                MASK    FLGWRD11
                EXTEND
                BZF     HIGATCHK        # NOT AT HIGATE LAST CYCLE -- CHK THIS CYCLE

POS2CHK         CAF     BIT7            # VERIFY LR IN POS2
                EXTEND
                RAND    CHAN33
                EXTEND
                BZF     UPDATCHK        # IT IS -- CHECK FOR LR UPDATE
                CAF     BIT13
                EXTEND
                RAND    CHAN12
                EXTEND
                BZF     LRPOSALM        # LR NOT IN POS2 OR REPOSITIONING -- BAD
                TCF     CONTSERV        # LR BEING REPOSITIONED -- CONTINUE SERV

HIGATCHK        CA      TTF/8           # IS TTF > CRITERION?  (TTF IS NEGATIVE)
                AD      RPCRTIME
                EXTEND
                BZMF    POS1CHK         # NO

                CA      EBANK4          # MUST SWITCH EBANKS
                XCH     EBANK
                TS      L               # SAVE IN L

                EBANK=  XNBPIP
                CS      XNBPIP          # UXBXP IN GSOP CH5
                EBANK=  DVCNTR
                LXCH    EBANK           # RESTORE EBANK
                AD      RPCRTQSW        # QSW - UXBXP
```

```
                EXTEND
                BZMF    HIGATASK        # IF UXBXP > QSW, THEN REPOSITION

POS1CHK         CAF     BIT6            # HIGATE NOT IN SIGHT -- DO POS1 CHK
                EXTEND
                RAND    33
                EXTEND
                BZF     UPDATCHK        # LR IN POS1 -- CHECK FOR LR UPDATE

LRPOSALM        TC      ALARM           # LR NOT IN PROPER POS-ALARM-BYPASS UPDATE
                OCT     511             # AND CONTINUE SERVICER
CONTSERV        INHINT
                CS      BITS4-7
                MASK    FLGWRD11        # CLEAR LR MEASUREMENT MADE DISCRETES.
                TS      FLGWRD11

                TC      IBNKCALL        # SET LR LITES PROPERLY
                CADR    R12LITES
```

```
COPYCYC1        TC      QUIKFAZ5

R29?            CA      FLAGWRD3
                MASK    NR29&RDR
                CCS     A               # IS NOR29FLG OR READRFLG SET?
                TCF     R29NODES        # YES, SO DON'T DESIGNATE.

                CA      RADMODES        # NO, SO R29 IS CALLED FOR.
                MASK    OCT10002        # IS THE RR NOT ZEROING ITS CDUS, AND
                CCS     A               # IS THE RENDEZVOUS RADAR IN AUTO MODE?
                TCF     R29NODES        # NO, SO DON'T DESIGNATE.

                CA      RADMODES
                MASK    PRIO22
                CCS     A               # IS RR REPOSITIONING OR REMODING?
                TCF     NOR29NOW        # YES:  COME BACK IN 2 SECONDS & TRY AGAIN.

                TCF     R29

R29NODES        INHINT                  # R29 NOT ALLOWED THIS CYCLE.
                CS      DESIGBIT        # SHOW THAT DESIGNATION IS OFF.
                MASK    RADMODES
                TS      RADMODES

NOR29NOW        TC      INTPRET         # INTPRET DOES A RELINT.
                VLOAD   ABVAL           # MPAC = ABVAL( NEW SM. POSITION VECTOR )
                        R1S
                PUSH    DSU             #                                (2)
                        /LAND/
                STORE   HCALC           # NEW HCALC*2(24)M.
                STORE   HCALC1
                DMPR    RTB
                        ALTCONV
                        SGNAGREE
                STOVL   ALTBITS         # ALTITUDE FOR R10 IN BIT UNITS.
                        UNIT/R/
                VXV     UNIT
                        UHYP
                STOVL   UHZP            # DOWNRANGE HALF-UNIT VECTOR FOR R10.
                        R1S
                VXM     VSR4
                        REFSMMAT
                STOVL   RN1             # TEMP. REF. POSITION VECTOR*2(29)M.
                        V1S
                VXM     VSL1
                        REFSMMAT
                STOVL   VN1             # TEMP. REF. VELOCITY VECTOR 2(7) M/CS.
                        UNIT/R/
                VXV     ABVAL
```

```
                        V1S
                SL1     DSQ
                DDV
                DMPR    RTB
                        ARCONV1
                        SGNAGREE
COPYCYC2        EXIT                    # LEAVE ALTITUDE RATE COMPENSATION IN MPAC
                INHINT
                CA      UNIT/R/         # UPDATE RUNIT FOR R10.
                TS      RUNIT
                CA      UNIT/R/ +2
                TS      RUNIT +1
                CA      UNIT/R/ +4
                TS      RUNIT +2
                CA      MPAC            # LOAD NEW DALTRATE FOR R10.
                TS      DALTRATE

                EXTEND
                DCA     R1S
                DXCH    R
                EXTEND
                DCA     R1S +2
                DXCH    R +2
                EXTEND
                DCA     R1S +4
                DXCH    R +4
                EXTEND
                DCA     V1S
                DXCH    V
                EXTEND
                DCA     V1S +2
                DXCH    V +2
                EXTEND
                DCA     V1S +4
                DXCH    V +4

                TCF     COPYCYCL        # COMPLETE THE COPYCYCL.
```

```
# ALTCHK COMPARES CURRENT ALTITUDE (IN HCALC) WITH A SPECIFIED ALTITUDE FROM A TABLE BEGINNING AT ALTCRIT.
# ITS CALLING SEQUENCE IS AS FOLLOWS:-
#
#       L       CAF     N
#       L+1     TC      BANKCALL
#       L+2     CADR    ALTCHK
#       L+3     RETURN HERE IF HCALC STILL > SPECIFIED CRITERION.   C(L) = +0.
#       L+4     RETURN HERE IF HCALC < OR = SPECIFIED CRITERION.   C(A) = C(L) = +0
#
# ALTCHK MUST BE BANKCALLED EVEN FROM ITS OWN BANK.   N IS THE LOCATION, RELATIVE TO THE TAG ALTCRIT,
# OF THE BEGINNING OF THE DP CONSTANT TO BE USED AS A CRITERION.

ALTCHK          EXTEND
                INDEX   A
                DCA     ALTCRIT
                DXCH    MPAC +1
                EXTEND
                DCS     HCALC
                DAS     MPAC +1
                TC      BRANCH +4
                CAF     ZERO            # BETTER THAN A NOOP, PERHAPS
                INCR    BUF2
                TCF     SWRETURN

ALTCRIT         =       25KFT

25KFT           2DEC    7620 B-24               # (0)

50KFT           2DEC    15240 B-24              # (2)

50FT            2DEC    15.24 B-24              # (4)

30KFT           2DEC    9144 B-24               # (6)

2KFT/SEC        DEC     6.096 B-7               # 2000 FT/SEC AT 2(7) M/CS


# (A REMARK WAS LIKELY TO BE NEEDED HERE TO EXPLAIN XORCHK) 4/JUN/09,FB

XORCHK          CAF     SIX             # ARE WE BELOW 30000 FT?
                TC      BANKCALL
                CADR    ALTCHK
                TCF     HITEST          # CONTINUE LR UPDATE
                TC      UPFLAG          # YES: INHIBIT X-AXIS OVERRIDE
                ADRES   XOVINFLG
                TC      UPFLAG
                ADRES   XORFLG
                TCF     HITEST          # CONTINUE LR UPDATE

35KCHK          CAF     TWO             # ARE WE BELOW 35000 FT?
```

```
                TC      BANKCALL
                CADR    ALTCHK
                TCF     CONTSERV
                TC      UPFLAG
                ADRES   READLR          # SET READLR FLAG TO ENABLE LR READING.
                TCF     CONTSERV
```

```
# ****************************************************************

CALCGRAV        UNIT     PUSH             # SAVE UNIT/R/ IN PUSHLIST      (18)
                STORE    UNIT/R/
                LXC,1    SLOAD            # RTX2 = 0 IF EARTH ORBIT, =2 IF LUNAR.
                         RTX2
                         RTX2
                DCOMP    BMN
                         CALCGRV1
                VLOAD    DOT              #                               (12)
                         UNITZ
                         UNIT/R/
                SL1      PUSH             #                               (14)
                DSQ      BDSU
                         DP1/20
                PDDL     DDV
                         RESQ
                         34D              # (RN)SQ
                STORE    32D              # TEMP FOR (RE/RN)SQ
                DMP      DMP
                         20J
                VXSC     PDDL
                         UNIT/R/
                DMP      DMP
                         2J
                         32D
                VXSC     VSL1
                         UNITZ
                VAD      STADR
                STORE    UNITGOBL
                VAD      PUSH             # MPAC = UNIT GRAVITY VECTOR.   (18)
CALCGRV1        DLOAD    NORM             # PERFORM A NORMALIZATION ON RMAGSQ IN
                         34D              # ORDER TO BE ABLE TO SCALE THE MU FOR
                         X2               # MAXIMUM PRECISION.
                BDDV*    SLR*
                         -MUDT,1
                         0 -21D,2
                VXSC     STADR
                STORE    GDT1/2           # SCALED AT 2(+7) M/CS
                RVQ

CALCRVG         VLOAD    VXM
                         DELV
                         REFSMMAT
                VXSC     VSL1
                         KPIP1
                STORE    DELVREF
                VSR1     PUSH
                VAD      PUSH             # (DV-OLDGDT)/2 TO PD SCALED AT 2(+7) M/CS.
```

```
                                GDT/2
                        VAD     PDDL
                                VN
                                PGUIDE
                        SL      VXSC
                                6D
                        VAD     STQ
                                RN
                                31D
                        STCALL  RN1             # TEMP STORAGE OF RN SCALED 2(+29) M
                                CALCGRAV

                        VAD     VAD
                        VAD
                                VN
                        STCALL  VN1             # TEMP STORAGE OF VN SCALED 2(+7) M/CS
                                31D

        DP1/20          2DEC    0.05
        SHIFT11         2DEC    1 B-11
```

```
#***************************************************************************
# MUNRVG IS A SPECIAL AVERAGE G INTEGRATION ROUTINE USED BY THRUSTING
# PROGRAMS WHICH FUNCTION IN THE VICINITY OF AN ASSUMED SPHERICAL MOON.
# THE INPUT AND OUTPUT QUANTITIES ARE REFERENCED TO THE STABLE MEMBER
# COORDINATE SYSTEM.

RVBOTH          VLOAD   PUSH
                        G(CSM)
                VAD     PDDL
                        V(CSM)
                        PGUIDE
                DDV     VXSC
                        SHIFT11
                VAD
                        R(CSM)
                STCALL  R1S
                        MUNGRAV
                VAD     VAD
                        V(CSM)
                STADR
                STORE   V1S
                EXIT
                TC      QUIKFAZ5
                TC      INTPRET
                VLOAD
                        GDT1/2
                STOVL   G(CSM)
                        R1S
                STOVL   R(CSM)
                        V1S
                STORE   V(CSM)
                EXIT
                TC      QUIKFAZ5
                TC      INTPRET
MUNRVG          VLOAD   VXSC
                        DELV
                        KPIP2
                PUSH    VAD             # 1ST PUSH:  DELV IN UNITS OF 2(8) M/CS
                        GDT/2
                PUSH    VAD             # 2ND PUSH:  (DELV + GDT)/2, UNITS OF 2(7)
                        V               #                                (12)
                PDDL    DDV
                        PGUIDE
                        SHIFT11
                VXSC
                VAD
                        R
                STCALL  R1S             # STORE R SCALED AT 2(+24) M
                        MUNGRAV
```

```
                VAD     VAD
                VAD
                        V
                STORE   V1S             # STORE V SCALED AT 2(+7) M/CS.
                ABVAL
                STOVL   ABVEL           # STORE SPEED FOR LR AND DISPLAYS.
                        UNIT/R/
                DOT     SL1
                        V1S
                STOVL   HDOTDISP        # HDOT = V. UNIT(R)*2(7) M/CS.
                        R1S
                VXV     VSL2
                        WM
                STODL   DELVS           # LUNAR ROTATION CORRECTION TERM*2(5) M/CS.
                        36D
                DSU
                        /LAND/
                STCALL  HCALC           # FOR NOW, DISPLAY WHETHER POS OR NEG
                        MUNRETRN
MUNGRAV         UNIT                    # AT 36D HAVE ABVAL(R), AT 34D R.R
                STODL   UNIT/R/
                        34D
                SL      BDDV
                        6D
                        -MUDTMUN
                DMP     VXSC
                        SHIFT11
                        UNIT/R/
                STORE   GDT1/2          # 1/2GDT SCALED AT 2(7) M/CS.
                RVQ

1.95SECS        DEC     195
7.5             2DEC    .02286 B-6      # 7.5 FT/SEC AT 2(6) M/CS

2SEC(18)        2DEC    200 B-18

2SEC(28)        2OCT    0000000310      # 2SEC AT 2(28)

4SEC(28)        2DEC    400 B-28

BITS4-7         OCT     110
```

```
UPDATCHK        CAF     NOLRRBIT        # SEE IF LR UPDATE INHIBITED.
                MASK    FLGWRD11
                CCS     A
                TCF     CONTSERV        # IT IS -- NO LR UPDATE
                CAF     RNGEDBIT        # NO INHIBIT -- SEE ALT MEAS. THIS CYCLE.
                MASK    FLGWRD11
                EXTEND
                BZF     VMEASCHK        # NO ALT MEAS THIS CYCLE -- CHECK FOR VEL

POSUPDAT        CA      FIXLOC          # SET PUSHLIST TO ZERO
                TS      PUSHLOC

                TC      INTPRET
                VLOAD   VXM
                        HBEAMNB
                        XNBPIP          # HBEAM SM AT 2(2)
                PDVL    VSL2            # STORE HBEAM IN PD 0-5
                        V1S             # SCALE V AT 2(5) M/CS
                VAD     DOT
                        DELVS           # V RELATIVE TO SURFACE AT 2(5) M/CS
                        0D              # V ALONG HBEAM AT 2(7) M/CS.
                DMP     EXIT
                        RADSKAL         # SCALE TO RADAR COUNTS X 5

                CS      FLGWRD12        # TEST LR ALTITUDE SCALE FACTOR
                MASK    ALTSCBIT
                EXTEND
                BZF     +3              # BRANCH IF HIGH SCALE

                CA      SKALSKAL        # RESCALE IF LOW SCALE
                TC      SHORTMP

            +3  TC      INTPRET
                DAD     SL              # CORRECT HMEAS FOR DOPPLER EFFECT
                        HMEAS
                        7D
                DMP     VXSC            # SLANT RANGE AT 2(21), PUSH UP FOR HBEAM
                        HSCAL           # SLANT RANGE VECTOR AT 2(23) M
                DOT     DSU
                        UNIT/R/         # ALTITUDE AT 2(24) M
                        HCALC           # DELTA H AT 2(24) M
                STORE   DELTAH
                EXIT

                CA      FLGWRD11
                MASK    PSTHIBIT
                EXTEND                  # DO NOT PERFORM DATA REASONABLENESS TEST
                BZF     NOREASON        # UNTIL AFTER HIGATE
```

```
                TC      INTPRET
                ABS     DSU
                        DELQFIX         # ABS(DELTAH) - DQFIX   50 FT NOM
                SL3     DSU             # SCALE TO 2(21)
                        HCALC           # ABS(DELTAH) - (50 + HCALC/8) AT 2(21)
                EXIT

                INCR    LRLCTR
                TC      BRANCH
                TCF     HFAIL           # DELTA H TOO LARGE
                TCF     HFAIL           # DELTA H TOO LARGE
                TC      DOWNFLAG        # TURN OFF ALT FAIL LAMP
                ADRES   HFLSHFLG

NOREASON        CS      FLGWRD11
                MASK    LRINHBIT
                CCS     A
                TCF     VMEASCHK        # UPDATE INHIBITED -- TEST VELOCITY ANYWAY

                TC      INTPRET         # DO POSITION UPDATE
                DLOAD   SR4
                        HCALC           # RESCALE H TO 2(28)M
                EXIT
                EXTEND
                DCA     DELTAH          # STORE DELTAH IN MPAC AND
                DXCH    MPAC            # BRING HCALC INTO A,L
                TC      ALSIGNAG
                EXTEND                  # IF HIGH PART OF HCALC IS NON-ZERO, THEN
                BZF     +2              # HCALC > HMAX,
                TCF     VMEASCHK        # SO UPDATE IS BYPASSED
                TS      MPAC +2         #       FOR LATER SHORTMP

                CS      L               # -H AT 2(14) M
                AD      LRHMAX          # HMAX - H
                EXTEND
                BZMF    VMEASCHK        # IF H >HMAX, BYPASS UPDATE
                EXTEND
                MP      LRWH            # WH(HMAX - H)
                EXTEND
                DV      LRHMAX          # WH(1 - H/HMAX)
                TS      MPTEMP
                TC      SHORTMP2        # DELTAH (WH)(1 - H/HMAX) IN MPAC
                TC      INTPRET         # MODE IS DP FROM ABOVE
                SL1
                VXSC    VAD
                        UNIT/R/         # DELTAR = DH(WH)(1 - H/HMAX) UNIT/R/
                        R1S
                STCALL  GNUR
                        MUNGRAV
                EXIT
```

```
                TC      QUIKFAZ5

                CA      ZERO
RUPDATED        TC      GNURVST

VMEASCHK        TC      QUIKFAZ5        # RESTART AT NEXT LOCATION
                CS      FLGWRD11
                MASK    VELDABIT        # IS V READING AVAILABLE?
                CCS     A
                TCF     VALTCHK         # NO:  SEE IF V READING TO BE TAKEN

VELUPDAT        CS      VSELECT         # PROCESS VELOCITY DATA
                TS      L
                ADS     L               # -2 VSELECT IN L
                AD      L
                AD      L               # -6 VSELECT IN A
                INDEX   FIXLOC
                DXCH    X1              # X1 = -6 VSELECT, X2 = -2 VSELECT

                CA      EBANK4
                TS      EBANK
                EBANK=  LRXCDU

                CA      LRYCDU          # STORE LRCDUS IN CDUSPOTS
                TS      CDUSPOT
                CA      LRZCDU
                TS      CDUSPOT +2
                CA      LRXCDU
                TS      CDUSPOT +4

                TC      BANKCALL
                CADR    QUICTRIG        # GET SINES AND COSINES FOR NBSM

                CA      FIXLOC
                TS      PUSHLOC         # SET PD TO ZERO

                TC      INTPRET
                VLOAD*  CALL
                        VZBEAMNB,1      # CONVERT VBEAM FROM NB TO SM
                        *NBSM*
                PDDL    SL              # STORE IN PD 0-5
                        VMEAS           # LOAD VELOCITY MEASUREMENT
                        12D
                DMP*    PUSH            # SCALE TO M/CS AT 2(6)
                        VZSCAL,2        # AND STORE IN PD 6-7
                EXIT
                CS      ONE
                TS      MODE            # CHANGE STORE MODE TO VECTOR

                CA      PIPTEM          # STORE DELV IN MPAC
```

```
                ZL
                DXCH    MPAC

                CA      PIPTEM +1
                ZL
                DXCH    MPAC +3

                CA      PIPTEM +2
                ZL
                DXCH    MPAC +5

                CA      EBANK7
                TS      EBANK           # RESTORE EBANK 7
                EBANK=  DVCNTR
                TC      INTPRET
                VXSC    PDDL
                        KPIP1           # SCALE DELV TO 2(7) M/CS AND PUSH
                        LRVTIME         # TIME OF DELV AT 2(28) CS
                DSU     DDV
                        PIPTIME         # TU - T(N-1)
                        2SEC(28)
                VXSC    VSL1            # G(N-1)(TU - T(N-1))
                        GDT/2           # SCALED AT 2(7) M/CS
                VAD     VAD             # PUSH UP FOR DELV
                        V               # VU = V(N-1) + DELVU + G(N-1) DTU
                VSL2    VAD             # SCALE TO 2(5) M/CS AND SUBTRACT
                        DELVS           #       MOON ROTATION.
                PUSH    ABVAL           # STORE IN PD
                SR4     DAD             # ABS(VM)/8 + 7.5 AT 2(6)
                        7.5
                STOVL   20D             # STORE IN 20D AND PICK UP VM
                DOT     BDSU            # V(EST) AT 2(6)
                        0               # DELTAV = VMEAS - V(EST)
                PUSH    ABS
                DSU     EXIT            # ABS(DV) - (7.5 + ABS(VM)/8))
                        20D

                INCR    LRMCTR
                TC      BRANCH
                TCF     VFAIL           # DELTA V TOO LARGE.    ALARM
                TCF     VFAIL           # DELTA V TOO LARGE.    ALARM

                TC      DOWNFLAG        # TURN OFF VEL FAIL LAMP
                ADRES   VFLSHFLG

                CA      FLGWRD11
                MASK    VXINHBIT
                EXTEND
                BZF     VUPDAT          # IF VX INHIBIT RESET, INCORPORATE DATA.
```

```
                TC      DOWNFLAG
                ADRES   VXINH           # RESET VX INHIBIT

                CA      VSELECT
                AD      NEG2            # IF VSELECT = 2 (X AXIS).
                EXTEND                  # BYPASS UPDATE
                BZF     ENDVDAT

VUPDAT          CS      FLGWRD11
                MASK    LRINHBIT
                CCS     A
                TCF     VALTCHK         # UPDATE INHIBITED

                TS      MPAC +1

                CA      ABVEL           # STORE E7 ERASABLES NEEDED IN TEMPS
                TS      ABVEL*
                CA      VSELECT
                TS      VSELECT*
                CA      EBANK5
                TS      EBANK           # CHANGE EBANKS

                EBANK=  LRVF
                CS      LRVF
                AD      ABVEL*          # IF V < VF, USE WVF
                EXTEND
                BZMF    USEVF

                CS      ABVEL*
                AD      LRVMAX          # VMAX - V
                EXTEND
                BZMF    WSTOR -1        # IF V > VMAX, W = 0

                EXTEND
                INDEX   VSELECT*
                MP      LRWVZ           # WV(VMAX - V)

                EXTEND
                DV      LRVMAX          # WV( 1 - V/VMAX )
                TCF     WSTOR

USEVF           INDEX   VSELECT*
                CA      LRWVFZ          # USE APPROPRIATE CONSTANT WEIGHT
                TCF     WSTOR

 -1             CA      ZERO
WSTOR           TS      MPAC
                CS      BIT7            # (=64D)
                AD      MODREG
                EXTEND
```

```
                BZMF    +3              # IF IN P65,P66,P67, USE ANOTHER CONSTANT

                CA      LRWVFF
                TS      MPAC

 +3             CA      EBANK7
                TS      EBANK           # CHANGE EBANKS

                EBANK=  ABVEL
                TC      INTPRET
                DMP     VXSC            # W(DELTA V)(VBEAMSM) UP 6-7, 0-5
                VAD
                        V1S             # ADD WEIGHTED DELTA V TO VELOCITY
                STORE   GNUV
                EXIT

                TC      QUIKFAZ5        # DO NOT RE-UPDATE

                CA      SIX
VUPDATED        TC      GNURVST         # STORE NEW VELOCITY VECTOR
ENDVDAT         =       VALTCHK

VALTCHK         TC      QUIKFAZ5        # DO NOT REPEAT ABOVE

                CAF     READVBIT        # TEST READVEL TO SEE IF VELOCITY READING
                MASK    FLGWRD11        # IS DESIRED.
                CCS     A
                TCF     READV           # YES -- READ VELOCITY
                CS      ABVEL           # NO -- SEE IF VELOCITY < 2000 FT/SEC
                AD      2KFT/SEC
                EXTEND
                BZMF    CONTSERV        # V > 2000 FT/SEC  DO NOT READ VEL

                TC      UPFLAG          # V < 2000 FT/SEC  SET READVEL AND READ.
                ADRES   READVEL

READV           CAF     PRIO32          # SET UP JOB TO READ VELOCITY BEAMS.
                TC      NOVAC
                EBANK=  HMEAS
                2CADR   LRVJOB

                TCF     CONTSERV        # CONTINUE WITH SERVICER

GNURVST         TS      BUF             # STORE GNUR (=GNUV) IN R1S OR V1S
                EXTEND                  # A = 0 FOR R, A = 6 FOR V
                DCA     GNUR
                INDEX   BUF
                DXCH    R1S
                EXTEND
```

```
                DCA      GNUR +2
                INDEX    BUF
                DXCH     R1S +2
                EXTEND
                DCA      GNUR +4
                INDEX    BUF
                DXCH     R1S +4
                TC       Q

QUIKFAZ5        CA       EBANK3
                XCH      EBANK          # SET EBANK 3
                DXCH     L              # Q TO A, A TO L
                EBANK=   PHSNAME5
                TS       PHSNAME5
                LXCH     EBANK
                EBANK=   DVCNTR
                TC       A

HFAIL           CS       LRRCTR
                EXTEND
                BZF      NORLITE        # IF R = 0, DO NOT TURN ON TRK FAIL
                AD       LRLCTR
                MASK     NEG3
                EXTEND                  # IF L-R LT 4, DO NOT TURN ON TRK FAIL
                BZF      +2
                TCF      NORLITE

                TC       UPFLAG         # AND SET BIT TO TURN ON TRACKER FAIL LITE
                ADRES    HFLSHFLG

NORLITE         CA       LRLCTR
                TS       LRRCTR         # SET R = L

                TCF      VMEASCHK

VFAIL           CS       LRSCTR         # DELTA Q LARGE
                EXTEND                  # IF S = 0, DO NOT TURN ON TRACKER FAIL
                BZF      NOLITE
                AD       LRMCTR         # M-S
                MASK     NEG3           # TEST FOR M-S > 3
                EXTEND                  # IF M-S > 3, THEN TWO OR MORE OF THE
                BZF      +2             #       LAST FOUR V READINGS WERE BAD,
                TCF      NOLITE         #       SO TURN ON VELOCITY FAIL LIGHT

                TC       UPFLAG         # AND SET BIT TO TURN ON TRACKER FAIL LITE
                ADRES    VFLSHFLG
```

```
NOLITE          CA      LRMCTR          # SET S = M
                TS      LRSCTR

                CCS     VSELECT         # TEST FOR Z COMPONENT
                TCF     ENDVDAT         # NOT Z, DO NOT SET VX INHIBIT

                TC      UPFLAG          # Z COMPONENT - SET FLAG TO SKIP X
                ADRES   VXINH           # COMPONENT, AS ERROR MAY BE DUE TO CROSS
                TCF     ENDVDAT         # LOBE LOCK UP NOT DETECTED ON X AXIS.
```

```
# ******************************************************************************
# LRVJOB IS SET WHEN THE LEM IS BELOW 15000 FT DURING THE LANDING PHASE
# THIS JOB INITIALIZES THE LANDING RADAR READ ROUTINE FOR 5 VELOCITY
# SAMPLES AND GOES TO SLEEP WHILE THE SAMPLING IS DONE -- ABOUT 500 MS.
# WITH A GOODEND RETURN THE DATA IS STORED IN VMEAS AND BIT7 OF LRSTAT
# IS SET.  THE GIMBAL ANGLES ARE READ ABOUT MIDWAY IN THE SAMPLINGS.

170MS           EQUALS  ND1

LRVJOB          CA      170MS           # SET TASK TO READ CDUS + PIPAS
                TC      WAITLIST
                EBANK=  LRVTIME
                2CADR   RDGIMS

                CCS     VSELECT         # SEQUENCE LR VEL BEAM SELECTOR
                TCF     +2
                CAF     TWO             # IF ZERO, RESET TO TWO
                DOUBLE                  # 2XVSELECT USED FOR VBEAM INDEX IN LRVEL
                TC      BANKCALL        # GO INITIALIZE LR VEL READ ROUTINE
                CADR    LRVEL
                TC      BANKCALL        # PUT LRVJOB TO SLEEP ABOUT 500 MS
                CADR    RADSTALL
                TCF     VBAD
                CCS     STILBADV        # IS DATA GOOD JUST PRESENT?
                TCF     VSTILBAD        # JUST GOOD -- MUST WAIT 4 SECONDS.

                INHINT
                EXTEND                  # GOOD RETURN -- STOW AWAY VMEAS
                DCA     SAMPLSUM
                DXCH    VMEAS
                CA      EBANK4          # FOR DOWNLINK
                TS      EBANK
                EBANK=  LRVTIME

                EXTEND
                DCA     LRVTIME
                DXCH    LRVTIMDL
                EXTEND
                DCA     LRXCDU
                DXCH    LRXCDUDL
                CA      LRZCDU
                TS      LRZCDUDL
                CA      EBANK7
                TS      EBANK
                EBANK=  VSELECT

                CS      FLGWRD11        # SET BIT TO INDICATE VELOCITY
                MASK    VELDABIT        # MEASUREMENT MADE
```

```
                ADS     FLGWRD11
ENDLRV          CCS     VSELECT         # UPDATE VSELECT
                TCF     +2
                CA      TWO
                TS      VSELECT
                TCF     ENDOFJOB

VBAD            CAF     TWO             # SET STILBAD TO WAIT 4 SECONDS
VSTILBAD        TS      STILBADV
                TCF     ENDLRV

# LRHJOB IS SET BY LRHTASK WHEN LEM IS BELOW 25000 FT.  THIS JOB
# INITIALIZES THE LR READ ROUTINE FOR AN ALT MEASUREMENT AND GOES TO
# SLEEP WHILE THE SAMPLING IS DONE -- ABOUT 95 MS.  WITH A GOODEND RETURN
# THE ALT DATA IS STORED IN HMEAS AND BIT7 OF LRSTAT IS SET.

                BANK    34
                SETLOC  R12STUFF
                BANK

                COUNT*  $$/SERV

LRHJOB          TC      BANKCALL        # INITIATE LR ALT MEASUREMENT
                CADR    LRALT
                TC      BANKCALL        # LRHJOB TO SLEEP ABOUT 95MS
                CADR    RADSTALL
                TCF     HBAD
                CCS     STILBADH        # IS DATA GOOD JUST PRESENT?
                TCF     HSTILBAD        # JUST GOOD -- MUST WAIT 4 SECONDS.

                INHINT
                EXTEND
                DCA     SAMPLSUM        # GOOD RETURN -- STORE AWAY LRH DATA
                DXCH    HMEAS           # LRH DATA 1.079 FT/BIT
                EXTEND                  # FOR DOWNLINK
                DCA     PIPTIME1
                DXCH    MKTIME

                EXTEND
                DCA     CDUTEMPY        # CDUY,Z = AIG,AMG
                DXCH    AIG

                CA      CDUTEMPX        # CDUX = AOG
                TS      AOG

                CS      FLGWRD11        # SET BIT TO INDICATE RANGE
                MASK    RNGEDBIT        # MEASUREMENT MADE.
                ADS     FLGWRD11
ENDLRH          TC      ENDOFJOB        # TERMATE LRHJOB
```

```
HBAD            CA      FLAGWRD5
                MASK    RNGSCBIT        # IS BAD RETURN DUE TO SCALE CHANGE?
                EXTEND
                BZF     HSTILBAD -1     # NO  RESET HSTILBAD
                TC      DOWNFLAG        # YES  RESET SCALE CHANGE BIT AND IGNORE
                ADRES   RNGSCFLG
                TC      ENDOFJOB

                CAF     TWO             # SET STILBAD TO WAIT 4 SECONDS
HSTILBAD        TS      STILBADH
                TC      ENDOFJOB

                BANK    34
                SETLOC  SERV4
                BANK

                COUNT*  $$/SERV

# RDGIMS IS A TASK SET UP BY LRVJOB TO PICK UP THE IMU CDUS AND TIME
# AT ABOUT THE MIDPOINT OF THE LR VEL READ ROUTINE WHEN 5 VEL SAMPLES
# ARE SPECIFIED.

                EBANK=  LRVTIME
RDGIMS          EXTEND
                DCA     TIME2           # PICK UP TIME2, TIME1
                DXCH    LRVTIME         #       AND SAVE IN LRVTIME

                EXTEND
                DCA     CDUX            # PICK UP CDUX AND CDUY
                DXCH    LRXCDU          #       AND SAVE IN LRXCDU AND LRYCDU

                CA      CDUZ
                TS      LRZCDU          # SAVE CDUZ IN LRZCDU

                CA      PIPAX
                TS      PIPTEM          # SAVE PIPAX IN PIPTEM

                EXTEND
                DCA     PIPAY           # PICK UP PIPAY AND PIPAZ
                DXCH    PIPTEM +1       #       AND SAVE IN PIPTEM +1 AND PIPTEM +2
                TC      TASKOVER

                BANK    33
                SETLOC  SERVICES
                BANK

                COUNT*  $$/SERV

                EBANK=  DVCNTR
```

```
# HIGATJOB IS SET APPROXIMATELY 6 SECONDS PRIOR TO HIGH GATE DURING
# THE DESCENT BURN PHASE OF LUNAR LANDING.  THIS JOB INITIATES THE
# LANDING RADAR REPOSITIONING ROUTINE AND GOES TO SLEEP UNTIL THE
# LR ANTENNA MOVES FROM POSITION 1 TO POSITION 2.  IF THE LR ANTENNA
# ACHIEVES POSITION 2 WITHIN 22 SECONDS THE ALTITUDE AND VELOCITY
# BEAM VECTORS ARE RECOMPUTED TO REFLECT THE NEW ORIENTATION WITH
# RESPECT TO THE NB.  BIT10 OF LRSTAT IS CLEARED TO ALLOW LR
# MEASUREMENTS AND THE JOB TERMINATES.

HIGATJOB        TC      BANKCALL        # START LRPOS2 JOB
                CADR    LRPOS2
                TC      BANKCALL        # PUT HIGATJOB TO SLEEP UNTIL JOB IS DONE
                CADR    RADSTALL
                TCF     POSALARM        # BAD END    ALARM

POSGOOD         CA      PRIO23          # REDUCE PRIORITY FOR INTERPRETIVE COMPS.
                TC      PRIOCHNG

                TC      SETPOS2         # LR IN POS2 -- SET UP TRANSFORMATIONS

                TC      DOWNFLAG
                ADRES   NOLRREAD        # RESET NOLRREAD FLAG TO ENABLE LR READING
                TC      ENDOFJOB

POSALARM        CA      OCT523
                TC      BANKCALL
                CADR    PRIOLARM        # FLASH ALARM CODE
                TCF     GOTOPOOH        # TERMINATE
                TCF     +3              # PROCEED -- TRY AGAIN
                TCF     ENDOFJOB        # V 32 E    TERMINATE R12
                TC      ENDOFJOB

 +3             CA      BIT7            # SEE IF IN POS2 YET
                EXTEND
                RAND    CHAN33
                EXTEND
                BZF     POSGOOD         # POS2 ACHIEVED   SET UP ANTENNA BEAMS
                TCF     POSALARM        # STILL DIDN'T MAKE IT   REALARM

OCT523          OCT     00523


SETPOS1         TC      MAKECADR        # MUST BE CALLED BY BANKCALL
                TS      LRADRET1        # SAVE RETURN CADR.  SINCE BUP2 CLOBBERED

                CAF     TWO
                TS      STILBADH        # INITIALIZE STILBAD
                TS      STILBADV        # INITIALIZE STILBAD

                CA      ZERO            # INDEX FOR LRALPHA, LRBETA IN POS 1.
```

```
                TS      LRLCTR          # SET L,M,R, ANS S TO ZERO
                TS      LRMCTR
                TS      LRRCTR
                TS      LRSCTR
                TS      VSELECT         # INITIALIZE VSELECT

                TC      SETPOS          # CONTINUE WITH COMPUTATIONS.

                CA      LRADRET1
                TC      BANKJUMP        # RETURN TO CALLER

SETPOS2         CA      TWO             # INDEX FOR POS2
SETPOS          XCH     Q               # SAVE INDEX IN Q
                TS      LRADRET         # SAVE RETURN

                CA      EBANK5
                TS      EBANK
                EBANK=  LRALPHA

                EXTEND
                INDEX   Q
                DCA     LRALPHA         # LRALPHA IN A, LRBETA IN L
                TS      CDUSPOT +4      # ROTATION ABOUT X
                LXCH    CDUSPOT         # ROTATION ABOUT Y
                CA      ZERO
                TS      CDUSPOT +2      # ZERO ROTATION ABOUT Z.

                CA      EBANK7
                TS      EBANK
                EBANK=  LRADRET

                TC      INTPRET
                VLOAD   CALL
                        UNITY           # CONVERT UNITY(ANTENNA) TO NB
                        TRG*SMNB
                STOVL   VYBEAMNB
                        UNITX           # CONVERT UNITX(ANTENNA) TO NB
                CALL
                        *SMNB*
                STORE   VXBEAMNB
                VXV     VSL1
                        VYBEAMNB
                STOVL   VZBEAMNB        # Z = X * Y
                        HBEAMANT
                CALL
                        *SMNB*          # CONVERT TO NB
                STORE   HBEAMNB
                EXIT
```

TC LRADRET

```
                BANK    21
                SETLOC  R10
                BANK

                EBANK=  UNIT/R/
                COUNT*  $$/R10

LANDISP         LXCH    PIPCTR1         # UPDATE TBASE2 AND PIPCTR SIMULTANEOUSLY.
                CS      TIME1
                DXCH    TBASE2

                CS      FLAGWRD7        # IS LANDING ANALOG DISPLAYS FLAG SET?
                MASK    SWANDBIT
                CCS     A
                TCF     DISPRSET        # NO.
                CA      IMODES33        # BIT 7 = 0 (DO ALTRATE), =1 (DO ALT.)
                MASK    BIT7
                CCS     A
                TCF     ALTOUT
ALTROUT         TC      DISINDAT        # CHECK MODE SELECT SWITCH AND DIDFLG.
                CS      IMODES33
                MASK    BIT7
                ADS     IMODES33        # ALTERNATE ALTITUDE RATE WITH ALTITUDE.
                CAF     BIT2            # RATE COMMAND IS EXECUTED BEFORE RANGE.
                EXTEND
                WOR     CHAN14          # ALTRATE (BIT2 = 1), ALTITUDE (BIT2 = 0).
ARCOMP          CA      RUNIT           # COMPUTE ALTRATE=RUNIT.VVECT M/CS *2(-6).
                EXTEND
                MP      VVECT           # MULTIPLY X-COMPONENTS.
                XCH     RUPTREG1        # SAVE SINGLE PRECISION RESULT M/CS*2(-6).
                CA      RUNIT +1        # MULTIPLY Y-COMPONENTS.
                EXTEND
                MP      VVECT +1
                ADS     RUPTREG1        # ACCUMULATE PARTIAL PRODUCTS.
                CA      RUNIT +2        # MULTIPLY Z-COMPONENTS.
                EXTEND
                MP      VVECT +2
                ADS     RUPTREG1        # ALTITUDE RATE IN M/CS *2(-6).
                CA      ARCONV          # CONVERT ALTRATE TO BIT UNITS (.5FPS/BIT)
                EXTEND
                MP      RUPTREG1
                DDOUBL
                DDOUBL
                XCH     RUPTREG1        # ALTITUDE RATE IN BIT UNITS*2(-14).
                CA      DALTRATE        # ALTITUDE RATE COMPENSATION FACTOR.
                EXTEND
                MP      DT
                AD      RUPTREG1
                TS      ALTRATE         # ALTITUDE RATE IN BIT UNITS*2(-14).
                CS      ALTRATE
```

```
                EXTEND                  # CHECK POLARITY OF ALTITUDE RATE.
                BZMF    +2
                TCF     DATAOUT         # NEGATIVE - SEND POS. PULSES TO ALTM REG.
                CA      ALTRATE         # POSITIVE OR ZERO - SET SIGN BIT = 1 AND
                AD      BIT15           # SEND TO ALTM REGISTER. *DO NOT SEND +0*
DATAOUT         TS      ALTM            # ACTIVATE THE LANDING ANALOG DISPLAYS - -
                CAF     BIT3
                EXTEND
                WOR     CHAN14          # BIT3 DRIVES THE ALT/ALTRATE METER.
                TCF     TASKOVER        # EXIT

ALTOUT          TC      DISINDAT        # CHECK MODE SELECT SWITCH AND DIDFLG.
                CS      BIT7
                MASK    IMODES33
                TS      IMODES33        # ALTERNATE ALTITUDE RATE WITH ALTITUDE.
                CS      BIT2
                EXTEND
                WAND    CHAN14
                CCS     ALTBITS         # =-1 IF OLD ALT. DATA TOBE EXTRAPOLATED.
                TCF     +4
                TCF     +3
                TCF     OLDDATA
                TS      ALTBITS         # SET ALTBITS FROM -0 TO +0.
                CS      ONE
                DXCH    ALTBITS         # SET ALTBITS=-1 FOR SWITCH USE NEXT PASS.
                DXCH    ALTSAVE
                CA      BIT10           # NEW ALTITUDE EXTRAPOLATION WITH ALTRATE.
                XCH     Q
                LXCH    7               # ZL
                CA      DT
                EXTEND
                DV      Q               # RESCALE DT*2(-14) TO *2(-9) TIME IN CS.
                EXTEND
                MP      ARTOA2          # .0021322 *2(+8)
                TCF     OLDDATA +1      # RATE APPLIES FOR DT CS.

ZDATA2          DXCH    ALTSAVE
                TCF     NEWDATA
OLDDATA         CA      ARTOA           # RATE APPLIES FOR .5 SEC. (4X/SEC. CYCLE)
                EXTEND
                MP      ALTRATE         # EXTRAPOLATE WITH ALTITUDE RATE.
                DDOUBL
                AD      ALTSAVE +1
                TS      ALTSAVE +1
                CAF     ZERO
                ADS     ALTSAVE
                CAF     POSMAX          # FORCE SIGN AGREEMENT ASSUMING A
                AD      ONE             # NON-NEGATIVE ALTSAVE.
                AD      ALTSAVE +1      # IF ALTSAVE IS NEGATIVE, ZERO ALTSAVE
                TS      ALTSAVE +1      # AND ALTSAVE +1 AT ZERODATA.
```

```
                CAF     ZERO
                AD      POSMAX
                AD      ALTSAVE
                TS      ALTSAVE         # POSSIBLY SKIP TO NEWDATA.
                TCF     ZERODATA
NEWDATA         CCS     ALTSAVE +1
                TCF     +4
                TCF     +3
                CAF     ZERO            # SET NEGATIVE ALTSAVE +1 TO +0.
                TS      ALTSAVE +1
                CCS     ALTSAVE         # PROVIDE A 15 BIT UNSIGNED OUTPUT.
                CAF     BIT15           # THE HI-ORDER PART IS +1 OR +0.
                AD      ALTSAVE +1
                TCF     DATAOUT         # DISPATCH UNSIGNED BITS TO ALTM REG.
DISINDAT        EXTEND
                QXCH    LADQSAVE        # SAVE RETURN TO ALTROUT +1 OR ALTOUT +1
                CAF     BIT6
                EXTEND                  # WISHETH THE ASTRONAUT THE ANALOG
                RAND    CHAN30          # DISPLAYS?  I.E.,
                CCS     A               # IS THE MODE SELECT SWITCH IN PGNCS?
                TCF     DISPRSET        # NO.  ASTRONAUT REQUESTS NO INERTIAL DATA
                CS      FLAGWRD1        # YES. CHECK STATUS OF DIDFLAG.
                MASK    DIDFLBIT
                EXTEND
                BZF     SPEEDRUN        # SET. PERFORM DATA DISPLAY SEQUENCE.
                CS      FLAGWRD1        # RESET. PERFORM INITIALIZATION FUNCTIONS.
                MASK    DIDFLBIT
                ADS     FLAGWRD1        # SET DIDFLAG.
                CS      BIT7
                MASK    IMODES33        # TO DISPLAY ALTRATE FIRST AND ALT. SECOND
                TS      IMODES33
                CS      FLAGWRD0        # ARE WE IN DESCENT TRAJECTORY?
                MASK    R10FLBIT
                EXTEND
                BZF     TASKOVER        # NO
                CAF     BIT8            # YES.
                EXTEND
                WOR     CHAN12          # SET DISPLAY INERTIAL DATA OUTBIT.
                CAF     ZERO
                TS      TRAKLATV        # LATERAL VELOCITY MONITOR FLAG
                TS      TRAKFWDV        # FORWARD VELOCITY MONITOR FLAG
                TS      LATVMETR        # LATVEL MONITOR METER
                TS      FORVMETR        # FORVEL MONITOR METER
                CAF     BIT4
                TC      TWIDDLE
                ADRES   INTLZE
                TCF     TASKOVER
INTLZE          CAF     BIT2
                EXTEND
                WOR     CHAN12          # ENABLE RR ERROR COUNTER.
```

```
                CS      IMODES33
                MASK    BIT8
                ADS     IMODES33        # SET INERTIAL DATA FLAG.
                TCF     TASKOVER

SPEEDRUN        CS      PIPTIME +1      # UPDATE THE VELOCITY VECTOR
                AD      TIME1           # COMPUTE T - TN
                AD      HALF            # CORRECT FOR POSSIBLE OVERFLOW OF TIME1.
                AD      HALF
                XCH     DT              # SAVE FOR LATER USE
                CA      1SEC
                TS      ITEMP5          # INITIALIZE FOR DIVISION LATER
                EXTEND
                DCA     GDT/2           # COMPUTE THE X-COMPONENT OF VELOCITY.
                DDOUBL
                DDOUBL
                EXTEND
                MP      DT
                EXTEND
                DV      ITEMP5
                XCH     VVECT           # VVECT = G(T-TN) M/CS *2(-5)
                EXTEND
                DCA     V               # M/CS *2(-7)
                DDOUBL                  # RESCALE TO 2(-5)
                DDOUBL
                ADS     VVECT           # VVECT = VN + G(T-TN) M/CS *2(-5)
                CA      PIPAX           # DELV CM/SEC *2(-14)
                AD      PIPATMPX        # IN CASE PIPAX HAS BEEN ZEROED
                EXTEND
                MP      KPIP1(5)        # DELV M/CS *2(-5)
                ADS     VVECT           # VVECT = VN + DELV + GN(T-TN) M/CS *2(-5)
                EXTEND
                DCA     GDT/2 +2        # COMPUTE THE Y-COMPONENT OF VELOCITY.
                DDOUBL
                DDOUBL
                EXTEND
                MP      DT
                EXTEND
                DV      ITEMP5
                XCH     VVECT +1
                EXTEND
                DCA     V +2
                DDOUBL
                DDOUBL
                ADS     VVECT +1
                CA      PIPAY
                AD      PIPATMPY
                EXTEND
                MP      KPIP1(5)
                ADS     VVECT +1
```

```
                EXTEND
                DCA     GDT/2 +4        # COMPUTE THE Z-COMPONENT OF VELOCITY.
                DDOUBL
                DDOUBL
                EXTEND
                MP      DT
                EXTEND
                DV      ITEMP5
                XCH     VVECT +2
                EXTEND
                DCA     V +4
                DDOUBL
                DDOUBL
                ADS     VVECT +2
                CA      PIPAZ
                AD      PIPATMPZ
                EXTEND
                MP      KPIP1(5)
                ADS     VVECT +2

                CAF     BIT3            # PAUSE 40 MS TO LET OTHER RUPTS IN.
                TC      VARDELAY

                CS      FLAGWRD0        # ARE WE IN DESCENT TRAJECTORY?
                MASK    R10FLBIT
                CCS     A
                TCF     +2              # YES.
                TC      LADQSAVE        # NO.

                CA      DELVS           # HI X OF VELOCITY CORRECTION TERM.
                AD      VVECT           # HI X OF UPDATED VELOCITY VECTOR.
                TS      ITEMP1          # = VX - DVX M/CS*2(-5).
                CA      DELVS +2        #    Y
                AD      VVECT +1        #    Y
                TS      ITEMP2          # = VY - DVY M/CS*2(-5).
                CA      DELVS +4        #    Z
                AD      VVECT +2        #    Z
                TS      ITEMP3          # = VZ - DVZ M/CS*2(-5).
                CA      ITEMP1          # COMPUTE VHY, VELOCITY DIRECTED ALONG THE
                EXTEND                  # Y-COORDINATE.
                MP      UHYP            # HI X OF CROSS-RANGE HALF-UNIT VECTOR.
                XCH     RUPTREG1
                CA      ITEMP2
                EXTEND
                MP      UHYP +2         #    Y
                ADS     RUPTREG1        # ACCUMULATE PARTIAL PRODUCTS.
                CA      ITEMP3
                EXTEND
                MP      UHYP +4         #    Z
                ADS     RUPTREG1
```

```
                CA      RUPTREG1
                DOUBLE
                XCH     VHY             # VHY=VMP.UHYP M/CS*2(-5).
                CA      ITEMP1          # NO COMPUTE VHZ, VELOCITY DIRECTED ALONG
                EXTEND                  # THE Z-COORDINATE.
                MP      UHZP            # HI X OF DOWN-RANGE HALF-UNIT VECTOR.
                XCH     RUPTREG1
                CA      ITEMP2
                EXTEND
                MP      UHZP +2         #     Y
                ADS     RUPTREG1        # ACCUMULATE PARTIAL PRODUCTS.
                CA      ITEMP3
                EXTEND
                MP      UHZP +4         #     Z
                ADS     RUPTREG1
                CA      RUPTREG1
                DOUBLE
                XCH     VHZ             # VHZ = VMP.UHZP M/CS*2(-5).
GET22/32        CAF     EBANK6          # GET SIN(AOG),COS(AOG) FROM GPMATRIX.
                TS      EBANK
                EBANK=  M22
                CA      M22
                TS      ITEMP3
                CA      M32
                TS      ITEMP4
                CAF     EBANK7
                TS      EBANK
                EBANK=  UNIT/R/
LATFWDV         CA      ITEMP4          # COMPUTE LATERAL AND FORWARD VELOCITIES.
                EXTEND
                MP      VHY
                XCH     RUPTREG1
                CA      ITEMP3
                EXTEND
                MP      VHZ
                ADS     RUPTREG1        # =VHY(COS)AOG+VHZ(SIN)AOG M/CS *2(-5)
                CA      VELCONV         # CONVERT LATERAL VELOCITY TO BIT UNITS.
                EXTEND
                MP      RUPTREG1
                DDOUBL
                XCH     LATVEL          # LATERAL VELOCITY IN BIT UNITS *2(-14).
                CA      ITEMP4          # COMPUTE FORWARD VELOCITY.
                EXTEND
                MP      VHZ
                XCH     RUPTREG1
                CA      ITEMP3
                EXTEND
                MP      VHY
                CS      A
                ADS     RUPTREG1        # =VHZ(COS)AOG-VHY(SIN)AOG M/CS *2(-5).
```

```
                CA      VELCONV         # CONVERT FORWARD VELOCITY TO BIT UNITS.
                EXTEND
                MP      RUPTREG1
                DDOUBL
                XCH     FORVEL          # FORWARD VELOCITY IN BIT UNITS *2(-14).

                CS      MAXVBITS        # ACC.=-199.9989 FT./SEC.
                TS      ITEMP6          # -547 BIT UNITS (OCTAL) AT 0.5571 FPS/BIT

                CAF     ONE             # LOOP TWICE.
VMONITOR        TS      ITEMP5          # FORWARD AND LATERAL VELOCITY LANDING
                INDEX   ITEMP5          #    ANALOG DISPLAYS MONITOR.
                CCS     LATVEL
                TCF     +4
                TCF     LVLIMITS
                TCF     +8D
                TCF     LVLIMITS
                INDEX   ITEMP5
                CS      LATVEL
                AD      MAXVBITS        # +199.9989 FT.SEC.
                EXTEND
                BZMF    CHKLASTY
                TCF     LVLIMITS
                INDEX   ITEMP5
                CA      LATVEL
                AD      MAXVBITS
                EXTEND
                BZMF    +2
                TCF     LVLIMITS
CHKLASTY        INDEX   ITEMP5
                CCS     LATVMETR
                TCF     +4
                TCF     LASTOK
                TCF     +7
                TCF     LASTOK
                INDEX   ITEMP5
                CA      LATVEL
                EXTEND
                BZMF    LASTPOSY +5
                TCF     +5
                INDEX   ITEMP5
                CS      LATVEL
                EXTEND
                BZMF    LASTNEGY +4
LASTOK          INDEX   ITEMP5
                CCS     TRAKLATV
                TCF     LASTPOSY
                TCF     +2
                TCF     LASTNEGY
                INDEX   ITEMP5
```

```
                CA      LATVEL
                EXTEND
                BZMF    NEGVMAXY
                TCF     POSVMAXY
LASTPOSY        INDEX   ITEMP5
                CA      LATVEL
                EXTEND
                BZMF    +2
                TCF     POSVMAXY
                CS      MAXVBITS
                TCF     ZEROLSTY
POSVMAXY        INDEX   ITEMP5
                CS      LATVMETR
                AD      MAXVBITS
                INDEX   ITEMP5
                XCH     RUPTREG3
                CAF     ONE
                TCF     ZEROLSTY +3
LASTNEGY        INDEX   ITEMP5
                CA      LATVEL
                EXTEND
                BZMF    NEGVMAXY
                CA      MAXVBITS
                TCF     ZEROLSTY
NEGVMAXY        INDEX   ITEMP5
                CA      LATVMETR
                AD      MAXVBITS
                COM
                INDEX   ITEMP5
                XCH     RUPTREG3
                CS      ONE
                TCF     ZEROLSTY +3
LVLIMITS        INDEX   ITEMP5
                CCS     TRAKLATV
                TCF     LATVPOS
                TCF     +2
                TCF     LATVNEG
                INDEX   ITEMP5
                CS      LATVMETR
                EXTEND
                BZMF    +2
                TCF     NEGLMLV
                INDEX   ITEMP5
                CS      LATVEL
                EXTEND
                BZMF    LVMINLM
                AD      ITEMP6
                INDEX   ITEMP5
                AD      LATVMETR
                EXTEND
```

```
                BZMF    LVMINLM
                INDEX   ITEMP5
                AD      LATVEL
                EXTEND
                INDEX   ITEMP5
                SU      LATVMETR
                TCF     ZEROLSTY
LATVPOS         INDEX   ITEMP5
                CS      LATVEL
                EXTEND
                BZMF    LVMINLM
                TCF     +5
LATVNEG         INDEX   ITEMP5
                CA      LATVEL
                EXTEND
                BZMF    LVMINLM
                INDEX   ITEMP5
                CS      LATVMETR
                TCF     ZEROLSTY
NEGLMLV         INDEX   ITEMP5
                CA      LATVEL
                EXTEND
                BZMF    LVMINLM
                CA      MAXVBITS
                INDEX   ITEMP5
                AD      LATVMETR
                COM
                INDEX   ITEMP5
                AD      LATVEL
                EXTEND
                BZMF    LVMINLM
                EXTEND
                INDEX   ITEMP5
                SU      LATVEL
                INDEX   ITEMP5
                AD      LATVMETR
                COM
                TCF     ZEROLSTY
LVMINLM         INDEX   ITEMP5
                CS      LATVMETR
                INDEX   ITEMP5
                AD      LATVEL
ZEROLSTY        INDEX   ITEMP5
                XCH     RUPTREG3
                CAF     ZERO
                INDEX   ITEMP5
                TS      TRAKLATV
                INDEX   ITEMP5
                CA      RUPTREG3
                AD      NEGO            # AVOIDS +0 DINC HARDWARE MALFUNCTION
```

```
                INDEX   ITEMP5
                TS      CDUTCMD
                INDEX   ITEMP5
                CA      RUPTREG3
                INDEX   ITEMP5
                ADS     LATVMETR
                CCS     ITEMP5          # FIRST MONITOR FORWARD THEN LATERAL VEL.
                TCF     VMONITOR

                CAF     BITSET          # DRIVE THE X-POINTER DISPLAY.
                EXTEND
                WOR     CHAN14
                TC      LADQSAVE        # GO TO ALTROUT +1 OR TO ALTOUT +1
ZERODATA        CAF     ZERO            # ZERO ALTSAVE AND ALTSAVE +1 - - -
                TS      L               #        NO NEGATIVE ALTITUDES ALLOWED.
                TCF     ZDATA2

# ************************************************************************

DISPRSET        CS      FLAGWRD0        # ARE WE IN DESCENT TRAJECTORY?
                MASK    R10FLBIT
                EXTEND
                BZF     ABORTON         # NO.
                CAF     BIT8            # YES.
                MASK    IMODES33        # CHECK IF INERTIAL DATA JUST DISPLAYED.
                CCS     A
                CAF     BIT2            # YES. DISABLE RR ERROR COUNTER
                AD      BIT8            # NO. REMOVE DISPLAY INERTIAL DATA
                COM
                EXTEND
                WAND    CHAN12
ABORTON         CS      BITS8/7         # RESET INERTIAL DATA, INTERLEAVE FLAGS.
                MASK    IMODES33
                TS      IMODES33
                CS      DIDFLBIT
                MASK    FLAGWRD1
                TS      FLAGWRD1        # RESET DIDFLAG.
                TCF     TASKOVER
# ************************************************************************
BITS8/7         OCT     00300           # INERTIAL DATA AND INTERLEAVE FLAGS.

BITSET          =       PRIO6
# ************************************************************************
```

```
# PROGRAM NAME:  FINDCDUW
# MOD NUMBER:  1         68-07-15
# MOD AUTHOR:  KLUMPP
#
# OBJECTS OF MOD:        1.      TO SUPPLY COMMANDED GIMBAL ANGLES FOR NOUN 22.
#                        2.      TO MAINTAIN CORRECT AND CURRENT THRUST
#                                DIRECTION DATA IN ALL MODES.  THIS IS DONE BY
#                                FETCHING FOR THE THRUST DIRECTION FILTER THE
#                                CDUD'S IN PNGCS-AUTO, THE CDU'S IN ALL OTHER
#                                MODES.
#                        3.      TO SUBSTITUTE A STOPRATE FOR THE NORMAL
#                                AUTOPILOT COMMANDS WHENEVER
#                                1) NOT IN PNGCS-AUTO, OR
#                                2) ENGINE IS OFF.
#
# FUNCTIONAL DESCRIPTION:
#
# FINDCDUW PROVIDES THE INTERFACES BETWEEN THE VARIOUS POWERED FLITE GUIDANCE PROGRAMS
# AND THE DIGITAL AUTOPILOT.  THE INPUTS TO FINDCDUW ARE THE THRUST COMMAND VECTOR
# AND THE WINDOW COMMAND VECTOR, AND THE OUTPUTS ARE THE GIMBAL ANGLE
# INCREMENTS, THE COMMANDED ATTITUDE ANGLE RATES, AND THE COMMANDED
# ATTITUDE LAG ANGLES (WHICH ACCOUNT FOR THE ANGLES BY WHICH THE BODY WILL
# LAG BEHIND A RAMP COMMAND IN ATTITUDE ANGLE DUE TO THE FINITE ANGULAR
# ACCELERATIONS AVAILABLE).
#
# FINDCDUW ALIGNS THE ESTIMATED THRUST VECTOR FROM THE THRUST DIRECTION
# FILTER WITH THE THRUST COMMAND VECTOR, AND, WHEN XDVINHIB SET,
# ALIGNS THE +Z HALF OF THE LM ZX PLANE WITH THE WINDOW COMMAND VECTOR.
#
```

```
# SPECIFICATIONS:
#
# INITIALIZATION:       A SINGLE INTERPRETIVE CALL TO INITCDUW IS REQUIRED
#                       BEFORE EACH GUIDED MANEUVER USING FINDCDUW.
#
# CALL:                 INTERPRETIVE CALL TO FINDCDUW WITH THE THRUST COMMAND
#                       VECTOR IN MPAC.  INTERPRETIVE CALL TO FINDCDUW -2 WITH
#                       THE THRUST COMMAND VECTOR IN UNFC/2 AND NOT IN MPAC.
#
# RETURNS:              NORMAL INTERPRETIVE IN ALL CASES
#                       1.      NORMALLY ALL AUTOPILOT CMDS ARE ISSUED.
#                       2.      IF NOT PNGCS AUTO, DO STOPRATE AND RETURN
#                               WITHOUT ISSUING AUTOPILOT CMDS.
#                       3.      IF ENGINE OFF, DO STOPRATE AND RETURN WITHOUT
#                               ISSUING AUTOPILOT CMDS.
#
# ALARMS:               00401   IF INPUTS DETERMINE AN ATTITUDE IN GIMBAL LOCK.
#                               FINDCDUW DRIVES CDUXD AND CDUYD TO THE RQD VALUES,
#                               BUT DRIVES CDUZD ONLY TO THE GIMBAL LOCK CONE.
#
#                       00402   IF UNFC/2 OR UNWC/2 PRODUCE OVERFLOW WHEN
#                               UNITIZED USING NORMUNIT.  FINDCDUW ISSUES
#                               STOPRATE AS ONLY INPUT TO AUTOPILOT.
#
# INPUTS:               UNFC/2          THRUST COMMAND VECTOR, NEED NOT BE SEMI-UNIT.
#                       UNWC/2          WINDOW COMMAND VECTOR, NEED NOT BE SEMI-UNIT.
#                       OGABIAS         POSSIBLE BIAS FOR OUTER GIMBAL ANGLE (ZEROED IN INITCDUW), UNITS OF PI.
#                       XOVINHIB        FLAG DENOTING X AXIS OVERRIDE INHIBITED.
#                       CSMDOCKD        FLAG DENOTING CSM DOCKED.
#                       STEERSW         FLAG DENOTING INSUFF THRUST FOR THRUST DIR FLTR.
#
# OUTPUTS:              DELCDUX,Y,Z
#                       OMEGAPD,+1,+2
#                       DELPEROR,+1,+2
#                       CPHI,+1,+2 FOR NOUN22
#
# DEBRIS:               FINDCDUW DESTROYS SINCDUX,Y,Z AND COSCDUX,Y,Z BY
#                       WRITING INTO THESE LOCATIONS THE SINES AND COSINES
#                       OF THE CDUD'S IN PNGCS-AUTO, OF THE CDU'S OTHERWISE.
```

```
# INITIALIZATION FOR FINDCDUW

                BANK    30
                SETLOC  FCDUW
                BANK

                EBANK=  ECDUW
                COUNT*  $$/FCDUW

INITCDUW        VLOAD
                        UNITX
                STORE   UNFV/2
                STORE   UNWC/2
                RVQ

# FINDCDUW PRELIMINARIES

                VLOAD                   # FINDCDUW -2:  ENTRY WHEN UNFC/2 PRE-STORD
                        UNFC/2          # INPUT VECTORS NEED NOT BE SEMI-UNIT
FINDCDUW        BOV     SETPD           # FINDCDUW: ENTRY WHEN UNFC/2 IN MPAC
                        FINDCDUW        # INTERPRETER NOW INITIALIZED
                        22              # LOCS 0 THRU 21 FOR DIRECTION COSINE MAT
                STQ     EXIT
                        QCDUWUSR        # SAVE RETURN ADDRESS

# MORE HAUSKEEPING
                CA      ECDUWL
                XCH     EBANK           # SET EBANK
                TS      ECDUWUSR        # SAVE USER'S EBANK

                CA      DAPBOOLS
                MASK    CSMDOCKD        # CSMDOCKD MUST NOT BE BIT15
                CCS     A
                CA      ONE             # INDEX IF CSM DOCKED
                TS      NDXCDUW

                CA      XOVINHIB        # XOVINHIB MUST NOT BE BIT15
                TS      FLPAUTNO        # SET TO POS-NON-ZERO FLAG PNGCS AUTO NOT

                MASK    DAPBOOLS
                TS      FLAGOODW        # FLAGOODW = ANY PNZ NUMBER IF XOV INHIBTD
```

```
# FETCH BASIC DATA
                INHINT                          # RELINT AT PAUTNO (TC INTPRET)

                CA      CDUX                    # FETCH CDUX,CDUY,CDUZ IN ALL CASES, BUT
                TS      CDUSPOTX                #       REPLACE BELOW IF PNGCS AUTO
                CA      CDUY
                TS      CDUSPOTY
                CA      CDUZ
                TS      CDUSPOTZ

                CA      BIT10                   # PNGCS CONTROL BIT
                EXTEND
                RAND    CHAN30
                CCS     A
                TCF     PAUTNO                  # NOT PNGCS (BITS INVERTED)

                CA      BIT14                   # AUTO MODE BIT
                EXTEND
                RAND    CHAN31
                CCS     A
                TCF     PAUTNO                  # NOT AUTO (BITS INVERTED)

                TS      FLPAUTNO                # RESET FLAG PNGCS AUTO NOT

                CA      CDUXD                   # PNGCS AUTO:  FETCH CDUXD,CDUYD,CDUZD
                TS      CDUSPOTX
                CA      CDUYD
                TS      CDUSPOTY
                CA      CDUZD
                TS      CDUSPOTZ
```

```
# FETCH INPUTS
PAUTNO          TC      INTPRET         # ENTERING THRUST CMD STILL IN MPAC
                RTB
                        NORMUNIT
                STOVL   UNX/2           # SEMI-UNIT THRUST CMD AS INITIAL UNX/2
                        UNWC/2
                RTB     RTB
                        NORMUNIT
                        QUICTRIG        # ALWAYS RQD TO OBTAIN TRIGS OF CDUD'S
                STOVL   UNZ/2           # SEMI-UNIT WINDOW CMD AS INITIAL UNZ/2
                        DELV
                BOVB    UNIT
                        NOATTCNT        # AT LEAST ONE ENTERING CMD VCT ZERO
                BOV     CALL
                        AFTRFLTR        # IF UNIT DELV OVERFLOWS SKIP FILTER
                        *SMNB*          # YIELDS UNIT(DELV) IN VEH COORDS FOR FLTR

# THRUST DIRECTION FILTER

                EXIT

                CA      UNFVY/2         # FOR RESTARTS, UNFV/2 ALWAYS INTACT, MPAC
                LXCH    MPAC    +3      #       RENEWD AFTER RETURN FROM CALLER,
                TC      FLTRSUB         #       TWO FILTER UPDATES MAY BE DONE.
                TS      UNFVY/2         # UNFV/2 NEED NOT BE EXACTLY SEMI-UNIT.

                CA      UNFVZ/2
                LXCH    MPAC +5
                TC      FLTRSUB
                TS      UNFVZ/2

                TC      INTPRET         # COMPLETES FILTER
```

```
# FIND A SUITABLE WINDOW POINTING VECTOR

AFTRFLTR        SLOAD   BHIZ            # IF XOV NOT INHIBITED, GO FETCH ZNB
                        FLAGOODW
                        FETCHZNB
                VLOAD   CALL
                        UNZ/2
                        UNWCTEST

FETCHZNB        VLOAD
                        ZNBPIP
                STCALL  UNZ/2
                        UNWCTEST

                VLOAD   VCOMP           # Z AND -X CAN'T BOTH PARALLEL UNFC/2
                        XNBPIP
                STORE   UNZ/2

# COMPUTE THE REQUIRED DIRECTION COSINE MATRIX

DCMCL           VLOAD   VXV
                        UNZ/2
                        UNX/2
                UNIT    PUSH            # UNY/2 FIRST ITERATION
                VXV     VSL1
                        UNX/2
                STORE   UNZ/2           # -UNZ/2 FIRST ITERATION
                VXSC    PDVL            # EXCHANGE -UNFVZ/2 UNZ/2 FOR UNY/2
                        UNFVZ/2         # MUST BE SMALL
                VXSC    BVSU            # YIELDS -UNFVY/2 UNY/2-UNFVZ/2 UNZ/2
                        UNFVY/2         # MUST BE SMALL
                VSL1    VAD
                        UNX/2
                UNIT                    # TOTALLY ELIMINATES THRUST POINTING ERROR
                STORE   UNX/2           # UNX/2
                VXV     VSL1
                        UNZ/2           # -UNZ/2 WAS STORED HERE REMEMBER
                STORE   UNY/2           # UNY/2
                VCOMP   VXV
                        UNX/2
                VSL1
                STORE   UNZ/2           # UNZ/2
```

```
# COMPUTES THE REQUIRED GIMBAL ANGLES

                CALL
                        NB2CDUSP        # YIELDS THE RQD GIMBAL ANGLES, 2'S, PI
                EXIT

# LIMIT THE MIDDLE GIMBAL ANGLE & COMPUTE THE UNLIMITED GIMBAL ANGLE CHGS

                CA      MPAC +2         # LIMIT THE MGA
                TS      L               # CAN'T LXCH:  NEED UNLIMITED MGA FOR ALARM
                CA      CDUZDLIM
                TC      LIMITSUB        # YIELDS LIMITED MGA.  1 BIT ERROR POSSIBLE
                XCH     MPAC +2         #       BECAUSE USING 2'S COMP.  WHO CARES?
                EXTEND
                MSU     MPAC +2         # THIS BETTER YIELD ZERO
                EXTEND
                BZF     +2
                TCF     ALARMMGA

MGARET          INHINT                  # RELINT AT TC INTPRET AFTER TCQCDUW

                ZL
                CA      TWO
DELGMBLP        TS      TEM2

                CA      L               # TO PREVENT FALSE STARTS ABOUT X, ZERO
                EXTEND                  #       FLAGOODW IF DELGMBZ OR Y TOO BIG.
                SQUARE
                AD      HI5             # WITHIN 1 BIT OF -(45 DEG SQUARED)
                EXTEND
                BZMF    +3
                CA      ZERO
                TS      FLAGOODW

                INDEX   TEM2
                CA      MPAC
                INDEX   TEM2
                TS      CPHI            # OUTPUTS TO NOUN22
                EXTEND
                INDEX   TEM2
                MSU     CDUXD           # NO MATTER THAT THESE SLIGHTLY DIFFERENT
                COM                     # FROM WHEN WE INITIALLY FETCHED THEM
                INDEX   TEM2
                TS      -DELGMB         # -UNLIMITED GIMBAL ANGLE CHGS, 1'S, PI
                TS      L               # FOR PRECEDING TEST ON NEXT LOOP PASS
                CCS     TEM2
                TCF     DELGMBLP
```

```
# BRANCHES TO NOATTCNT
                CCS     FLPAUTNO
                TCF     NOATTCNT +2     # NO PNGCS AUTO

                CA      FLAGWRD5
                MASK    ENGONBIT
                EXTEND
                BZF     NOATTCNT +2     # ENGINE NOT ON
```

```
# LIMIT THE ATTITUDE ANGLE CHANGES
#
# THIS SECTION LIMITS THE ATTITUDE ANGLE CHANGES ABOUT A SET OF ORTHOGONAL VEHICLE AXES X,YPRIME,ZPRIME,
# THESE AXES COINCIDE WITH THE COMMANDED VEHICLE AXES IF AND ONLY IF CDUXD IS ZERO.  THE PRIME SYSTEM IS
# THE COMMANDED VEHICLE SYSTEM ROTATED ABOUT THE X AXIS TO BRING THE Z AXIS INTO ALIGNMENT WITH THE MIDDLE GIMBAL
# AXIS.  ATTITUDE ANGLE CHANGES IN THE PRIME SYSTEM ARE RELATED TO SMALL GIMBAL ANGLE CHANGES BY:
#
#        [   -DELATTX    ]   [ 1      SIN(CDUZD)      0 ] [ -DELGMBX ]
#        [               ]   [                          ] [          ]
#        [ -DELATTYPRIME ] = [ 0      COS(CDUZD)      0 ] [ -DELGMBY ]
#        [               ]   [                          ] [          ]
#        [ -DELATTZPRIME ]   [ 0          0           1 ] [ -DELGMBZ ]

                LXCH    -DELGMB +2      # SAME AS -DELATTZPRIME UNLIMITED
                INDEX   NDXCDUW
                CA      DAZMAX
                TC      LIMITSUB
                TS      -DELGMB +2      # -DELGMBZ

                CA      -DELGMB +1
                EXTEND
                MP      COSCDUZ         # YIELDS -DELATTYPRIME/2 UNLIMITED
                TS      L
                INDEX   NDXCDUW
                CA      DAY/2MAX
                TC      LIMITSUB
                EXTEND
                DV      COSCDUZ
                XCH     -DELGMB +1      # -DELGMBY, FETCHING UNLIMITED VALUE

                EXTEND
                MP      SINCDUZ
                DDOUBL
                COM
                EXTEND                  # YIELDS +DELATTX UNLIMITD, MAG < 180 DEG.
                MSU     -DELGMB         #       BASED ON UNLIMITED DELGMBV.
                TS      L               #       ONE BIT ERROR IF OPERANDS IN MSU
                INDEX   NDXCDUW         #       OF MIXED SIGNS.  WHO CARES?
                CA      DAXMAX
                TC      LIMITSUB
                TS      -DELGMB         # SAVE LIMITED +DELATTX
                CCS     FLAGOODW
                CS      -DELGMB         # FETCH IT BACK CHGING SIGN IF WINDOW GOOD
                TS      -DELGMB         # OTHERWISE USE ZERO FOR -DELATTX
                CS      -DELGMB +1
                EXTEND
                MP      SINCDUZ
                DDOUBL                  # YIELDS -CNTRIB TO -DELATTX FROM -DELGMBY
                ADS     -DELGMB         # -DELGMBX.  NO OVERFLOW SINCE LIMITED TO
                                        # 20DEG(1+SIN(70DEG)/COS(70DEG)) < 180DEG
```

```
# COMPUTE COMMANDED ATTITUDE RATES
#
#       [ OMEGAPD ]   [ -2         -4 SINCDUZ              +0    ] [ -DELGMBZ ]
#       [         ]   [                                          ] [          ]
#       [ OMEGAQD ] = [ +0     -8 COSCDUZ COSCDUX     -4 SINCDUX ] [ -DELGMBY ]
#       [         ]   [                                          ] [          ]
#       [ OMEGARD ]   [ +0     +8 COSCDUZ SINCDUX     -4 COSCDUX ] [ -DELGMBZ ]
#
# ATTITUDE ANGLE RATES IN UNITS OF PI/4 RAD/SEC = K TRIG FCNS IN UNITS OF 2 X GIMBAL ANGLE RATES IN UNITS OF
# PI/2 RAD/SEC.  THE CONSTANTS ARE BASED ON DELGMB BEING THE GIMBAL ANGLE CHANGES IN UNITS OF PI RADIANS,
# AND 2 SECONDS BEING THE COMPUTATION PERIOD (THE PERIOD BETWEEN SUCCESSIVE PASSES THRU FINDCDUW).

                CS      -DELGMB
                TS      OMEGAPD
                CS      -DELGMB +1
                EXTEND
                MP      SINCDUZ
                DDOUBL
                ADS     OMEGAPD
                ADS     OMEGAPD

                CS      -DELGMB +1
                EXTEND
                MP      COSCDUX
                DDOUBL
                EXTEND
                MP      COSCDUZ
                TS      OMEGAQD
                CS      -DELGMB +2
                EXTEND
                MP      SINCDUX
                ADS     OMEGAQD
                ADS     OMEGAQD
                ADS     OMEGAQD

                CA      -DELGMB +1
                EXTEND
                MP      SINCDUX
                DDOUBL
                EXTEND
                MP      COSCDUZ
                TS      OMEGARD
                CS      -DELGMB +2
                EXTEND
                MP      COSCDUX
                ADS     OMEGARD
                ADS     OMEGARD
                ADS     OMEGARD
```

```
# FINAL TRANSFER

                CA      TWO
CDUWXFR         TS      TEM2
                INDEX   TEM2
                CA      -DELGMB
                EXTEND
                MP      DT/DELT         # RATIO OF DAP INTERVAL TO CDUW INTERVAL
                TC      ONESTO2S
                INDEX   TEM2
                TS      DELCDUX         # ANGLE INTERFACE

                INDEX   TEM2
                CCS     OMEGAPD
                AD      ONE
                TCF     +2
                AD      ONE
                EXTEND                  # WE NOW HAVE ABS(OMEGAPD,QD,RD)
                INDEX   TEM2
                MP      OMEGAPD
                EXTEND
                MP      BIT11           # 1/16
                EXTEND
                INDEX   TEM2            #                  2
                DV      1JACC           # UNITS PI/4 RAD/SEC
                TS      L
                CA      DELERLIM
                TC      LIMITSUB
                INDEX   TEM2
                TS      DELPEROR        # LAG ANGLE = OMEGA ABS(OMEGA)/2 ACCEL
                CCS     TEM2
                TCF     CDUWXFR

# HAUSKEEPING AND RETURN

TCQCDUW         CA      ECDUWUSR
                TS      EBANK           # RETURN USER'S EBANK

                TC      INTPRET
                SETPD   GOTO
                        0
                        QCDUWUSR        # NORMAL AND ABNORMAL RETURN TO USER
```

```
# THRUST VECTOR FILTER SUBROUTINE

FLTRSUB         EXTEND
                QXCH    TEM2
                TS      TEM3            # SAVE ORIGINAL OFFSET
                COM                     # ONE MCT, NO WDS, CAN BE SAVED IF NEG OF
                AD      L               #       ORIG OFFSET ARRIVES IN A, BUT IT'S
                EXTEND                  #       NOT WORTH THE INCREASED OBSCURITY.
                INDEX   NDXCDUW
                MP      GAINFLTR
                TS      L               # INCR TO OFFSET, UNLIMITED
                CA      DUNFVLIM        # SAME LIMIT FOR Y AND Z
                TC      LIMITSUB        # YIELDS INCR TO OFFSET, LIMITED
                AD      TEM3            # ORIGINAL OFFSET
                TS      L               # TOTAL OFFSET, UNLIMITED
                CA      UNFVLIM         # SAME LIMIT FOR Y AND Z
                TC      LIMITSUB        # YIELDS TOTAL OFFSET, LIMITED
                TC      TEM2

# SUBR TO TEST THE ANGLE BETWEEN THE PROPOSED WINDOW AND THRUST CMD VCTS

UNWCTEST        DOT     DSQ
                        UNX/2
                DSU     BMN
                        DOTSWFMX
                        DCMCL
                SSP     RVQ             # RVQ FOR ALT CHOICE IF DOT MAGN TOO LARGE
                        FLAGOODW        #       ZEROING WINDOW GOOD FLAG
                        0
```

```
# NB2CDUSP RETURNS THE 2'S COMPLEMENT, PI, SP CDU ANGLES X,Y,Z IN MPAC,+1,+2 GIVEN THE MATRIX WHOSE ROW VECTORS
# ARE THE SEMI-UNIT NAV BASE VECTORS X,Y,X EXPRESSED IN STABLE MEMBER COORDINATES, LOCATED AT 0 IN THE PUSH LIST.
# NB2CDUSP USES THE ARCTRGSP WHICH HAS A MAXIMUM ERROR OF +-4 BITS.

NB2CDUSP        DLOAD   DSQ
                        2
                BDSU    BPL
                        DP1/4TH
                        +3
                DLOAD
                        ZEROVECS        # IN CASE SIN WAS SLIGHTLY > 1/2
                SQRT    EXIT            # YIELDS COS(CDUZ) IN UNITS OF 2

                EXTEND
                DCA     MPAC
                DDOUBL
                TS      TEM5
                TCF     +3
                CA      POSMAX          # OVERFLOW.  FETCH POSMAX, MPAC ALWAYS POS
                TS      TEM5            # COS(CDUZ) IN TEM5, UNITS 1

                INDEX   FIXLOC
                CA      2
                LXCH    MPAC
                TC      ARCTRGSP
                TS      MPAC +2         # CDUZ

                CA      ZERO
                TC      DVBYCOSM
                CA      FOUR
                TC      DVBYCOSM
                CS      TEM1
                TC      ARCTRGSP
                TS      MPAC +1         # CDUY

                CA      BIT4
                TC      DVBYCOSM
                CA      16OCT
                TC      DVBYCOSM
                CS      TEM1
                TC      ARCTRGSP
                TS      MPAC            # CDUX

                TC      INTPRET
                RVQ

16OCT           OCT     16
```

```
# THE ELEMENTS OF THE NAV BASE MATRIX WHICH WE MUST DIVIDE BY COS(MGA)
# ALREADY CONTAIN COS(MGA)/2 AS A FACTOR. THEREFORE THE QUOTIENT SHOULD
# ORDINARILY NEVER EXCEED 1/2 IN MAGNITUDE.  BUT IF THE MGA IS NEAR PI/2
# THEN COS(MGA) IS NEAR ZERO, AND THERE MAY BE SOME CHAFF IN THE OTHER
# ELEMENTS OF THE MATRIX WHICH WOULD PRODUCE CHAOS UNDER DIVISION.
# BEFORE DIVIDING WE MAKE SURE COS(MGA) IS AT LEAST ONE BIT LARGER
# THAN THE MAGNITUDE OF THE HIGH ORDER PART OF THE OPERAND.
#
# IF ONE OR MORE DIVIDES CANNOT BE PERFORMED, THIS MEANS THAT THE
# REQUIRED MGA IS VERY NEARLY +-PI/2 AND THEREFORE THE OTHER GIMBAL
# ANGLES ARE INDETERMINATE.  THE INNER AND OUTER GIMBAL ANGLES RETURNED
# IN THIS CASE WILL BE RANDOM MULTIPLES OF PI/2.

DVBYCOSM        AD      FIXLOC
                TS      ADDRWD          # ADRES OF OPERAND

                INDEX   ADDRWD          # FETCH NEG ABS OF OPERAND, AD TEM5, AND
                CA      0               #       SKIP DIVIDE IF RESULT NEG OR ZERO
                EXTEND
                BZMF    +2
                COM
                AD      TEM5            # C(A) ZERO OR NEG, C(TEM5) ZERO OR POS
                EXTEND
                BZMF    TSL&TCQ         # DIFFERENCE ALWAYS SMALL IF BRANCH

                EXTEND                  # TEM5 EXCEEDS ABS HIGH ORDER PART OF
                INDEX   ADDRWD          #       OPERAND BY AT LEAST ONE BIT.
                DCA     0               #       THEREFORE IT EXCEEDS THE DP OPERAND
                EXTEND                  #       AND DIVISION WILL ALWAYS SUCCEED.
                DV      TEM5
TSL&TCQ         TS      L
                LXCH    TEM1
                TC      Q
```

```
# ARCTRGSP RETURNS THE 2'S COMPLEMENT, PI, SP ANGLE IN THE A REGISTER GIVEN ITS SINE IN A AND ITS COSINE IN L IN
# UNITS OF 2.  THE RESULT IS AN UNAMBIGUOUS ANGLE ANYWHERE IN THE CIRCLE, WITH A MAXIMUM ERROR OF +-4 BITS.
# THE ERROR IS PRODUCED BY THE SUBROUTINE SPARCSIN WHICH IS USED ONLY IN THE REGION +-45 DEGREES.

ARCTRGSP        EXTEND
                BZF     SINZERO         # TO AVOID DIVIDING BY ZERO

                EXTEND
                QXCH    TEM4
                TS      TEM2
                CA      L
                TS      TEM3
                CA      ZERO
                EXTEND
                DV      TEM2
                EXTEND
                BZF     USECOS

                CCS     TEM3            # SIN IS SMALLER OR EQUAL
                CA      ZERO
                TCF     +4
                CS      TEM2            # IF COS NEG, REVERSE SIGN OF SIN,
                TS      TEM2            #       ANGLE = PI-ARCSIN(SIN)
                CA      NEGMAX          # PICK UP PI, 2'S COMPLEMENT
                TS      TEM3            # WE NO LONGER NEED COS
                CA      TEM2
                TC      SPARCSIN -1
                TC      ONESTO2S
                EXTEND
                MSU     TEM3
1TO2&TCQ        TC      ONESTO2S
                TC      TEM4

USECOS          CS      TEM3            # COS IS SMALLER
                TC      SPARCSIN -1     # ANGLE = SIGN(SIN)(FI/2-ARCSIN(COS))
                AD      HALF
                TS      TEM3            # WE NO LONGER NEED COS
                CCS     TEM2
                CA      TEM3
                TCF     1TO2&TCQ
                CS      TEM3
                TCF     1TO2&TCQ

SINZERO         CCS     L
                CA      ZERO
                TC      Q
                CA      NEGMAX          # PI, 2'S COMP
                TC      Q
```

```
# SPARCSIN TAKES AN ARGUMENT SCALED UNITY IN A AND RETURNS AN ANGLE SCALED
# 180 DEGREES IN A.  IT HAS BEEN UNIT TESTED IN THE REGION +-.94 (+-70
# DEGREES) AND THE MAXIMUM ERROR IS +-5 BITS WITH AN AVERAGE TIME OF
# 450 MICROSECONDS.  SPARCSIN -1 TAKES THE ARGUMENT SCALED TWO.  (BOB CRISP)

                DOUBLE
SPARCSIN        TS      SR
                TCF     +4
                INDEX   A
                CS      LIMITS
                TS      SR
                EXTEND
                MP      A
                TS      TEM1
                EXTEND
                MP      DPL9
                AD      DPL7
                EXTEND
                MP      TEM1
                AD      DPL5
                EXTEND
                MP      TEM1
                AD      DPL3
                EXTEND
                MP      TEM1
                AD      DPL1
                EXTEND
                MP      SR
                TC      Q
DPL1            DEC     10502
DPL3            DEC     432
DPL5            DEC     7300
DPL7            DEC     -11803
DPL9            DEC     8397
```

```
# LIMITSUB LIMITS THE MAGNITUDE OF THE POSITIVE OR NEGATIVE VARIABLE
# ARRIVING IN L TO THE POSITIVE LIMIT ARRIVING IN A.
# THE SIGNED LIMITED VARIABLE IS RETURNED IN A.
#
# VERSION COUTESY HUGH BLAIR-SMITH

LIMITSUB        TS      TEM1
                CA      ZERO
                EXTEND
                DV      TEM1
                CCS     A
                LXCH    TEM1
                TCF     +2
                TCF     +3
                CA      L
                TC      Q
                CS      TEM1
                TC      Q

# SUBROUTINE TO CONVERT 1'S COMP SP TO 2'S COMP

ONESTO2S        CCS     A
                AD      ONE
                TC      Q
                CS      A
                TC      Q

# NO ATTITUDE CONTROL

NOATTCNT        TC      ALARM
                OCT     00402           # NO ATTITUDE CONTROL

 +2             INHINT                  # COME HERE FOR NOATTCNT WITHOUT ALARM
                TC      IBNKCALL        # RELINT AT TC INTPRET AFTER TCQCDUW
                FCADR   STOPRATE
                TCF     TCQCDUW         # RETURN TO USER SKIPPING AUTOPILOT CMDS

# MIDDLE GIMBAL ANGLE ALARM

ALARMMGA        TC      ALARM
                OCT     00401
                TCF     MGARET
```

```
#*******************************************************************
# CONSTANTS
#*******************************************************************

# ADDRESS CONSTANTS

ECDUWL          ECADR   ECDUW

# THRUST DIRECTION FILTER CONSTANTS

GAINFLTR        DEC     .2              # GAIN FILTER SANS CSM
                DEC     .1              # GAIN FILTER WITH CSM

DUNFVLIM        DEC     .007 B-1        # 7 MR MAX CHG IN F DIR IN VEH IN 2 SECS.
                                        # THIS DOES NOT ALLOW FOR S/C ROT RATE.

UNFVLIM         DEC     .129 B-1        # 129 MR MAX THRUST OFFSET.  105 MR TRAVEL
                                        # +10MR DEFL+5MR MECH MOUNT+9MR ABLATION.

# CONSTANT RELATED TO GIMBAL ANGLE COMPUTATIONS

DOTSWFMX        DEC     .93302 B-4      # LIM COLNRTY OF UNWC/2 & UNFC/2 TO 85 DEG
                                        # LOWER PART COMES FROM NEXT CONSTANT

DAXMAX          DEC     .11111111111    # DELATTX LIM TO 20 DEG IN 2 SECS, 1'S, PI
                DEC     .0111111111     # 2 DEG WHEN CSM DOCKED

DAY/2MAX        DEC     .05555555555    # LIKEWISE FOR DELATTY
                DEC     .0055555555

DAZMAX          =       DAXMAX          # LIKEWISE FOR DELATTZ

CDUZDLIM        DEC     .3888888888     # 70 DEG LIMIT FOR MGA, 1'S, PI

# CONSTANTS FOR DATA TRANSFER

DT/DELT         DEC     .05             # .1 SEC/2 SEC WHICH IS THE AUTOPILOT
                                        # CONTROL SAMPLE PERIOD/COMPUTATION PERIOD

DELERLIM        =       DAY/2MAX        #       0 DEG LIMIT FOR LAG ANGLES, 1'S, PI

# *** END OF FLY        .132 ***
```

```
# PROGRAM NAME -- PROG52                         DATE -- JAN 9, 1967
# MOD NO -- 0                                    LOG SECTION -- P51-P53
# MODIFICATION BY -- LONSKE                      ASSEMBLY -- SUNDANCE REV 46
#
# FUNCTIONAL DESCRIPTION --
#
#       ALIGNS THE IMU TO ONE OF THREE ORIENTATIONS SELECTED BY THE ASTRONAUT.  THE PRESENT IMU ORIENTATION IS KNOWN
#       AND IS STORED IN REFSMMAT.  THE THREE POSSIBLE ORIENTATIONS MAY BE:
#
#       (A)     PREFERRED ORIENTATION
#
#               AN OPTIMUM ORIENTATION FOR A PREVIOUSLY CALCULATED MANEUVER.  THIS ORIENTATION MUST BE CALCULATED AND
#               STORED BY A PREVIOUSLY SELECTED PROGRAM.
#
#       (B)     NOMINAL ORIENTATION
#
#               X   = UNIT ( R )
#               -SM
#
#               Y   = UNIT (V X R)
#                SM
#
#               Z   = UNIT (X   X Y  )
#                SM          SM    SM
#
#               WHERE:
#
#               R = THE GEOCENTRIC RADIUS VECTOR AT TIME T(ALIGN) SELECTED BY THE ASTRONAUT
#               -
#
#               V = THE INERTIAL VELOCITY VECTOR AT TIME T(ALIGN) SELECTED BY THE ASTRONAUT
#               -
#
#       (C)     RERSMMAT ORIENTATION
#
#       (D)     LANDING SITE -- THIS IS NOT AVAILABLE IN SUNDANCE
#
#       THIS SELECTION CORRECTS THE PRESENT IMU ORIENTATION.  THE PRESENT ORIENTATION DIFFERS FROM THAT TO WHICH IT
#       WAS LAST ALIGNED ONLY DUE TO GYRO DRIFT (I.E., NEITHER GIMBAL LOCK NOR IMU POWER INTERRUPTION HAS OCCURRED
#       SINCE THE LAST ALIGNMENT).
#
#       AFTER A IMU ORIENTATION HAS BEEN SELECTED ROUTINE S52.2 IS OPERATED TO COMPUTE THE GIMBAL ANGLES USING THE
#       NEW ORIENTATION AND THE PRESENT VEHICLE ATTITUDE.  CAL52A THEN USES THESE ANGLES, STORED IN THETAD,+1,+2, TO
#       COARSE ALIGN THE IMU.  THE STARS SELECTION ROUTINE, R56, IS THEN OPERATED.  IF 2 STARS ARE NOT AVAILABLE AN ALARM
#       IS FLASHED TO NOTIFY THE ASTRONAUT.  AT THIS POINT THE ASTRONAUT WILL MANEUVER THE VEHICLE AND SELECT 2 STARS
#       EITHER MANUALLY OR AUTOMATICALLY.  AFTER 2 STARS HAVE BEEN SELECTED THE IMU IS FINE ALIGNED USING ROUTINE R51.  IF
#       THE RENDEZVOUS NAVIGATION PROCESS IS OPERATING (INDICATED BY RNDVZFLG) P20 IS DISPLAYED.  OTHERWISE P00 IS
#       REQUESTED.
#
# CALLING SEQUENCE --
#
```

```
#       THE PROGRAM IS CALLED BY THE ASTRONAUT BY DSKY ENTRY.
#
# SUBROUTINES CALLED --
#
#       1. FLAGDOWN                  7. S52.2                 13. NEWMODEX
#       2. R02BOTH                   8. CAL53A                14. PRIOLARM
#       3. GOPERF4                   9. FLAGUP
#       4. MATMOVE                  10. R56
#       5. GOFLASH                  11. R51
#       6. S52.3                    12. GOPERF3
#
# NORMAL EXIT MODES --
#
#       EXITS TO ENDOFJOB
#
# ALARM OR ABORT EXIT MODES --
#
#       NONE
#
# OUTPUT --
#
#       THE FOLLOWING MAY BE FLASHED ON THE DSKY
#               1. IMU ORIENTATION CODE
#               2. ALARM CODE 215 -- PREFERRED IMU ORIENTATION NOT SPECIFIED
#               3. TIME OF NEXT IGNITION
#               4. GIMBAL ANGLES
#               5. ALARM CODE 405 -- TWO STARS NOT AVAILABLE
#               6. PLEASE PERFORM P00
#       THE MODE DISPLAY MAY BE CHANGED TO 20
#
# ERASABLE INITIALIZATION REQUIRED --
#
#       PFRATFLG SHOULD BE SET IF A PREFERRED ORIENTATION HAS BEEN COMPUTED.  IF IT HAS BEEN COMPUTED IT IS STORED IN
#       XSMD, YSMD, ZSMD.
#
#       RNDVZFLG INDICATES WHETHER THE RENDEZVOUS NAVIGATION PROCESS IS OPERATING.
#
# DEBRIS --
#
#       WORK AREA

                BANK    33
                SETLOC  P50S
                BANK

                EBANK=  BESTI
                COUNT*  $$/P52
PROG52          TC      BANKCALL
                CADR    R02BOTH         # IMU STATUS CHECK
                CAF     PFRATBIT
                MASK    FLAGWRD2        # IS PFRATFLG SET?
                CCS     A
```

```
                TC      P52A            # YES
                CAF     BIT2            # NO
                TC      P52A +1
P52A            CAF     BIT1
                TS      OPTION2
P52B            CAF     BIT1
                TC      BANKCALL        # FLASH OPTION CODE AND ORIENTATION CODE
                CADR    GOPERF4R        # FLASH V04N06
                TC      GOTOPOOH
                TCF     +5              # V33 -- PROCEED
                TC      P52B            # NEW CODE -- NEW ORIENTATION CODE INPUT
                TC      PHASCHNG        # DISPLAY RETURN
                OCT     00014
                TC      ENDOFJOB

                CA      OPTION2
                MASK    THREE
                INDEX   A
                TC      +1
                TCF     OPT4            # OPTION 4 LANDING SITE
                TCF     P52H            # OPTION 1 PREFERRED
                TCF     P52T            # OPTION 2 NOMINAL
P52E            TC      INTPRET         # OPTION 3 REFSMMAT
                GOTO
                        P52F            # GO DO R51

OPT4            EXTEND
                DCA     TLAND           # IF OPTION 4 DISPLAY TLAND
                TCF     P52T +2

P52T            EXTEND
                DCA     NEGO
                DXCH    DSPTEM1
                CAF     V06N34*
                TC      BANKCALL
                CADR    GOFLASH
                TC      GOTOPOOH
                TC      +2
                TC      -5
                DXCH    DSPTEM1
                EXTEND
                BZMF    +2              # IF TIME ZERO OR NEG USE TIME2
                TCF     +3
                EXTEND
                DCA     TIME2
                DXCH    TALIGN
P52V            CA      OPTION2
                MASK    BIT2
                CCS     A
                TC      P52W
```

```
                TC      INTPRET         # OPTION 4 -- GET LS ORIENTATION
                GOTO
                        P52LS
```

```
P52W            TC      INTPRET
                DLOAD   CALL            # PICK UP ALIGN TIME
                        TALIGN          # COMPUTED NOMINAL IMU
                        S52.3           #       ORIENTATION
P52D            CALL                    # READ VEHICLE ATTITUDE AND
                        S52.2           #       COMPUTE GIMBAL ANGLES
                EXIT
                CAF     V06N22
                TC      BANKCALL        # DISPLAY GIMBAL ANGLES
                CADR    GOFLASH
                TC      GOTOPOOH
                TCF     COARSTYP        # V33 -- PROCEED, SEE IF GYRO TORQUE COARSE
P52H            TC      INTPRET
                GOTO
                        P52D
REGCOARS        TC      INTPRET
                CALL                    # DO COARSE ALIGN
                        CAL53A          #       ROUTINE
COARSRET        SET     CLEAR
                        REFSMFLG
                        PFRATFLG
P52F            CALL
                        R51
P52OUT          EXIT
                TC      GOTOPOOH
VB05N09         =       V05N09
V06N34*         VN      634
```

```
# CHECK FOR GYRO TORQUE COARSE ALIGNMENT

COARSTYP        CAF     OCT13
                TC      BANKCALL        # DISPLAY V 50N25 WITH COARSE ALIGN OPTION
                CADR    GOPERF1
                TCF     GOTOPOOH        # V34 -- TERMIN&OE
                TCF     REGCOARS        # V33 -- NORMAL COARSE
                TC      INTPRET         # V32 -- GYRO TORQUE COARSE
                VLOAD   MXV
                        XSMD            # GET SM(DESIRED) WRT SM(PRESENT)
                        REFSMMAT
                UNIT
                STOVL   XDC
                        YSMD
                MXV     UNIT
                        REFSMMAT
                STOVL   YDC
                        ZSMD
                MXV     UNIT
                        REFSMMAT
                STCALL  ZDC
                        GYCOARS
                GOTO
                        P52OUT
OCT13           OCT     13
```

```
# COMPUTE LANDING ORIENTATION FOR OPTION 4

P52LS           SET     CLEAR           # GET LANDING SITE ORIENTATION
                        LUNAFLAG
                        ERADFLAG        # TO PICK UP RLS
                SETPD   VLOAD
                        0
                        RLS             # PICK UP LANDING SITE VEC IN MF
                PDDL    PUSH            # RLS PD 0-5
                        TALIGN
                STCALL  TLAND           # JAM ALIGN TIME IN TLAND FOR OPTION 4
                        RP-TO-R         # TRANS RLS TO REF
                VSR2
                STODL   ALPHAV          # INPUT TO LAT-LONG
                        TALIGN
                CALL
                        N89DISP
                VLOAD   UNIT            # COMPUTE LANDING SITE ORIENT (XSMD)
                        ALPHAV
                STCALL  XSMD
                        LSORIENT
                GOTO
                        P52D            # NOW GO COMPUTE GIMBAL ANGLES.
```

```
# SUBROUTINE TO CALCULATE AND DISPLAY THE LUNAR LANDING SITE

                SETLOC  P50S1
                BANK
                EBANK=  XSM

N89DISP         STQ
                        QMAJ
                STCALL  GDT/2 +4        # TEMP STORE TIME
                        LAT-LONG
                DLOAD   SR1
                        LONG
                STODL   LANDLONG
                        ALT
                STODL   LANDALT
                        LAT
                STODL   LANDLAT
                EXIT

LSDISP          CAF     V06N89*         # DISPLAY LAT,LONG/2,ALT
                TC      BANKCALL
                CADR    GOFLASH
                TCF     GOTOPOOH        # V34 -- TERMINATE -- EXIT P57
                TCF     +2              # V33 -- PROCEED -- ACCEPT LS DATA
                TCF     LSDISP          # V32 OR E -- LOOK AGAIN AND/OR LOAD NEW LS

                TC      INTPRET
                DLOAD   SL1
                        LANDLONG
                STODL   LONG
                        LANDALT
                STODL   ALT
                        LANDLAT
                STODL   LAT
                        GDT/2 +4        # PICK UP TIME
                CALL                    # GET RLS BACK FROM LAT,LONG,ALT
                        LALOTORV        # RLS B-29 IN MPAC AND ALPHAV
                GOTO
                        QMAJ
V06N89*         VN      689
```

```
# NAME -- S50 ALIAS LOCSAM
# BY
# VINCENT
#
# FUNCTION -- COMPUTE INPUTS FOR PICAPAR AND PLANET
#
#       DEFINE
#
#       U   = UNIT( SUN WRT EARTH )
#        ES
#
#       U   = UNIT( MOON WRT EARTH )
#        EM
#
#       R   = POSITION VECTOR OF LEM
#        L
#
#       R   = MEAN DISTANCE (384402KM) BETWEEN EARTH AND MOON
#        EM
#
#       P   = RATIO R  /(DISTANCE SUN TO EARTH) > .00257125
#                    EM
#
#       R   = EQUATORIAL RADIUS (6378.166KM) OF EARTH
#        E
#
#       LOCSAM COMPUTES IN EARTH INFLUENCE
#
#       VSUN = U
#               ES
#
#       VEARTH = -UNIT( R  )
#                        L
#
#       VMOON = UNIT(R   .U    - R  )
#                     EM   EM     L
#
#       CSUN = COS 90
#
#       CEARTH = COS(5 + ARCSIN(R /MAG(R )))
#                               E      L
#
#       CMOON    = COS 5
#
# INPUT -- TIME IN MPAC
#
# OUTPUT -- LISTED ABOVE
#
# SUBROUTINES -- LSPOS, LEMPREC
#
# DEBRIS -- VAC AREA, TSIGHT
```

```
                COUNT*   $$/LOSAM

S50             =        LOCSAM
LOCSAM          STQ
                         QMIN
                STCALL   TSIGHT
                         LSPOS
                DLOAD
                         TSIGHT
                STCALL   TDEC1
                         LEMPREC
                SSP      TIX,2
                         S2
                         0
                         MOONCNTR
EARTCNTR        VLOAD    VXSC
                         VMOON
                         RSUBEM
                VSL1     VSU
                         RATT
                UNIT
                STOVL    VMOON
                         RATT
                UNIT     VCOMP
                STODL    VEARTH
                         RSUBE
                CALL
                         OCCOS
                STODL    CEARTH
                         CSS5
                STCALL   CMOON
                         ENDSAM
MOONCNTR        VLOAD    VXSC
                         VMOON
                         ROE
                BVSU     UNIT
                         VSUN
                STOVL    VSUN
                         VMOON
                VXSC     VAD
                         RSUBEM
                         RATT
                UNIT     VCOMP
                STOVL    VEARTH
                         RATT
                UNIT     VCOMP
                STODL    VMOON
                         RSUBM
                CALL
                         OCCOS
```

```
                STODL   CMOON
                        CSS5
                STORE   CEARTH
ENDSAM          DLOAD
                        CSSUN
                STORE   CSUN
                GOTO
                        QMIN
OCCOS           DDV     SR1
                        36D
                ASIN    DAD
                        5DEGREES
                COS     SR1
                RVQ
CEARTH          =       14D
CSUN            =       16D
CMOON           =       18D
CSS5            2DEC    .2490475        # (COS 5)/4
CSSUN           2DEC    .125            # (COS 60)/4
5DEGREES        2DEC    .013888889      # SCALED IN REVS
```

```
# PROGRAM NAME -- R56               DATE: DEC 20 66
# MOD 1                             LOG SECTION: P51-P53
#                                   ASSEMBLY:  SUNDISK REV4D
# BY KEN VINCENT
#
# FUNCTION
#       THIS PROGRAM READS THE IMU-CDUS AND COMPUTES THE VEHICLE ORIENTATION
#       WITH RESPECT TO INERTIAL SPACE.  IT THEN COMPUTES THE SHAFT AXIS (SAX)
#       WITH RESPECT TO REFERENCE INTERTIAL.  EACH STAR IN THE CATALOG IS TESTED
#       TO DETERMIN IF IT IS OCCULTED BY EITHER EARTH, SUN OR MOON.  IF A
#       STAR IS NOT OCCULTED THEN IT IS PAIRED WITH ALL STARS OF LOWER INDEX.
#       THE PAIRED STAR IS TESTED FOR OCCULTATION.  PAIRS OF STARS THAT PASS
#       THE OCCULTATION TESTS ARE TESTED FOR GOOD SEPARATION.  A PAIR OF STARS
#       HAVE GOOD SEPARATION IF THE ANGLE BETWEEN THEM IS LESS THAN 100 DEGREES
#       AND MORE THAN 50 DEGREES.  THOSE PAIRS WITH GOOD SEPARATION
#       ARE THEN TESTED TO SEE IF THEY LIE IN CURRENT FIELD OF VIEW.  (WITHIN
#       50 DEGREES OF SAX).  THE PAIR WITH MAX SEPARATION IS CHOSEN FROM
#       THOSE WITH GOOD SEPARATION, AND IN FIELD OF VIEW.
#
# CALLING SEQUENCE
#       L       TC      BANKCALL
#       L+1     CADR    R56
#       L+2     ERROR RETURN -- NO STARS IN FIELD OF VIEW
#       L+3     NORMAL RETURN
#
# OUTPUT
#       BESTI, BESTJ -- SINGLE PREC, INTEGERS, STAR NUMBERS TIMES 6
#       VFLAG -- FLAG BIT SET IMPLIES NO STARS IN FIELD OF VIEW
#
# INITIALIZATION
#       1)      A CALL TO LOCSAM MUST BE MADE
#
# DEBRIS
#       WORKAREA
#       X,Y,ZNB
#       SINCDU, COSCDU
#       STARAD -- STAR +5

R56             =       PICAPAR
                COUNT*  $$/R56
PICAPAR         TC      MAKECADR
                TS      QMIN
                TC      INTPRET
                CALL
                        CDUTRIG
                CALL
                        CALCSMSC
                SETPD
                        0
                SET     DLOAD           # VFLAG = 1
                        VFLAG
```

```
                                DPZERO
                        STOVL   BESTI
                                XNB
                        VXSC    PDVL
                                HALFDP
                                ZNB
                        AXT,1   VXSC
                                228D            # X1 = 37 X 6 + 6
                                HALFDP
                        VAD
                        VXM     UNIT
                                REFSMMAT
                        STORE   SAX             # SAX = SHAFT AXIS
                        SSP     SSP             # S1 = S2 = 6
                                S1
                                6
                                S2
                                6
        PIC1            TIX,1   GOTO            # MAJOR STAR
                                PIC2
                                PICEND
        PIC2            VLOAD*  DOT
                                CATLOG,1
                                SAX
                        DSU     BMN
                                CSS33
                                PIC1
                        LXA,2

                                X1
        PIC3            TIX,2   GOTO
                                PIC4
                                PIC1
        PIC4            VLOAD*  DOT
                                CATLOG,2
                                SAX
                        DSU     BMN
                                CSS33
                                PIC3
                        VLOAD*  DOT*
                                CATLOG,1
                                CATLOG,2
                        DSU     BPL
                                CSS40
                                PIC3
                        VLOAD*  CALL
                                CATLOG,1
                                OCCULT
                        BON

                                CULTFLAG
                                PIC1
```

```
                VLOAD*  CALL
                        CATLOG,2
                        OCCULT
                BON
                        CULTFLAG
                        PIC3
STRATGY         BONCLR
                        VFLAG
                        NEWPAR
                XCHX,1  XCHX,2
                        BESTI
                        BESTJ
STRAT           VLOAD*  DOT*
                        CATLOG,1
                        CATLOG,2
                PUSH    BOFINV
                        VFLAG
                        STRAT -3
                DLOAD   DSU
                BPL
                        PIC3
NEWPAR          SXA,1   SXA,2
                        BESTI
                        BESTJ
                GOTO
                        PIC3
OCCULT          MXV     BVSU
                        CULTRIX
                        CSS
                BZE
                        CULTED
                BMN     SIGN
                        CULTED
                        MPAC +3
                BMN     SIGN
                        CULTED
                        MPAC +5
                BMN     CLRGO
                        CULTED
                        CULTFLAG
                        QPRET
CULTED          SETGO
                        CULTFLAG
                        QPRET
CSS             =       CEARTH
CSS40           2DEC    .16070          # COS 50 / 4
CSS33           2DEC    .16070          # COS 50 / 4
PICEND          BOFF    EXIT
```

```
                              VFLAG
                              PICGXT
                     TC       PICBXT
PICGXT               LXA,1    LXA,2
                              BESTI
                              BESTJ
                     VLOAD    DOT*
                              SAX
                              CATLOG,1
                     PDVL     DOT*
                              SAX
                              CATLOG,2
                     DSU
                     BPL      SXA,1
                              PICNSWP
                              BESTJ
                     SXA,2
                              BESTI
PICNSWP              EXIT
                     INCR     QMIN
PICBXT               CA       QMIN
                     TC       SWCALL
VPD                  =        0D
V0                   =        6D
V1                   =        12D
V2                   =        18D
V3                   =        24D
DP0                  =        30D
DP1                  =        32D
```

```
# NAME -- R51   FINE ALIGN
# FUNCTION -- TO ALIGN THE STABLE MEMBER TO REFSSMAT
# CALLING SEQ -- CALL R51
# INPUT -- REFSMMAT
# OUTPUT -- GYRO TORQUE PULSES
# SUBROUTINES -- LOCSAM, PICAPAR, R52, R53, R54, R55

                COUNT*  $$/R51
R51             STQ
                        QMAJ
R51.1           EXIT
                TC      PHASCHNG
                OCT     04024

R51C            CAF     OCT15
                TC      BANKCALL
                CADR    GOPERF1
                TC      GOTOPOOH
                TC      +2              # V33E
                TC      R51E            # ENTER
                TC      INTPRET
                RTB     DAD
                        LOADTIME
                        TSIGHT1
                CALL
                        LOCSAM
                EXIT
                TC      BANKCALL
                CADR    R56
                TC      R51I
R51F            TC      R51E
R51I            TC      ALARM
                OCT     405
                CAF     VB05N09
                TC      BANKCALL
                CADR    GOFLASH
                TC      GOTOPOOH
                TC      R51E
                TC      R51C
R51E            CAF     ZERO
                TS      STARIND
R51.2           TC      INTPRET
R51.3           EXIT
                TC      PHASCHNG
                OCT     04024

                TC      INTPRET
                CALL
                        R52             # AOP WILL MAKE CALLS TO SIGHTING
                EXIT
                TC      BANKCALL
```

```
                CADR    AOTMARK
                TC      BANKCALL
                CADR    OPTSTALL
                TC      CURTAINS
                CCS     STARIND
                TCF     +2
                TC      R51.4
                TC      INTPRET
                VLOAD
                        STARAD +6
                STORE   STARSAV2
                EXIT
                TC      PHASCHNG
                OCT     04024

                TC      INTPRET
                DLOAD   CALL
                        TSIGHT
                        PLANET
                MXV     UNIT
                        REFSMMAT
                STOVL   STARAD +6
                        PLANVEC
                MXV     UNIT
                        REFSMMAT
                STOVL   STARAD
                        STARSAV1
                STOVL   6D
                        STARSAV2
                STCALL  12D
                        R54             # STAR DATA TEST
                BOFF    CALL
                        FREEFLAG
                        R51K
                        AXISGEN
                CALL
                        R55             # GYRO TORQUE
                CLEAR
                        PFRATFLG
R51K            EXIT
R51P63          CAF     OCT14
                TC      BANKCALL
                CADR    GOPERF1
                TC      GOTOPOOH
                TC      R51C
                TC      INTPRET
                GOTO
                        QMAJ
R51.4           TC      INTPRET
                VLOAD
```

```
                                STARAD +6
                        STORE   STARSAV1
                        DLOAD   CALL
                                TSIGHT
                                PLANET
                        STORE   PLANVEC
                        SSP
                                STARIND
                                1
                        GOTO
                                R51.3
TSIGHT1                 2DEC    36000           # 6 MIN TO MARKING
```

```
# GYRO TORQUE COARSE ALGNMENT

GYCOARS         STQ     CALL
                        QMAJ
                        CALCGTA
                CLEAR   CLEAR
                        DRIFTFLG
                        REFSMFLG
                EXIT
                CAF     V16N20          # MONITOR GIMBALS
                TC      BANKCALL
                CADR    GODSPR
                CA      R55CDR
                TC      BANKCALL
                CADR    IMUPULSE
                TC      BANKCALL
                CADR    IMUSTALL
                TC      CURTAINS
                TC      PHASCHNG
                OCT     04024

                TC      INTPRET
                AXC,1   AXC,2
                        XSMD
                        REFSMMAT
                CALL                    # STORE DESIRED REFSMMAT
                        MATMOVE
                CLEAR   SET
                        PFRATFLG
                        REFSMFLG
                CALL
                        NCOARSE         # SET DRIFT AND INITIALIZE 1/PIPADT
                GOTO
                        R51K
V16N20          VN      1620
```

```
# R55   GYRO TORQUE
# FUNCTION -- COMPUTE AND SEND GYRO PULSES
# CALLING SEQ -- CALL R55
# INPUT -- X,Y,ZDC -- REFSMMAT WRT PRESENT STABLE MEMBER
# OUTPUT -- GYRO PULSES
# SUBROUTINES -- CALCGTA, GOFLASH, GODSPR, IMUFINE, IMUPULSE, GOPERF1

                COUNT*  $$/R55
R55             STQ
                        QMIN
                CALL
                        CALCGTA
PULSEM          EXIT
R55.1           CAF     V06N93
                TC      BANKCALL
                CADR    GOFLASH
                TC      GOTOPOOH
                TC      R55.2
                TC      R55RET
R55.2           TC      PHASCHNG
                OCT     00214
                CA      R55CDR
                TC      BANKCALL
                CADR    IMUPULSE
                TC      BANKCALL
                CADR    IMUSTALL
                TC      CURTAINS
                TC      PHASCHNG
                OCT     04024

R55RET          TC      INTPRET
                GOTO
                        QMIN
V06N93          VN      0693
R55CDR          ECADR   OGC
R54             =       CHKSDATA

# ROUTINE NAME -- CHKSDATA              DATE -- JAN 9, 1967
# MOD NO -- 0                           LOG SECTION -- P51-P53
# MODIFICATION BY -- LONSKE             ASSEMBLY --
#
# FUNCTIONAL DESCRIPTION -- CHECKS THE VALIDITY OF A PAIR OF STAR SIGHTINGS.  WHEN A PAIR OF STAR SIGHTINGS ARE MADE
# BY THE ASTRONAUT THIS ROUTINE OPERATES AND CHECKS THE OBSERVED SIGHTINGS AGAINST STORED STAR VECTORS IN THE
# COMPUTER TO INSURE A PROPER SIGHTING WAS MADE.  THE FOLLOWING COMPUTATIONS ARE PERFORMED --
#       OS1     =       OBSERVED STAR 1 VECTOR
#       OS2     =       OBSERVED STAR 2 VECTOR
#       SS1     =       STORED STAR 1 VECTOR
#       SS2     =       STORED STAR 2 VECTOR
#        A1     =       ARCCOS(OS1 - OS2)
#        A2     =       ARCCOS(SS1 - SS2)
#         A     =       ABS(2(A1 - A2))
```

```
# THE ANGULAR DIFFERENCE IS DISPLAYED FOR ASTRONAUT ACCEPTANCE.
#
# EXIT MODE --  1. FREEFLAG SET IMPLIES ASTRONAUT WANTS TO PROCEED
#               2. FREEFLAG RESET IMPLIES ASTRONAUT WANTS TO RECYCLE
#
# OUTPUT --     1. VERB 6,NOUN 3 -- DISPLAYS ANGULAR DIFFERENCE BETWEEN 2 SETS OF STARS.
#               2. STAR VECTORS FROM STAR CATALOG ARE LEFT IN 6D AND 12D.
#
# ERASABLE INITIALIZATION REQUIRED --
#               1. MARK VECTORS ARE STORED IN STARAD AND STARAD +6.
#               2. CATALOG VECTORS ARE STORED IN 6D AND 12D.
#
# DEBRIS --

                COUNT*  $$/R54
CHKSDATA        STQ     SET
                        QMIN
                        FREEFLAG
CHKSAB          AXC,1                   # SET X1 TO STORE EPHEMERIS DATA
                        STARAD

CHKSB           VLOAD*  DOT*            # CAL. ANGLE THETA
                        0,1
                        6,1
                SL1     ACOS
                STORE   THETA
                BOFF    INVERT          # BRANCH TO CHKSD IF THIS IS 2ND PASS
                        FREEFLAG
                        CHKSD
                        FREEFLAG        # CLEAR FREEFLAG
                AXC,1   DLOAD           # SET X1 TO MARK ANGLES
                        6D
                        THETA
                STORE   18D
                GOTO
                        CHKSB           # RETURN TO CAL. 2ND ANGLE
CHKSD           DLOAD   DSU
                        THETA
                        18D
                ABS     RTB             # COMPUTE POS DIFF
                        SGNAGREE
                STORE   NORMTEM1
                SET     EXIT
                        FREEFLAG
                CAF     VB6N5
                TC      BANKCALL
                CADR    GOFLASH
                TCF     GOTOPOOH
                TC      CHKSDA          # PROCEED
                TC      INTPRET
                CLEAR   GOTO
                        FREEFLAG
                        QMIN
CHKSDA          TC      INTPRET
```

```
                GOTO
                        QMIN
VB6N5           VN      605

# NAME -- CAL53A
# FUNCTION -- COMPUTE DESIRED GIMBAL ANGLES AND COARSE ALIGN IF NECESSARY
# CALLING SEQUENCE -- CALL CAL53A
# INPUT -- X,Y,ZSMD, CDUX,Y,Z
#          DESIRED GIMBAL ANGLES -- THETAD,+1,+2
# OUTPUT -- THE IMU COORDINATES AT STORED IN REFSMMAT
# SUBROUTINES -- S52.2, IMUCOARSE, IMUFINE

                COUNT*  $$/R50
CAL53A          CALL
                        S52.2           # MAKE ONE FINAL COMP OF GIMBAL ANGLES
                RTB     SSP
                        RDCDUS          # READ CDUS
                        S1
                        1
                AXT,1   SETPD
                        3
                        4
CALOOP          DLOAD*  SR1
                        THETAD +3D,1
                PDDL*   SR1
                        4,1
                DSU     ABS
                PUSH    DSU
                        DEGREE1
                BMN     DLOAD
                        CALOOP1
                DSU     BPL
                        DEG359
                        CALOOP1
                EXIT
                TC      PHASCHNG
                OCT     04024

                TC      INTPRET
COARFINE        CALL
                        COARSE
                CALL
                        NCOARSE
                GOTO
                        FINEONLY
CALOOP1         TIX,1
                        CALOOP
FINEONLY        AXC,1   AXC,2
                        XSM
                        REFSMMAT
                CALL
                        MATMOVE
```

```
                GOTO
                        COARSRET
MATMOVE         VLOAD*                          # TRANSFER MATRIX
                        0,1
                STORE   0,2
                VLOAD*
                        6D,1
                STORE   6D,2
                VLOAD*
                        12D,1
                STORE   12D,2
                RVQ
DEGREE1         DEC     46                      # 1 DEG SCALED CDU/2
DEG359          DEC     16338                   # 359 DEG SCALED CDU/2
RDCDUS          INHINT                          # READ CDUS
                CA      CDUX
                INDEX   FIXLOC
                TS      1
                CA      CDUY
                INDEX   FIXLOC
                TS      2
                CA      CDUZ
                INDEX   FIXLOC
                TS      3
                RELINT
                TC      DANZIG
                COUNT*  $$/INFLT
```

```
# NAME -- P51 -- IMU ORIENTATION DETERMINATION
# MOD. NO. 1    23 JAN 67                              LOG SECTION -- P51-P53
# MOD BY STURLAUGSON                                   ASSEMBLY SUNDANCE REV56
#
# FUNCTIONAL DESCRIPTION
#       DETERMINES THE INERTIAL ORIENTATION OF THE IMU.  THE PROGRAM IS SELECTED BY DSKY ENTRY.  THE SIGHTING
#       (AOTMARK) ROUTINE IS CALLED TO COLLECT AND PROCESS MARKED-STAR DATA.  AOTMARK (R53) RETURNS THE STAR NUMBER AND THE
#       STAR LOS VECTOR IN STARAD +6.  TWO STARS ARE THUS SIGHTED.  THE ANGLE BETWEEN THE TWO STARS IS THEN CHECKED AT
#       CHKSDATA (R54).  REFSMMAT IS THEN COMPUTED AT AXISGEN.
#
# CALLING SEQUENCE
#       THE PROGRAM IS CALLED BY THE ASTRONAUT BY DSKY ENTRY.
#
# SUBROUTINES CALLED
#       GOPERF3
#       GOPERF1
#       GODSPR
#       IMUCOARS
#       IMUFIN20
#       AOTMARK (R53)
#       CHKSDATA (R54)
#       MKRELEAS
#       AXISGEN
#       MATMOVE
#
# ALARMS
#       NONE.
#
# ERASABLE INITIALIZATION
#       IMU ZERO FLAG SHOULD BE SET.
#
# OUTPUT
#       REFSMMAT
#       REFSMFLG
#
# DEBRIS
#       WORK AREA
#       STARAD
#       STARIND
#       BESTI
#       BESTJ

                COUNT*  $$/P51
```

```
P51             TC      BANKCALL        # IS ISS ON - IF NOT, IMUCHK WILL SEND
                CADR    IMUCHK          # ALARM CODE 210 AND EXIT VIA GOTOPOOH.

                CAF     OCT15
                TC      BANKCALL
                CADR    GOPERF1
                TC      GOTOPOOH        # TERM.
                TCF     P51B            # V33
                TC      PHASCHNG
                OCT     04024

                CAF     ZERO
                TS      THETAD          # ZERO THE GIMBALS
                TS      THETAD +1
                TS      THETAD +2
                CAF     V06N22
                TC      BANKCALL
                CADR    GODSPRET
                CAF     V41K            # NOW DISPLAY COARSE ALIGN VERB 41
                TC      BANKCALL
                CADR    GODSPRET
                TC      INTPRET
                CALL
                        COARSE
                EXIT
                TC      PHASCHNG
                OCT     04024
                TCF     P51 +2

P51B            TC      PHASCHNG
                OCT     00014
                TC      INTPRET
                CALL
                        NCOARSE
                SSP     SETPD
                        STARIND         # INDEX -- STAR 1 OR 2
                        0
                        0
P51C            EXIT
                TC      PHASCHNG
                OCT     04024

                TC      BANKCALL
                CADR    AOTMARK         # R53
                TC      BANKCALL
                CADR    AOTSTALL
                TC      CURTAINS
                CCS     STARIND
                TCF     P51D +1
                TC      INTPRET
```

```
                VLOAD
                        STARAD +6
                STORE   STARSAV1
P51D            EXIT
                TC      PHASCHNG
                OCT     04024

                CCS     STARIND
                TCF     P51E
                TC      PHASCHNG
                OCT     04024

                TC      INTPRET
                DLOAD   CALL
                        TSIGHT
                        PLANET
                STORE   PLANVEC
                EXIT
                CAF     BIT1
                TS      STARIND
                TCF     P51C +1         # DO SECOND STAR
P51E            TC      PHASCHNG
                OCT     04024

                TC      INTPRET
                DLOAD   CALL
                        TSIGHT
                        PLANET
                STOVL   12D
                        PLANVEC
                STOVL   6D
                        STARSAV1
                STOVL   STARAD
                        STARSAV2
                STCALL  STARAD +6
                        CHKSDATA        # CHECK STAR ANGLES IN STARAD AND
                BON     EXIT
                        FREEFLAG
                        P51G
                TC      P51 +2
P51G            CALL
                        AXISGEN         # COME BACK WITH REFSMMAT IN XDC
                AXC,1   AXC,2
                        XDC
                        REFSMMAT
                CALL
                        MATMOVE
                SET     EXIT
                        REFSMFLG
                TC      GOTOPOOH        # FINIS
```

```
V41K            VN      4100
COARSE          EXIT
                TC      BANKCALL
                CADR    IMUCOARS
                TC      BANKCALL
                CADR    IMUSTALL
                TC      CURTAINS
                TC      BANKCALL
                CADR    IMUFINE
                TC      BANKCALL
                CADR    IMUSTALL
                TC      CURTAINS
                TC      INTPRET
                RVQ
NCOARSE         EXIT
                CA      TIME1
                TS      1/PIPADT
                CS      ZERO
                TS      PIPAX
                TS      PIPAY
                TS      PIPAZ
                TC      INTPRET
                VLOAD
                        ZEROVEC
                STORE   GCOMP
                SET     RVQ
                        DRIFTFLG
```

```
# NAME -- S52.2
# FUNCTION -- COMPUTE GIMBAL ANGLES FOR DESIRED SM AND PRESENT VEHICLE
# CALL -- CALL S52.2
# INPUT -- X,Y,ZSMD
# OUTPUT -- OGC,IGC,MGC,THETAD,+1,+2
# SUBROUTINES -- CDUTRIG, CALCSMSC, MATMOVE, CALCGA

                COUNT*  $$/S52.1
S52.2           STQ     CALL
                        QMAJ
                        CDUTRIG
                CALL
                        CALCSMSC
                AXT,1   SSP
                        18D
                        S1
                        6D
S52.2A          VLOAD*  VXM
                        XNB +18D,1
                        REFSMMAT
                UNIT
                STORE   XNB +18D,1
                TIX,1
                        S52.2A
S52.2.1         AXC,1   AXC,2
                        XSMD
                        XSM
                CALL
                        MATMOVE
                CALL
                        CALCGA
                GOTO
                        QMAJ
```

```
# NAME -- S52.3
# FUNCTION --    XSMD= UNIT R
#                YSMD= UNIT(V X R)
#                ZSMD= UNIT(XSMD X YSMD)
# CALL --        DLOAD   CALL
#                        TALIGN
#                        S52.3
# INPUT --       TIME OF ALIGNMENT IN MPAC
# OUTPUT --      X,Y,ZSMD
# SUBROUTINES -- CSMCONIC

                COUNT*  $$/S52.3
S52.3           STQ
                        QMAJ
                STCALL  TDEC1
                        LEMCONIC
                VLOAD   UNIT
                        RATT
                STOVL   XSMD
                        VATT
                VXV     UNIT
                        RATT
                STOVL   YSMD
                        XSMD
                VXV     UNIT
                        YSMD
                STCALL  ZSMD
                        QMAJ
```

```
# NAME -- R52 (AUTOMATIC OPTICS POSITIONING ROUTINE)
#
# FUNCTION -- POINT THE AOT APTICS AXIS BY MANEUVERING THE LEM TO A NAVIGATION
# STAR SELECTED BY ALIGNMENT PROGRAMS OR DSKY INPUT
#
# CALLING -- CALL R52
#
# INPUT -- BESTI AND BESTJ (STAR CODES TIMES 6)
#
# OUTPUT -- STAR CODE IN BITS 1-6, DETENT CODE IN BITS 7-9
# (NO CHECK IS MADE TO INSURE THE DETENT CODE TO BE VALID)
# POINTVSM-1/2 UNIT NAV STAR VEC IN SM
# SCAXIS-AOT OPTIC AXIS VEC IN NB X-Z PLANE
#
# SUBROUT -- R60LEM

                COUNT*  $$/R52
R52             STQ     EXIT
                        SAVQR52
                INDEX   STARIND
                CA      BESTI           # PICK UP STARCODE DETERMINED BY R56
                EXTEND
                MP      1/6TH
                AD      BIT8            # SET DETENT POSITION 2
                TS      STARCODE        # SCALE AND STORE IN STARCODE

R52A            CAF     V01N70
                TC      BANKCALL
                CADR    GOFLASH         # DISPLAY STARCODE AND WAIT FOR RESPONSE
                TC      GOTOPOOH        # V34 -- TERMINATE
                TCF     R52B            # V33 -- PROCEED TO ORIENT LEM
                TCF     R52A            # ENTER -- SELECT NEW STARCODE -- RECYCLE

R52B            TC      DOWNFLAG
                ADRES   3AXISFLG        # BIT6 OF FLAGWRD5 ZERO TO ALLOW VECPOINT
                CA      STARCODE        # GRAB DETENT CODE
                MASK    HIGH9
                EXTEND
                MP      BIT9
                TS      L               # TEMP STORE DETENT

                EXTEND
                BZMF    GETAZEL         # CODE 0, COAS CALIBRATION

                AD      NEG7
                EXTEND
                BZF     GETAZEL         # CODE 7, COAS SIGHTING

                EBANK=  XYMARK
                CA      EBANK7
                TS      EBANK
```

```
                INDEX   L
                CA      AOTAZ -1        # PICK UP AZ CORRESPONDING TO DETENT
                TS      L
                EBANK=  XSM
                CA      EBANK5          # CHANGE TO EBANK5 BUT DON'T DISTURB L
                TS      EBANK
                CA      BIT13           # SET ELV TO 45 DEG
                XCH     L               # SET C(A)=AZ, C(L)=45 DEG
                TCF     AZEL            # GO COMP OPTIC AXIS

GETAZEL         CAF     V06N87          # CODE 0 OR 7, GET AZ AND EL KEY IN
                TC      BANKCALL
                CADR    GOFLASH
                TC      GOTOPOOH        # V34 -- TERMINATE
                TCF     +2              # PROCEED -- CALC OPTIC AXIS
                TCF     GETAZEL         # ENTER -- RECYCLE

                EXTEND
                DCA     AZ              # PICK UP AZ AND EL IN SP 2'S COMP
AZEL            INDEX   FIXLOC          # JAM AZ AND EL IN 8 AND 9 OF VAC
                DXCH    8D
                TC      INTPRET
                CALL                    # GO COMPUTE OPTIC AXIS AND STORE IN
                        OANB            # SCAXIS IN NB COORDS
                RTB     CALL
                        LOADTIME
                        PLANET
                MXV     UNIT
                        REFSMMAT
                STORE   POINTVSM        # STORE FOR VECPOINT

                EXIT
                TC      BANKCALL
                CADR    R60LEM          # GO TORQUE LEM OPTIC AXIS TO STAR LOS

                CAF     HIGH9           # IF COAS CALIBRATION CODE 0.  RECYCLE
                MASK    STARCODE
                EXTEND
                BZF     R52A

                TC      INTPRET         # RETURN FROM KALCMANU
                GOTO
                        SAVQR52         # RETURN TO CALLER

1/6TH           DEC     .1666667
V01N70          VN      0170
V06N87          VN      687
```

```
# LUNAR SURFACE STAR ACQUISITION

                BANK    15
                SETLOC  P50S
                BANK
                COUNT*  $$/R59

R59             CS      FLAGWRD3
                MASK    REFSMBIT        # IF REFSMMAT FLAG CLEAR BYPASS STAR ACQUIRE
                CCS     A
                TCF     R59OUT          # NO REFSMMAT GO TO AOTMARK

                CAF     V01N70*         # SELECT STAR CODE FOR ACQUISITION
                TC      BANKCALL
                CADR    GOFLASH
                TC      GOTOPOOH        # V34 -- TERMINATE
                TCF     R59A            # V33 -- PROCEED
                TCF     R59             # V32 -- RECYCLE

R59A            CS      HIGH9           # GRAB STARCODE FOR INDEX
                MASK    AOTCODE
                EXTEND
                MP      REVCNT          # JUST 6
                XCH     L
                INDEX   STARIND
                TS      BESTI
                INDEX   FIXLOC
                TS      X1              # CODE X 6 FOR CATLOG STAR INDEX
                EXTEND
                BZF     R59OUT          # BYPASS ACQUISITION IF NOT CATLOG STAR
                COM
                AD      DEC227
                EXTEND
                BZMF    R59OUT

                TC      INTPRET
                VLOAD*  MXV
                        CATLOG,1        # GRAB STAR VECTOR
                        REFSMMAT        # TRANSFORM TO SM
                UNIT    CALL
                        CDU*SMNB
                STORE   STAR            # TEMP STORE STAR VEC(NB)
                EXIT

                CAF     BIT1            # INITIALIZE AZ POSITION COD TO 1 (-60)
                TS      POSCODE

                EBANK=  XYMARK
INCAZ           CA      EBANK7
                TS      EBANK
```

```
                INDEX   POSCODE
                CA      AOTAZ -1        # PICK UP AZ CORRESPONDING TO POSCODE
                TS      L

                EBANK=  XSM
                CA      EBANK5
                TS      EBANK

                CA      BIT13           # SET ELV TO 45 DEG
                XCH     L               # SET C(A)=AZ, C(L)=45 DEG
                TS      QMIN            # STORE QMIN=AZ FOR LATER
                INDEX   FIXLOC
                DXCH    8D              # JAM AZ IN 8D, 45 DEG IN 9D FOR OANB

                TC      INTPRET
                CALL
                        OANB            # GO CALC OPTIC AXIS WRT NB
                VLOAD   DOT
                        STAR            # DOT STAR WITH OA
                        SCAXIS
                SL1     ARCCOS
                STORE   24D             # TEMP STORE ARCCOS(STAR.OPTAXIS)

                DSU     BPL
                        DEG30           # SEE IF STAR IN AOT FIELD-OF-VIEW
                        NXAX            # NOT IN FIELD -- TRY NEXT POSITION
                DLOAD   DSU             # SEE IF STAR AT FIELD CENTER
                        24D
                        DEG.5
                BMN     DLOAD           # CALC SPIRAL AND CURSOR
                        ZSPCR           # GO ZERO CURSOR AND SPIRAL
                        24D             # GET SPIRAL
                DMP     SL4
                        3/4             # 12 SCALED AT 16
                STOVL   24D             # 12(ARCCOS(AO.STAR)) SCALED IN REVS

                        SCAXIS          # OA
                VXV     UNIT
                        XUNIT
                PUSH    VXV             # OA X UNITX    PD 0-5
                        SCAXIS
                VCOMP
                UNIT    PDVL            # UNIT(OA X (OA X UNITX))       PD 6-11
                        SCAXIS
                VXV     UNIT
                        STAR
                PUSH    DOT             # 1/2(OA X STAR)        PD 12-17
                        0               # DOT WITH 1/2(OA X UNITX) FOR YROT
                SL1     ARCCOS
                STOVL   26D             # STORE THET SCALED IN REVS
```

```
                DOT                     # UP 12-17, UP 6-11 FOR C2
                BPL     DLOAD           # IF THET NEG -- GET 360-THET
                        R59D
                        ABOUTONE
                DSU
                        26D
                STORE   26D             # 360-THET SCALED IN REVS
R59D            SLOAD   SR1
                        QMIN            # RESCALE AZ(N) TO REVS
                DAD     PUSH            # PUSH YROT + AZ(N) REVS
                        26D
                RTB
                        1STO2S
                STODL   CURSOR          # YROT IN 1/2 REVS
                        24D             # LOAD SROT IN REVS
                DAD                     # 12(SEP) + YROT
                RTB
                        1STO2S
                STORE   SPIRAL          # SROT IN 1/2 REVS
                EXIT
                TCF     79DISP          # GO DISPLAY CURSOR-SPIRAL-POS CODE

ZSPCR           EXIT
                CAF     ZERO            # STAR ALMOST OPTIC AXIS, ZERO CURSOR
                TS      CURSOR          # AND SPIRAL ANGLES
                TS      SPIRAL
                TCF     79DISP

NXAX            EXIT
                INCR    POSCODE
                CS      POSCODE
                AD      SEVEN
                EXTEND
                BZMF    R59ALM          # THIS STAR NOT AT ANY POSITION
                TCF     INCAZ

R59ALM          TC      ALARM           # THIS STAR CAN'T BE LOCATED IN AOT FIELD
                OCT     404
                CAF     VB05N09         # DISPLAY ALARM
                TC      BANKCALL
                CADR    GOFLASH
                TCF     GOTOPOOH        # VB34 -- TERMINATE
                TCF     R59OUT          # VB33 -- PROCEED, GO WITHOUT ACQUIRE
                TCF     R59             # VB32 -- RECYCLE AND TRY ANOTHER STAR

79DISP          CAF     V06N79          # DISPLAY CURSOR, SPIRAL AND POS CODE
                TC      BANKCALL
                CADR    GOFLASH
                TCF     GOTOPOOH        # V34 -- TERMINATE
```

```
                TCF     R59E            # V33 -- PROCEED TO MARK ROUTINE
                TCF     R59             # V32 -- RECYCLE TO TOP OF R59 AGAIN

R59E            CAF     SEVEN           # GET DETENT CODE CORRESPONDING TO POSCODE
                MASK    POSCODE
                EXTEND
                MP      BIT7            # DETEND CODE NOW IN L
                CS      HIGH9
                MASK    AOTCODE         # ISOLATE STAR NO BIT 1-6
                AD      L
                TS      AOTCODE         # STORE DETENT 7-9

R59OUT          TC      BANKCALL        # GO TO AOTMARK FOR SIGHTING
                CADR    AOTMARK
                TC      BANKCALL
                CADR    AOTSTALL        # SLEEP TILL SIGHTING DONE
                TC      CURTAINS        # BADEND RETURN FROM AOTMARK
                TCF     R59RET          # RETURN TO 1 STAR OR 2STAR

V01N70*         VN      170
V06N79          VN      679
DEG30           2DEC    .083333333      # 30 DEGREES
DEG.5           2DEC    .00138888       # .5 DEGREES SCALED IN REVS.
DEG60           OCT     12525           # 60 DEG CDU SCALING
CURSOR          EQUALS  GDT/2
SPIRAL          EQUALS  GDT/2 +2
POSCODE         EQUALS  GDT/2 +4
```

```
# NAME --          PLANET
# FUNCTION --      TO PROVIDE THE REFERENCE VECTOR FOR THE SIGHTED CELESTIAL
#                  BODY.  STARS ARE FETCHED FROM THE CATALOG, SUN, EARTH AND
#                  MOON ARE COMPUTED BY LOCSAM, PLANET VECTORS ARE ENTERED
#                  BY DSKY INPUT.
# CALL --          CALL
#                          PLANET
# INPUT --         TIME IN MPAC
# OUTPUT --        VECTOR IN MPAC
# SUBROUTINES -- LOCSAM
# DEBRIS --        VAC, STARAD - STARAD +17

                   SETLOC  P50S
                   BANK
                   COUNT*  $$/P51

PLANET             STOVL   TSIGHT
                           ZEROVEC
                   STORE   STARAD
                   STQ     EXIT
                           GCTR
                   CS      HIGH9
                   MASK    AOTCODE
                   EXTEND
                   MP      REVCNT
                   XCH     L
                   INDEX   STARIND
                   TS      BESTI
                   CCS     A
                   TCF     NOTPLAN
                   CAF     VNPLANV
                   TC      BANKCALL
                   CADR    GOFLASH
                   TC      -3
                   TC      +2
                   TC      -5
                   TC      INTPRET
                   VLOAD   UNIT
                           STARAD
                   GOTO
                           GCTR
NOTPLAN            CS      A
                   AD      DEC227
                   EXTEND
                   BZMF    CALSAM1
                   INDEX   STARIND
                   CA      BESTI
                   INDEX   FIXLOC
                   TS      X1
                   TC      INTPRET
```

```
                VLOAD*  GOTO
                        CATLOG,1
                        GCTR
CALSAM1         TC      INTPRET
CALSAM          DLOAD   CALL
                        TSIGHT
                        LOCSAM
                LXC,1   VLOAD
                        STARIND
                        VEARTH
                STOVL   OD
                        VSUN
                STOVL   VEARTH
                        OD
                STORE   VSUN
                DLOAD*  LXC,1
                        BESTI,1
                        MPAC
                VLOAD*  GOTO
                        STARAD -228D,1
                        GCTR
DEC227          DEC     227
VNPLANV         VN      0688
PIPSRINE        =       PIPASR +3       # EBANK NOT 4 SO DON'T LOAD PIPTIME1
```

```
# GRAVITY VECTOR DETERMINATION ROUTINE
# BY KEN VINCENT
#
# FOR DETAILED DESCRIPTION SEE 504GSOP 5.6.3.2.5.
#
# THIS PROGRAM FINDS THE DIRECTION OF THE MOON'S GRAVITY
# WHILE THE LM IS IN THE MOON'S SURFACE.  IT WILL BE USED
# FOR LUNAR SURFACE ALIGNMENT.  THE GRAVITY VECTOR IS
# DETERMINED BY READING THE PIPAS WITH THE IMU AT TWO
# PARTICULAR ORIENTATIONS.  THE TWO READINGS ARE AVERAGED
# AND UNITIZED AND TRANSFORMED TO NB COORDINATES.  THE TWO
# ORIENTATIONS WERE CHOSEN TO REDUCE BIAS ERRORS IN THE
# READINGS.
#
# CALL --
#       TC      BANKCALL
#       CADR    GVDETER
#
# INPUTS --
#       PIPAS, CDUS
#
# OUTPUTS --
#       STARSAV1 = UNIT GRAVITY
#       GSAV     = DITTO
#       GRAVBIT  = 1
#
# SUBROUTINES --
#       PIPASR, IMUCOARS, IMUFINE, IMUSTALL, 1/PIPA, DELAYJOB, CDUTRIG,
#       *NBSM*, *SNMB*, CALCGA, GOFLASH
#
# DEBRIS --
#       VAC, SAC, STARAD, XSM, XNB, THETAD, DELV, COSCDU, SINCDU

GVDETER         CAF     42DEG
                TS      THETAD
                COM
                TS      THETAD  +1
                CAF     35DEG
                TS      THETAD  +2
                TC      INTPRET
                CLEAR   CALL
                        REFSMFLG
                        LUNG

# FIND GIMBAL ANGLES WHICH ROTATE SM 180 DEG ABOUT G VEC
#
#       DEFINE G COOR SYS
#                     _
#                   [ X ]   [    UNIT G    ]
#               *   [ _ ]   [            _ ]
#               M = [ Y ] = [ UNITEZSM * X ]
#                   [ _ ]   [       _   _  ]
#                   [ Z ]   [ UNIT( X * Y )]
#
#       THEN ROTATED SM WRT PRESENT IS
#
```

```
#                       [ 1    0    0 ]
#                *      *T  [              ] *              *       *
#              XSM =    M   [ 0   -1    0 ] M   =   2 (X X ) - 1/2 I
#                           [              ]            I J
#                           [ 0    0   -1 ]
#
#      ALSO NB WRT PRES SM IS
#
#                *      *  *
#              XNB = NBSM I
#
#                                 *    *
#      GIMBAL ANGLES = CALCGA( XSM, XNB )

               SETLOC   P50S
               BANK
               COUNT*   $$/P57
               AXT,1    SSP              # X1=18
                        18D              # S1=6
                        S1               # X2, -2
                        6D
               LXC,2
                        S1
GRAVEL         VLOAD*   CALL
                        XUNIT -6,2
                        *NBSM*           # SIN AND COS COMPUTED IN LUNG
               STORE    XNB +18D,1
               VLOAD
                        STAR
               LXC,2    VXSC*            # COMPLEMENT -- UNITX ARE BACKWARD --
                        X2
                        STAR +6,2        # OUTER PRODUCT
               VSL2     LXC,2
                        X2
               VSU*     INCR,2
                        XUNIT -6,2
                        2D
               STORE    XSM +18D,1
               TIX,1    CALL
                        GRAVEL
                        CALCGA
               VLOAD    VSR1
                        GOUT
               STCALL   STARAD +12D
                        LUNG
               VLOAD    VSR1
                        GOUT
               VAD      UNIT
                        STARAD +12D
               STORE    STARSAV1
               DOT
                        GSAV
               SL1      ACOS
```

```
                STORE   DSPTEM1
                EXIT
                TC      DOWNFLAG        # CLAR FREEFLAG IN CASE OF RECYCLE
                ADRES   FREEFLAG

                CA      DISGRVER
                TC      BANKCALL
                CADR    GOFLASH
                TC      GOTOPOOH
                TCF     PROGRAV         # VB33 -- PROCEED
                TC      UPFLAG          # VB32 -- RECYCLE -- STORE GRAV AND DO IT AGAIN
                ADRES   FREEFLAG        # AND SET FREEFLAG TO SHOW RECYCLE

PROGRAV         TC      PHASCHNG
                OCT     04024

                TC      INTPRET
                VLOAD
                        STARSAV1
                STORE   GSAV
                EXIT
                CAF     FREEFBIT        # IF FREEFLAG SET, RE-COMPUTE GRAVITY
                MASK    FLAGWRD0
                CCS     A
                TCF     GVDETER         # SET
                TCF     ATTCHK          # EXIT FROM GVDETER

LUNG            STQ     VLOAD
                        QMIN
                        ZEROVEC
                STORE   GACC
                EXIT
                TC      PHASCHNG
                OCT     04024

                TC      BANKCALL
                CADR    IMUCOARS
                TC      BANKCALL
                CADR    IMUSTALL
                TC      CURTAINS
                TC      BANKCALL
                CADR    IMUFINE
                TC      BANKCALL
                CADR    IMUSTALL
                TC      CURTAINS
                CA      T/2SEC
                TS      GCTR
                CA      PRIO31
                TS      1/PIPADT
                TC      BANKCALL
```

```
                CADR    GCOMPZER        # INITIALIZE COMPENSATION
                TC      PHASCHNG
                OCT     04024

                TC      BANKCALL        # DON'T NEED TO INHINT.  THIS USED TO
                CADR    PIPSRINE        # INITIALIZE PIPAS.  DON'T USE DATA
                TC      INTPRET
GREED           EXIT                    # = MASK 7776 IN BASIC SO DON'T CARE
                CAF     2SECS
                TC      TWIDDLE         # SET UP 2 SEC TASK TO READ PIPAS
                ADRES   GRABGRAV

                TC      ENDOFJOB

GRABGRAV        TC      IBNKCALL
                CADR    PIPSRINE
                CAF     PRIO13          # RE-ESTABLISH MAINLINE JOB
                TC      FINDVAC
                EBANK=  STARAD
                2CADR   ADDGRAV

                TC      TASKOVER

ADDGRAV         TC      BANKCALL
                CADR    1/PIPA
                INCR    GCTR
                TC      INTPRET
                VLOAD   VAD
                        DELV
                        GACC
                STORE   GACC            # ACCUMULATE G VECTOR
                SLOAD   BMN
                        GCTR
                        GREED
                VLOAD   UNIT
                        GACC
                STCALL  STAR
                        CDUTRIG         # TRANSFORM IN NB COOR AND STORE
                CALL                    #       IN OUTPUT
                        *SMNB*
                STORE   GOUT
                EXIT
                TC      PHASCHNG
                OCT     04024

QMINEXIT        TC      INTPRET
                GOTO
                        QMIN
T/2SEC          DEC     -20
```

DISGRVER	VN	0604
42DEG	OCT	07357
35DEG	OCT	06211

```
# NAME -- GYROTRIM
#
# THIS PROGRAM COMPUTES AND SENDS GYRO COMMANDS WHICH CAUSE THE CDUS
# TO ATTAIN A PRESCRIBED SET OF ANGLES.  THIS ROUTINE ASSUMES THE
# VEHICLES ATTITUDE REMAINS STATIONARY DURING ITS OPERATION.
#
# CALL          CALL
#                       GYROTRIM
#
# INPUT         THETAD,+1,+2 = DESIRED CDU ANGLES
#               CDUX,CDUY,CDUZ
#
# OUTPUT        GYRO TORQUE PULSES
#
# SUBROUTINES   TRG*NBSM, *NBSM*, CDUTRIG, AXISGEN, CALCGTA, IMUFINE
#               IMPULSE, IMUSTALL
#               _______  ______  ______     *          *   ___
# DEBRIS        CDUSPOT, SINCDU, COSCDU, STARAD, VAC, XDC, OGC

                COUNT*  $$/P57
GYROTRIM        STQ     DLOAD
                        QMIN
                        THETAD
                PDDL    PDDL
                        THETAD +2
                        THETAD +1
                VDEF
                STOVL   CDUSPOT
                        XUNIT
                CALL
                        TRG*NBSM
                STOVL   STARAD
                        YUNIT
                CALL
                        *NBSM*
                STCALL  STARAD +6
                        CDUTRIG
                CALL
                        CALCSMSC
                VLOAD
                        XNB
                STOVL   6D
                        YNB
                STCALL  12D
                        AXISGEN
                CALL
                        CALCGTA
JUSTTRIM        EXIT
                TC      BANKCALL
                CADR    IMUFINE
                TC      BANKCALL
```

```
                CADR    IMUSTALL
                TC      CURTAINS
                CA      GYRCDR
                TC      BANKCALL
                CADR    IMUPULSE
                TC      BANKCALL
                CADR    IMUSTALL
                TC      CURTAINS
                TCF     QMINEXIT

GYRCDR          ECADR   OGC
```

```
# PERFORM STAR ACQUISITION AND STAR SIGHTINGS

2STARS          CAF     ZERO            # INITIALIZE STARIND
                TCF     +2              # ZERO FOR 1ST STAR, ONE FOR 2ND STAR
1STAR           CAF     BIT1
                TS      STARIND

                TC      PHASCHNG
                OCT     04024

                TCF     R59             # GO DO STAR ACQUIRE AND AOTMARK

R59RET          CA      STARIND         # BACK FROM SURFACE MARKING
                EXTEND
                BZF     ASTAR           # 1ST STAR MARKED

                TC      PHASCHNG
                OCT     04024

                TC      INTPRET
                DLOAD   CALL
                        TSIGHT          # TIME OF 2ND MARK
                        PLANET
                STCALL  VEC2            # STORE 2ND CATALOG VEC (REF)
                        SURFLINE

ASTAR           TC      INTPRET
                VLOAD
                        STARAD +6
                STORE   STARSAV1        # 1ST OBSERVED STAR (SM)
                DLOAD   CALL
                        TSIGHT          # TIME OF 1ST MARK
                        PLANET
                STORE   VEC1            # STORE 1ST CATALOG VEC (REF)
                EXIT
                TCF     1STAR           # GO GET 2ND STAR SIGHTING
```

```
# DO FINE OR COARSE ALIGNMENT OF IMU

SURFLINE        SSP     AXT,2
                        S2
                        6
                        12D
WRTDESIR        VLOAD*  MXV
                        VEC1 +12D,2     # PICK UP VEC IN REF, TRANS TO DESIRED SH
                        XSMD
                UNIT
                STORE   STARAD +12D,2   # VEC IN SM
                VLOAD*
                        STARSAV1 +12D,2 # PICK UP VEC IN PRESENT SM
                STORE   18D,2
                TIX,2   BON
                        WRTDESIR
                        INITALGN        # IF INITIAL PASS (OPTION 0) BYPASS R54
                        INITBY
DOALIGN         CALL
                        R54             # DO CHKSDATA
                BOFF
                        FREEFLAG
                        P57POST         # ASTRO DOES NOT LIKE DATA TEST RESULTS
INITBY          CALL
                        AXISGEN         # GET DESIRED ORIENT WRT PRES. XDC,YDC,ZDC
                CALL
                        CALCGTA         # GET GYRO TORQ ANGLES, OGC,IGC,MGC
                EXIT
                CAF     INITABIT        # IF INITIAL PASS BYPASS NOUN 93 DISPLAY
                MASK    FLAGWRD8
                CCS     A
                TCF     5DEGTEST
                CAF     DISPGYRO        # DISPLAY GYRO TORQ ANGLES V 06N93
                TC      BANKCALL
                CADR    GOFLASH
                TC      GOTOPOOH        # V34 -- TERMINATE
                TCF     5DEGTEST        # VB33 -- PROCEED TO COARSE OR FINE
                TCF     P57POST +1      # VB32 -- RECYCLE, MAYBE RE-ALIGN

5DEGTEST        TC      INTPRET         # IF ANGLES GREATER THAN 5 DEGS, DO COARSE
                VLOAD   BOV
                        OGC
                        SURFSUP
SURFSUP         STORE   OGCT
                V/SC    BOV
                        5DEGREES
                        COATRIM
                SSP     GOTO
                        QMIN
                        SURFDISP
```

```
                        JUSTTRIM        # ANGLES LESS THAN 5 DEG, DO GYRO TORQ

SURFDISP        EXIT
                TC      PHASCHNG
                OCT     04024

                TC      INTPRET
                AXC,1   AXC,2
                        XSMD
                        REFSMMAT
                SET     CALL
                        REFSMFLG
                        MATMOVE
                EXIT
                CCS     OPTION2         # IF OPTION ZERO DO FINISH
                TCF     B2F8
                TCF     P57POST +1

B2F8            CAF     INITABIT        # IF INITIAL FLAG SET, RE-CYCLE.
                MASK    FLAGWRD8
                CCS     A
                TCF     P57JUMP         # IT'S SET
                TC      INTPRET
                CALL
                        REFMF           # GO GET ATTITUDE VEC IN MF(YNBSAV,XNBSAV)
P57POST         EXIT
                CAF     OCT14           # DISPLAY V50N25 CHK CODE 14
                TC      BANKCALL
                CADR    GOPERF1
                TCF     GOTOPOOH        # VB34 -- TERMINATE
                TCF     P57JUMP         # VB33 -- PROCEED TO RE-ALIGN
                CS      BIT2            # TEST TO SEE IF ALIGNED BY OPTION 2
                AD      OPTION2
                EXTEND
                BZF     +2              # YES -- GO CALCULATE LANDING SITE
                TCF     GOTOPOOH        # NO -- EXIT P57
                TC      PHASCHNG        # RESTART PLACE
                OCT     04024
                TC      INTPRET
                VLOAD   CALL            # USE GNB
                        GSAV
                        CDU*NBSM        # GO TO SM COORDS
                VXM     SET             #       ON MOON SO SET LUNAFLAG
                        REFSMMAT        #       G(REF) = (REFSMMAT)T (NBSM)GNB
                        LUNAFLAG
                PDVL    ABVAL
                        RLS
                VXSC    STADR
                STORE   ALPHAV          # ALPHAV = RLSMAG * G(REF)
                CLEAR   RTB
```

```
                ERADFLAG
                LOADTIME
        CALL
                N89DISP         # SUBROUTINE TO CALC LS AND GIVE RLS BACK
        STORE   RN              # RN=RLS B-29 = LM POSITION
        VSL2    PDDL            # R-TO-RP GETS RLS B-27 AT 0-50 IN PDLIST
                GDT/2 +4        # TIME TEMP STORED IN N89DISP
        PUSH                    # TIME AT 6-7 IN PDLIST
        STCALL  PIPTIME         # PIPTIME = LM STATE TIME
                R-TO-RP
        STORE   RLS             # RLS IN MOON-FIXED COORDS
        EXIT
        TCF     GOTOPOOH        # EXIT P57
```

```
# COARSE AND FINE ALIGN IMU

COATRIM         AXC,1   AXC,2
                        XDC
                        XSM
                CALL
                        MATMOVE
                CALL
                        CDUTRIG
                CALL
                        CALCSMSC
                CALL
                        CALCGA
                BOFF    EXIT
                        INITALGN        # IF INITIAL ALIGNMENT DISPLAY FINAL
                        CORSIT          # GIMBAL ANGLES IF COARSE ANGLES GREATER
                CAF     V06N22          # THAN 5 DEGREES
                TC      BANKCALL
                CADR    GOFLASH
                TC      GOTOPOOH
                TCF     +2
                TCF     -5
                TC      PHASCHNG
                OCT     04024

                TC      INTPRET
CORSIT          CALL
                        COARSE
                CALL
                        NCOARSE
                CALL
                        GYROTRIM
                GOTO
                        SURFDISP
DISPGYRO        VN      0693
```

```
# LUNAR SURFACE IMU ALIGNMENT PROGRAM

P57             TC      BANKCALL        # IS ISS ON -- IF NOT, IMUCHK WILL SEND
                CADR    IMUCHK          # ALARM CODE 210 AND EXIT VIA GOTOPOOH

                CAF     THREE           # JAM REFSMMAT OPTION 3 FOR INITIAL DISP.
                TS      OPTION2
P57OPT          CAF     BIT1
                TC      BANKCALL
                CADR    GOPERF4R        # FLASH V04N06 FOR ALIGNMENT CODE
                TC      GOTOPOOH        # V34 TERMINATE
                TCF     ALIGNOPT        # V33 PROCEED
                TCF     P57OPT          # V32 RECYCLE

                TC      PHASCHNG
                OCT     00014
                TC      ENDOFJOB

ALIGNOPT        CA      OPTION2
                MASK    THREE
                INDEX   A
                TCF     +1
                TCF     TDISP           # OPTION 4 LS ORIENTATION
                TCF     PACKOPTN        # OPTION 1 PREFERRED
                TCF     P57OPT          # OPTION 2 INVALID IN P57, RECYCLE
                TC      INTPRET         # OPTION 3 REFSMMAT
                AXC,1   AXC,2           # JAM REFSMMAT IN XSMD LOC
                        REFSMMAT
                        XSMD
                CALL
                        MATMOVE
                GOTO
                        PACKOPTN -1

TDISP           TC      INTPRET
                DLOAD
                        TIG             # LOAD ASCENT TIME FOR DISPLAY
P57A            STORE   DSPTEM1
                EXIT
P57AA           CAF     V06N34*         # DISPLAY TALIGN, TALIGN : DSPTEM1
                TC      BANKCALL
                CADR    GOFLASH
                TCF     GOTOPOOH        # V34 -- TERMINATE
                TCF     +2
                TCF     P57AA           # VB32 -- RECYCLE

                TC      INTPRET
                RTB     PDDL
                        LOADTIME        # PUSH CURRENT TIME AND PICK UP KEY IN
                        DSPTEM1
```

```
                BZE      PDDL
                         P57C           # IF KEY IN TIME ZERO - TALIGN=CURRENT TIME
                DSU      BPL            # NOT ZERO SO EXCHANGE PD WITH DSPTEM1
                         DSPTEM1
                         P57C
                DLOAD    STADR          # IF KEYIN TIME GREATER THAN CURRENT TIME
                STORE    TIG            # STORE IT IN TIG
                STCALL   TALIGN
                         P57D
P57C            DLOAD    STADR
                STORE    TALIGN
P57D            STCALL   TDEC1
                         LEMPREC        # COMPUTE DESIRED IMU ORIENTATION STORE
                VLOAD    UNIT           # IN X,Y,ZSMD
                         RATT
                STCALL   XSMD
                         LSORIENT
                EXIT
PACKOPTN        CAF      ZERO           # PACK FLAG BITS FOR OPTION DISPLAY
                TS       OPTION1 +1     # JAM ZERO IN ALIGNMENT OPTION
                TS       OPTION1 +2     # INITIALIZE FLAG BIT CONFIGURATION
                CAF      REFSMBIT
                MASK     FLAGWRD3       # REFSMFLG
                CCS      A
                CAF      BIT7           # SET
                ADS      OPTION1 +2     # CLEAR -- JUST ZERO
                CAF      ATTFLBIT
                MASK     FLAGWRD6       # ATTFLG
                CCS      A
                CAF      BIT4           # SET
                ADS      OPTION1 +2     # CLEAR -- ZERO IN A
                CAF      BIT4
                TS       OPTION1        # JAM 00010 IN OPTION1 FOR CHECK LIST

DSPOPTN         CAF      VB05N06        # DISPLAY OPTION CODE AND FLAG BITS
                TC       BANKCALL
                CADR     GOFLASH
                TCF      GOTOPOOH       # VB34 -- TERMINATE
                TCF      +2             # V33 -- PROCEED
                TCF      DSPOPTN        # V32 -- RECYCLE

                CAF      REFSMBIT
                MASK     FLAGWRD3
                CCS      A
                TCF      GETLMATT       # SET, GO COMPUTE LM ATTITUDE
                CAF      ATTFLBIT       # CLEAR -- CHECK ATTFLAG FOR STORED ATTITUDE.
                MASK     FLAGWRD6
                CCS      A
                TCF      BYLMATT        # ALLFLG SET, CHK OPTION FOR GRAVITY COMP
                CAF      BIT2           # SEE IF OPTION 2 OR 3
```

```
          MASK    OPTION2
          CCS     A
          TCF     BYLMATT          # OPTION 2 OR 3 BUT DON'T HAVE ATTITUDE
          TC      ALARM            # OPTION INCONSISTENT WITH FLAGS -- ALARM 701
          OCT     701
          CAF     VB05N09          # DISPLAY ALARM FOR ACTION
          TC      BANKCALL
          CADR    GOFLASH
          TCF     GOTOPOOH         # VB34 -- TERMINATE
          TCF     DSPOPTN          # V33 -- PROCEED *********TEMPORARY
          TCF     DSPOPTN          # VB32 -- RECYCLE TO OPTION DISPLAY V 05N06
```

```
# TRANSFORM VEC1,2 FROM MOON FIXED TO REF AND JAM BACK IN VEC1,2

MFREF           STQ     SETPD
                        QMAJ
                        0
                RTB
                        LOADTIME
                STOVL   TSIGHT
                        VEC1
                PDDL    PUSH
                        TSIGHT
                CALL
                        RP-TO-R
                STOVL   VEC1
                        VEC2
                SETPD   PDDL
                        0
                        TSIGHT
                PUSH    CALL
                        RP-TO-R
                STCALL  VEC2
                        QMAJ
```

```
# COMPUTE LM ATTITUDE IN MOON FIXED COORDINATES USING REFSMMAT AND
# STORE IN YNBSAV AND ZNBSAV.

REFMF           STQ     CALL
                        QMAJ
                        CDUTRIG         # GET SIN AND COS OF CDUS
                RTB     SETPD
                        LOADTIME
                        0
                STCALL  TSIGHT
                        CALCSMSC        # GET YNB IN SM
                VLOAD   VXM
                        YNB
                        REFSMMAT        # YNB TO REF
                UNIT    PDDL
                        TSIGHT
                PUSH    CALL
                        R-TO-RP
                STOVL   YNBSAV          # YNB TO MF
                        ZNB
                VXM     UNIT
                        REFSMMAT        # ZNB TO REF
                PDDL    PUSH
                        TSIGHT
                CALL
                        R-TO-RP         # ZNB TO MF
                STORE   ZNBSAV
                SETGO
                        ATTFLAG
                        QMAJ
```

```
# BRANCH TO ALIGNMENT OPTION

GETLMATT        TC      INTPRET
                CALL
                        REFMF           # GO TRANSFORM TO MF IN YNBSAV, ZNBSAV
                EXIT

BYLMATT         TC      UPFLAG          # SET INITIAL ALIGN FLAG
                ADRES   INITALGN
                CAF     BIT1
                MASK    OPTION2         # SEE IF OPTION 1 OR 3
                CCS     A
                TCF     GVDETER         # OPTION 1 OR 2, GET GRAVITY

ATTCHK          TC      PHASCHNG
                OCT     04024

                CAF     ATTFLBIT        # NOT 1 OR 3, CHECK ATTFLAG
                MASK    FLAGWRD6
                CCS     A
                TCF     P57OPTO         # GET ALIGNMENT VECS FOR OPTION 0
P57JUMP         TC      PHASCHNG
                OCT     04024

                TC      DOWNFLAG        # ATTFLG CLEAR -- RESET INTALIGN FLAG
                ADRES   INITALGN
                CAF     THREE
                MASK    OPTION2         # BRANCH ON OPTION CODE
                INDEX   A
                TCF     +1
                TCF     P57OPT0         # OPTION IS 0
                TCF     P57OPT1         # OPTION IS 1
                TCF     P57OPT2         # OPTION IS 2
                TCF     P57OPT3         # OPTION IS 3
```

```
# OPTION 0, GET TWO ATTITUDE VECS

P57OPT0         TC      INTPRET
                VLOAD
                        YNBSAV          # Y AND Z ATTITUDE WILL BE PUT IN REF
                STOVL   VEC1
                        ZNBSAV
                STCALL  VEC2
                        CDUTRIG
                CALL
                        CALCSMSC        # COMPUTE SC AXIS WRT PRESENT SM
                VLOAD
                        YNB
SAMETYP         STOVL   STARSAV1        # Y SC AXIS WRT PRESENT SM
                        ZNB
                STCALL  STARSAV2        # Z SC AXIS WRT PRESENT SM
                        MFREF           # TRANSFORM VEC1,2 FROM MF TO REF
                GOTO
                        SURFLINE

# OPTION 1, GET LANDING SITE AND Z-ATTITUDE VEC

P57OPT1         TC      INTPRET
                VLOAD   UNIT
                        RLS             # LANDING SITE VEC
                STOVL   VEC1
                        ZNBSAV          # Z ATTITUDE VEC
                STCALL  VEC2
                        CDUTRIG
                CALL
                        CALCSMSC        # GET ZNB AXIS WRT PRES SM FOR STARSAV2
                VLOAD   CALL
                        GSAV            # TRANS GSAV FROM NB TO SM FOR STARSAV1
                        CDU*NBSM
                GOTO
                        SAMETYP         # NOW DO SAME AS OPTION 0
```

```
# OPTION 2, GET TWO STAR SIGHTINGS

P57OPT2         TCF     2STARS          # DO SIGHTING ON 2 STARS

# OPTION 3, GET LANDING SITE VEC AND ONE STAR SIGHTING

P57OPT3         TC      INTPRET
                VLOAD   UNIT
                        RLS             # LANDING SITE VEC
                STORE   VEC1
                STOVL   VEC2            # DUMMY VEC2 FOR 2ND CATALOG STAR
                        GSAV            # GRAVITY VEC NB
                CALL
                        CDU*NBSM        # TRANS GSAV FROM NB TO SM FOR STARSAV1
                STCALL  STARSAV1
                        MFREF           # STARSAV2 IS STORED AS 2ND OBSERVED STAR
                EXIT
                TCF     1STAR           # 1STAR GET VEC2, STARSAV2 GOES TO SURFLINE.

VB05N06         VN      506
```

```
# CHECK IMODES30 TO VERIFY IMU IS ON

IMUCHK          CS      IMODES30
                MASK    BIT9
                CCS     A               # IS IMU ON
                TCF     +4              # YES

                TC      ALARM           # NO, SEND ALARM AND EXIT
                OCT     210
                TC      GOTOPOOH

                TC      UPFLAG
                ADRES   IMUSE           # SET IMUSE FLAG

                TC      SWRETURN

                BANK    04
                SETLOC  AOTMARK2
                BANK
                COUNT*  $$/P57

LSORIENT        STQ     VLOAD
                        QMAJ
                        RRECTCSM
                VXV     VXV
                        VRECTCSM
                        XSMD
                UNIT
                STORE   ZSMD
                VXV     UNIT
                        XSMD
                STCALL  YSMD
                        QMAJ
```

```
# NAME - LSPOS  - LOCATE SUN AND MOON                         DATE - 25 OCT 67
# MOD NO.1
# MOD BY NEVILLE                                              ASSEMBLY SUNDANCE
#
# FUNCTIONAL DESCRIPTION
#
#        COMPUTES UNIT POSITION VECTOR OF THE SUN AND MOON IN THE BASIC REFERENCE SYSTEM. THE SUN VECTOR S IS
# LOCATED VIA TWO ANGLES. THE FIRST ANGLE(OBLIQUITY) IS THE ANGLE BETWEEN THE EARTH EQUATOR AND THE ECLIPTIC. THE
# SECOND ANGLE IS THE LONGITUDE OF THE SUN MEASURED IN THE ECLIPTIC.
# THE POSITION VECTOR OF THE SUN IS
#        -
#        S=(COS(LOS), COS(OBL)*SIN(LOS), SIN(OBL)*SIN(LOS)), WHERE
#
#      LOS=LOS +LOS *T-(C *SIN(2PI*T)/365.24 +C *COS(2PI*T)/365.24)
#             O    R     O                    1
#      LOS  (RAD) IS THE LONGITUDE OF THE SUN FOR MIGNIGHT JUNE 30TH OF THE PARTICULAR YEAR.
#         O
#      LOS  (RAD/DAY) IS THE MEAN RATE FOR THE PARTICULAR YEAR.
#         R
# LOS  AND LOS  ARE STORED AS LOSO AND LOSR IN RATESP.
#    O        R
# COS(OBL) AND SIN(OBL) ARE STORED IN THE MATRIX KONMAT.
# T, TIME MEASURED IN DAYS(24 HOURS), IS STORED IN TIMEP.
# C  AND C  ARE FUDGE FACTORS TO MINIMIZE THE DEVIATION. THEY ARE STORED AS ONE CONSTANT(CMOD), SINCE
#  O      1                         2  2 1/2
# C *SIN(X)+C *COS(X) CAN BE WRITTEN AS (C +C )   *SIN(X+PHI), WHERE PHI=ARCTAN(C /C ).
#  O         1                          O  1                                   1  O
#
# THE MOON IS LOCATED VIA FOUR ANGLES. THE FIRST IS THE OBLIQUITY. THE SECOND IS THE MEAN LONGITUDE OF THE MOON,
# MEASURED IN THE ECLIPTIC FROM THE MEAN EQUINOX TO THE MEAN ASCENDING NODE OF THE LUNAR ORBIT, AND THEN ALONG THE
# ORBIT. THE THIRD ANGLE IS THE ANGLE BETWEEN THE ECLIPTIC AND THE LUNAR ORBIT. THE FOURTH ANGLE IS THE LONGITUDE
# OF THE NODE OF THE MOON, MEASURED IN THE LUNAR ORBIT. LET THESE ANGLES BE OBL,LOM,IM, AND LON RESPECTIVELY.
#
# THE SIMPLIFIED POSITION VECTOR OF THE MOON IS
#   -
#   M=(COS(LOM), COS(OBL)*SIN(LOM)-SIN(OBL)*SIN(IM)*SIN(LOM-LON), SIN(OBL)*SIN(LOM)+COS(OBL)*SIN(IM)*SIN(LOM-LON))
#
#   WHERE
#      LOM=LOM +LOM *T-(A *SIN(2PI*T/27.5545)+A *COS(2PI*T/27.5545)+B *SIN(2PI*T/32)+B *COS(2PI*T/32)), AND
#             O    R     O                    1                      O                 1
#      LON=LON +LON
#             O    R
# A , A , B  AND B  ARE STORED AS AMOD AND BMOD (SEE DESCRIPTION OF CMOD, ABOVE).  COS(OBL), SIN(OBL)*SIN(IM),
#  O   1   O      1
# SIN(OBL), AND COS(OBL)*SIN(IM) ARE STORED IN KONMAT AS K1, K2, K3 AND K4, RESPECTIVELY. LOM , LOM , LON , LON
# ARE STORED AS LOMO, LOMR, LONO, AND LONR IN RATESP.                                        O     R     O     R
# THE THREE PHIS ARE STORED AS AARG, BARG, AND CARG(SUN).  ALL CONSTANTS ARE UPDATED BY YEAR.
#
# CALLING SEQUENCE
```

```
#   CALL LSPOS.  RETURN IS VIA QPRET.
# ALARMS OR ABORTS
#   NONE
# ERASABLE INITIALIZATION REQUIRED
#   TEPHEM - TIME FROM MIGNIGHT 1 JULY PRECEDING THE LAUNCH TO THE TIME OF THE LAUNCH (WHEN THE AGC CLOCK WENT
# TO ZERO). TEPHEM IS TP WITH UNITS OF CENTI-SECONDS.
#   TIME2 AND TIME1 ARE IN MPAC AND MPAC +1 WHEN PROGRAM IS CALLED.
# OUTPUT
#   UNIT POSITIONAL VECTOR OF SUN IN VSUN.   (SCALED B-1)
#   UNIT POSITIONAL VECTOR OF MOON IN VMOON. (SCALED B-1)
# SUBROUTINES USED
#   NONE
# DEBRIS
#   CURRENT CORE SET,WORK AREA AND FREEFLAG
                BANK    04
                SETLOC  EPHEM
                BANK
                EBANK=  VSUN
                COUNT*  $$/EPHEM
LUNPOS          EQUALS  LSPOS
LSPOS           SETPD   SR
                        0
                        14D             # TP
                TAD     DDV
                        TEPHEM          # TIME OF LAUNCH [IN CENTISEC B 42]
                        CSTODAY         # 24 HOURS-8640000 CENTI-SECS/DAY B-33
                STORE   TIMEP           # T IN DAYS [@ B 9 = 512 DAYS]
                AXT,1   AXT,2           #           [□ GRANULARITY ≈ 0.164 SEC]
                        0
                        0
                CLEAR
                        FREEFLAG        # SWITCH BIT
POSITA          DLOAD
                        KONMAT +2       # ZERO$
                STORE   GTMP
POSITB          DLOAD   DMP*
                        TIMEP           # T
                        VAL67 +4,1      # 1/27 OR 1/32 OR 1/365
```

```
                        SL          DAD*
                                    8D
                                    VAL67 +2,1          # AARG
                        SIN         DMP*                # SIN(T/27+PHI) OR T/32 OR T/365
                                    VAL67,1             # (A0**2+A1**2)**1/2SIN(X+PHIA)
                        DAD         INCR,1              # PLUS
                                    GTMP                # (B0**2+B1**2)**1/2SIN(X+PHIB)
                        DEC         -6
                        STORE       GTMP                # OR (C0**2+C1**2)**1/2SIN(X+PHIC)
                        BOFSET
                                    FREEFLAG
                                    POSITB
        POSITD          DLOAD       DMP*
                                    TIMEP               # T
                                    RATESP,2            # LOMR,LOSR,LONR
                        SL          DAD*
                                    5D
                                    RATESP +6,2         # LOMO,LOSO,LONO
                        DSU
                                    GTMP
                        STORE       STMP,2              # LOM,LOS,LON
                        SLOAD       INCR,2
                                    X2
                        DEC         -2
                        DAD         BZE
                                    RCB-13              # PLUS 2
                                    POSITE              # 2ND
                        BPL
                                    POSITA              # 1ST
        POSITF          DLOAD       DSU                 # 3RD
                                    STMP                # LOM
                                    STMP +4             # LON
                        SIN         PDDL                # SIN(LOM-LON)
                                    STMP
                        SIN         PDDL                # SIN LOM
                                    STMP
                        COS         VDEF                # COS LOM
                        MXV         UNIT
                                    KONMAT              # K1,K2,K3,K4,
                        STORE       VMOON
                        DLOAD       PDDL
                                    KONMAT +2           # ZERO
                                    STMP +2
                        SIN         PDDL                # SIN LOS
                                    STMP +2
                        COS         VDEF                # COS LOS
                        MXV         UNIT
                                    KONMAT
                        STORE       VSUN
                        RVQ
```

```
POSITE          DLOAD
                        KONMAT +2       # ZEROS
                STORE   GTMP
                GOTO
                        POSITD
LUNVEL          RVQ                     #          TO FOOL INTEGRATION
                SETLOC  EPHEM1
                BANK

                COUNT*  $$/EPHEM
STMP            EQUALS  16D
GTMP            EQUALS  22D
TIMEP           EQUALS  24D

# *** END OF LEMP50S .115 ***
```

```
# PROGRAM NAME -- DOWN TELEMETRY PROGRAM
# MOD NO. -- 0          TO COMPLETELY REWRITE THE DOWN TELEMETRY PROGRAM AND DOWNLINK ERASABLE DUMP PROGRAM FOR THE
#                       PURPOSE OF SAVING APPROXIMATELY 150 WORDS OF CORE STORAGE.
#                       THIS CHANGE REQUIRES AN ENTIRELY NEW METHOD OF SPECIFYING DOWNLINK LISTS.  REFER TO DOWNLINK
#                       LISTS LOG SECTION FOR MORE DETAILS.  HOWEVER THIS CHANGE WILL NOT AFFECT THE GROUND PROCESSING
#                       OF DOWN TELEMETRY DATA.
# MOD BY -- KILROY, SMITH, DEWITT
# DATE -- 02 OCT 67
# AUTHORS -- KILROY, SMITH, DWWITT, DEWOLF, FAGIN
# LOG SECTION -- DOWN-TELEMETRY PROGRAM
#
# FUNCTIONAL DESCRIPTION -- THIS ROUTINE IS INITIATED BY TELEMETRY END
#       PULSE FROM THE DOWNLINK TELEMETRY CONVERTER.  THIS PULSE OCCURS
#       AT 50 TIMES PER SEC (EVERY 20 MS) THEREFORE DODOWNTM IS
#       EXECUTED AT THESE RATES.  THIS ROUTINE SELECTS THE APPROPRIATE
#       AGC DATA TO BE TRANSMITTED DOWNLINK AND LOADS IT INTO OUTPUT
#       CHANNELS 34 AND 35.  THE INFORMATION IS THEN GATED OUT FROM THE
#       LGC IN SERIAL FASHION.
#
#       THIS PROGRAM IS CODED FOR A 2 SECOND DOWNLIST.  SINCE DOWNRUPTS
#       OCCUR EVERY 20 MS AND 2 AGC COMPUTER WORDS CAN BE PLACED IN
#       CHANNELS 34 AND 35 DURING EACH DOWNRUPT THE PROGRAM IS CAPABLE
#       OF SENDING 200 AGC WORDS EVERY 2 SECONDS.
#
# CALLING SEQUENCE -- NONE
#       PROGRAM IS ENTERED VIA TCF DODOWNTM WHICH IS EXECUTED AS A
#       RESULT OF A DOWNRUPT.  CONTROL IS RETURNED VIA TCF RESUME WHICH
#       IN EFFECT IS A RESUME.
#
# SUBROUTINES CALLED -- NONE
#
# NORMAL EXIT MODE -- TCF RESUME
#
# ALARM OR ABORT EXIT MODE -- NONE
#
# RESTART PROTECTION:
#       ON A FRESH START AND RESTART THE `STARTSUB' SUBROUTINE WILL INITIALIZE THE DOWNLIST POINTER (ACTUALLY
#       DNTMGOTO) TO THE BEGINNING OF THE CURRENT DOWNLIST (I.E., CURRENT CONTENTS OF DNLSTADR).  THIS HAS THE
#       EFFECT OF IGNORING THE REMAINDER OF THE DOWNLIST WHICH THE DOWN-TELEMETRY PROGRAM WAS WORKING ON WHEN
#       THE RESTART (OR FRESH START) OCCURRED AND RESUME DOWN TELEMETRY FROM THE BEGINNING OF THE CURRENT
#       DOWNLIST.
#
#       ALSO OF INTEREST IS THE FACT THAT ON A RESTART THE AGC WILL ZERO DOWNLINK CHANNELS 13, 34 AND 35.
#
# DOWNLINK LIST SELECTION:
#       THE APPROPRIATE DOWNLINK LISTS ARE SELECTED BY THE FOLLOWING:
#       1.      FRESH START
#       2.      V37EXXE WHERE XX = THE MAJOR MODE BEING SELECTED.
#       3.      UPDATE PROGRAM (P27)
#       4.      NON-V37 SELECTABLE TYPE PROGRAMS (E.G., AGS INITIALIZATION (SUNDANCE, LUMINARY) AND P61-P62
#               TRANSITION (COLOSSUS) ETC.).
#
# DOWNLINK LIST RULES AND LIMITATIONS:
#       READ SECTION(S) WHICH FOLLOW `DEBRIS' WRITEUP.
#
# OUTPUT -- EVERY 2 SECONDS 100 DOUBLE PRECISION WORDS (I.E., 200 LGC
#       COMPUTER WORDS) ARE TRANSMITTED VIA DOWNLINK.
#
# ERASABLE INITIALIZATION REQUIRED -- NONE
```

```
#       `DNTMGOTO' AND `DNLSTADR' ARE INITIALIZED BY THE FRESH START PROGRAM.
#
# DEBRIS (ERASABLE LOCATIONS DESTROYED BY THIS PROGRAM) --
#       LDATALST, DNTMBUFF TO DNTMBUFF +21D, TMINDEX, DNQ.
```

```
# DODOWNTM IS ENTERED EVERY 20 MS BY AN INTERRUPT TRIGGERED BY THE
# RECEIPT OF AN ENDPULSE FROM THE SPACECRAFT TELEMETRY PROGRAMMER.
#
# NOTES REGARDING DOWNLINK LISTS ASSOCIATED WITH THIS PROGRAM:
# 1.     DOWNLISTS.  DOWNLISTS MUST BE COMPILED IN THE SAME BANK AS THE
#        DOWN-TELEMETRY PROGRAM.  THIS IS DONE FOR EASE OF CODING, FASTER
#        EXECUTION.
# 2.     EACH DOWNLINK LIST CONSISTS OF A CONTROL LIST AND A NUMBER OF
#        SUBLISTS.
# 3.     A SUBLIST REFERS TO A SNAPSHOT OR DATA COMMON TO THE SAME OR OTHER
#        DOWNLINK LISTS.  ANY SUBLIST CONTAINING COMMON DATA NEEDS TO BE
#        CODED ONLY ONCE FOR THE APPLICABLE DOWNLINK LISTS.
# 4.     SNAPSHOT SUBLISTS REFER SPECIFICALLY TO HOMOGENEOUS DATA WHICH MUST BE
#        SAVED IN A BUFFER DURING ONE DOWNRUPT.
# 5.     THE 1DNADR FOR THE 1ST WORD OF SNAPSHOT DATA IS FOUND AT THE END
#        OF EACH SNAPSHOT SUBLIST, SINCE THE PROGRAM CODING SENDS THIS DP WORD
#        IMMEDIATELY AFTER STORING THE OTHERS IN THE SNAPSHOT BUFFER.
# 6.     ALL LISTS ARE COMBINATIONS OF CODED ERASABLE ADDRESS CONSTANTS
#        CREATED FOR THE DOWNLIST PROGRAM.
#        A.      1DNADR                  1-WORD DOWNLIST ADDRESS.
#                SAME AS ECADR, BUT USED WHEN THE WORD ADDRESSED IS THE LEFT
#                HALF OF A DOUBLE-PRECISION WORD FOR DOWN TELEMETRY.
#        B.      2DNADR - 6DNADR         N-WORD DOWNLIST ADDRESS, N = 2 - 6.
#                SAME AS 1DNADR, BUT WITH THE 4 UNUSED BITS OF THE ECADR FORMAT
#                FILLED IN WITH 0001-0101.  USED TO POINT TO A LIST OF N DOUBLE-
#                PRECISION WORDS, STORED CONSECUTIVELY, FOR DOWN TELEMETRY.
#        C.      DNCHAN                  DOWNLIST CHANNEL ADDRESS.
#                SAME AS 1DNADR, BUT WITH PREFIX BITS 0111.  USED TO POINT TO
#                A PAIR OF CHANNELS FOR DOWN TELEMETRY.
#        D.      DNPTR                   DOWN-TELEMETRY SUBLIST POINTER.
#                SAME AS CAF BUT TAGGED AS A CONSTANT.  USED IN CONTROL LIST TO POINT TO A SUBLIST.
#                CAUTION --- A DNPTR CANNOT BE USED IN A SUBLIST.
# 7.     THE WORD ORDER CODE IS SET TO ZERO AT THE BEGINNING OF EACH DOWNLIST (I.E., CONTROL LIST) AND WHEN
#        A `1DNADR TIME2' IS DETECTED IN THE CONTROL LIST (ONLY).
# 8.     IN THE SNAPSHOT SUBLIST ONLY, THE DNADR'S CANNOT POINT TO THE FIRST WORD OF ANY EBANK.
#
# DOWNLIST LIST RESTRICTIONS:
# (THE FOLLOWING POINTS MAY BE LISTED ELSEWHERE BUT ARE LISTED HERE SO IT IS CLEAR THAT THESE THINGS CANNOT BE
# DONE)
# 1.     SNAPSHOT DOWNLIST:
#        (A) CANNOT CONTAIN THE FOLLOWING ECADRS (I.E., 1DNADR'S): Q, 400, 1000, 1400, 2000, 2400, 3000, 3400.
#        (B) CAN CONTAIN ONLY 1DNADR'S
# 2.     ALL DOWNLINKED DATA (EXCEPT CHANNELS) IS PICKED UP BY A DCA SO DOWNLINK LISTS CANNOT CONTAIN THE
#        EQUIVALENT OF THE FOLLOWING ECADRS (I.E., 1DNADRS): 377, 777, 1377, 1777, 2377, 2777, 3377, 3777.
#        (NOTE: THE TERM `EQUIVALENT' MEANT THAT THE 1DNADR TO 6DNADR WILL BE PROCESSED LIKE 1 TO 6 ECADRS)
# 3.     CONTROL LISTS AND SUBLISTS CANNOT HAVE ENTRIES = OCTAL 00000 OR OCTAL 77777
```

```
# 4.     THE `1DNADR TIME2' WHICH WILL CAUSE THE DOWNLINK PROGRAM TO SET THE WORDER CODE TO 3 MUST APPEAR IN THE
#        CONTROL SECTION OF THE DOWNLIST.
# 5.     `DNCHAN 0' CANNOT BE USED.
# 6.     `DNPTR 0' CANNOT BE USED.
# 7.     DNPTR CANNOT APPEAR IN A SUBLIST.
#
# EBANK SETTINGS
#        IN THE PROCESS OF SETTING THE EBANK (WHEN PICKING UP DOWNLINK DATA) THE DOWN TELEMETRY PROGRAM PUTS
#        `GARBAGE' INTO BITS15-12 OF EBANK.  HUGH BLAIR-SMITH WARNS US THAT BITS15-12 OF EBANK MAY BECOME
#        SIGNIFICANT SOMEDAY IN THE FUTURE.  IF/WHEN THAT HAPPENS, THE PROGRAM SHOULD INSURE (BY MASKING ETC.)
#        THAT BITS 15-12 OF EBANK ARE ZERO.
#
# INITIALIZATION REQUIRED -- TO INTERRUPT CURRENT LIST AND START A NEW ONE.
#        1. ADRES OF DOWNLINK LIST INTO DNLSTADR
#        2. NEGONE INTO SUBLIST
#        3. NEGONE INTO DNECADR

                BANK    22
                SETLOC  DOWNTELM
                BANK

                EBANK=  DNTMBUFF

                COUNT*  $$/DPROG
DODOWNTM        TS      BANKRUPT
                EXTEND
                QXCH    QRUPT           # SAVE Q
                TCF     WOTEST
WO1             EXTEND                  # SET WORD ORDER BIT TO 1 ONLY IF IT
                WOR     CHAN13          # ALREADY ISN'T
                TC      DNTMGOTO        # GOTO APPROPRIATE PHASE OF PROGRAM

DNPHASE1        CA      NEGONE          # INITIALIZE ALL CONTROL WORDS
                TS      SUBLIST         # WORDS TO MINUS ONE
                TS      DNECADR
                CA      LDNPHAS2        # SET DNTMGOTO = 0 ALL SUSEQUENT DOWRUPTS
                TS      DNTMGOTO        # GO TO DNPHASE2
                TCF     NEWLIST
DNPHASE2        CCS     DNECADR         # SENDING OF DATA IN PROGRESS
DODNADR         TC      FETCH2WD        # YES -- THEN FETCH THE NEXT 2 SP WORDS
MINTIME2        -1DNADR TIME2           # NEGATIVE OF TIME2 1DNADR
                TCF     +1              # (ECADR OF 3776 + 74001 = 77777)

                CCS     SUBLIST         # IS THE SUBLIST IN CONTROL
                TCF     NEXTINSL        # YES
```

```
DNADRDCR        OCT     74001           # DNADR COUNT AND ECADR DECREMENTER

CHKLIST         CA      CTLIST
                EXTEND
                BZMF    NEWLIST         # IT WILL BE NEGATIVE AT END OF LIST
                TCF     NEXTINCL
NEWLIST         INDEX   DNLSTCOD
                CA      DNTABLE         # INITIALIZE CTLIST WITH
                TS      CTLIST          #       STARTING ADDRESS OF NEW LIST
                CS      DNLSTCOD
                TCF     SENDID  +3
NEXTINCL        INDEX   CTLIST
                CA      0
                CCS     A
                INCR    CTLIST          # SET POINTER TO PICK UP NEXT CTLIST WORD
                TCF     +4              # ON NEXT ENTRY TO PROG.  (A SHOULD NOT =0)
                XCH     CTLIST          # SET CTLIST TO NEGATIVE AND PLACE(CODING)
                COM                     # UNCOMPLEMENTED DNADR INTO A.    (FOR LA)
                XCH     CTLIST          #                                 (ST IN )
 +4             INCR    A               #                                 (CTLIST)
                TS      DNECADR         # SAVE DNADR
                AD      MINTIME2        # TEST FOR TIME2 (NEG. OF ECADR)
                CCS     A
                TCF     SETWO   +1      # DON'T SET WORD ORDER CODE
MINB1314        OCT     47777           # MINUS BIT 13 AND 14 (CAN'T GET HERE)
                TCF     SETWO   +1      # DON'T SET WORD ORDER CODE
SETWO           TC      WOZERO          # GO SET WORD ORDER CODE TO ZERO.
 +1             CA      DNECADR         # RELOAD A WITH THE DNADR.
 +2             AD      MINB1314        # IS THIS A REGULAR DNADR?
                EXTEND
                BZMF    FETCH2WD        # YES.  (A MUST NEVER BE ZERO)
                AD      MINB12          # NO.  IS IT A POINTER (DNPTR) OR A
                EXTEND                  #       CHANNEL(DNCHAN)
                BZMF    DODNPTR         # IT'S A POINTER.  (A MUST NEVER BE ZERO)

DODNCHAN        TC      6               # (EXECUTED AS EXTEND)  IT'S A CHANNEL
                INDEX   DNECADR
                INDEX   0       -4000   # (EXECUTED AS READ)
                TS      L
                TC      6               # (EXECUTED AS EXTEND)
                INDEX   DNECADR
                INDEX   0       -4001   # (EXECUTED AS READ)
                TS      DNECADR         # SET DNECADR
                CA      NEGONE          #       TO MINUS
                XCH     DNECADR         #               WHILE PRESERVING A.
                TCF     DNTMEXIT        # GO SEND CHANNELS

WOZERO          CS      BIT7
                EXTEND
                WAND    CHAN13          # SET WORD ORDER CODE TO ZERO
```

```
                TC      Q               # RETURN TO CALLER

DODNPTR         INDEX   DNECADR         # DNECADR CONTAINS ADRES OF SUBLIST
                0       0               # CLEAR AND ADD LIST ENTRY INTO A.
                CCS     A               # IS THIS A SNAPSHOT SUBLIST
                CA      DNECADR         # NO, IT IS A REGULAR SUBLIST.
                TCF     DOSUBLST        # A MUST NOT BE ZERO.

                XCH     DNECADR         # YES.  IT IS A SNAPSHOT SUBLIST.
                TS      SUBLIST         # C(DNECADR) INTO SUBLIST
                CAF     ZERO            #       A     INTO     A
                XCH     TMINDEX         # (NOTE:  TMINDEX = DNECADR)

# THE FOLLOWING CODING (FROM SNAPLOOP TO SNAPEND) IS FOR THE PURPOSE OF TAKING A SNAPSHOT OF 12 DP REGISTERS.
# THIS IS DONE BY SAVING 11 DP REGISTERS IN DNTMBUFF AND SENDING THE FIRST DP WORD IMMEDIATELY.
# THE SNAPSHOT PROCESSING IS THE MOST TIME CONSUMING AND THEREFORE THE CODING AND LIST STRUCTURE WERE DESIGNED
# TO MINIMIZE TIME.  THE TIME OPTIMIZATION RESULTS IN RULES UNIQUE TO THE SNAPSHOT PORTION OF THE DOWNLIST.
# THESE RULES ARE ......
#       1.      ONLY 1DNADR'S CAN APPEAR IN THE SNAPSHOT SUBLIST
#       2.      THE 1DNADR'S CANNOT REFER TO THE FIRST LOCATION IN ANY BANK.

SNAPLOOP        TS      EBANK           # SET EBANK
                MASK    LOW8            # ISOLATE RELATIVE ADDRESS
                EXTEND
                INDEX   A
                EBANK=  1401
                DCA     1401            # PICK UP 2 SNAPSHOT WORDS.
                EBANK=  DNTMBUFF
                INDEX   TMINDEX
                DXCH    DNTMBUFF        # STORE 2 SNAPSHOT WORDS IN BUFFER
                INCR    TMINDEX         # SET BUFFER INDEX FOR NEXT 2 WORDS.
                INCR    TMINDEX
SNAPAGN         INCR    SUBLIST         # SET POINTER TO NEXT 2 WORDS OF SNAPSHOT
                INDEX   SUBLIST
                0       0               # = CA SSSS (SSSS = NEXT ENTRY IN SUBLIST)
                CCS     A               # TEST FOR LAST TWO WORDS OF SNAPSHOT.
                TCF     SNAPLOOP        # NOT LAST TWO.
LDNPHAS2        GENADR  DNPHASE2
                TS      SUBLIST         # YES, LAST.  SAVE A.
                CA      NEGONE          # SET DNECADR AND
                TS      DNECADR         #       SUBLIST POINTERS
                XCH     SUBLIST         #               TO NEGATIVE VALUES
                TS      EBANK
                MASK    LOW8
                EXTEND
                INDEX   A
                EBANK=  1401
                DCA     1401            # PICK UP FIRST 2 WORDS OF SNAPSHOT.
```

```
                EBANK=  DNTMBUFF
SNAPEND         TCF     DNTMEXIT        # NOW TO SEND THEM.

FETCH2WD        CA      DNECADR
                TS      EBANK           # SET EBANK
                MASK    LOW8            # ISOLATE RELATIVE ADDRESS
                TS      L
                CA      DNADRDCR        # DECREMENT COUNT AND ECADR
                ADS     DNECADR
                EXTEND
                INDEX   L
                EBANK=  1400
                DCA     1400            # PICK UP 2 DATA WORDS
                EBANK=  DNTMBUFF
                TCF     DNTMEXIT        # NOW GO SEND THEM.

DOSUBLST        TS      SUBLIST         # SET SUBLIST POINTER
NEXTINSL        INDEX   SUBLIST
                0       0               # = CA SSSS (SSSS = NEXT ENTRY IN SUBLIST)
                CCS     A               # IS IT THE END OF THE SUBLIST
                INCR    SUBLIST         # NO --
                TCF     +4
                TS      SUBLIST         # SAVE A.
                CA      NEGONE          # SET SUBLIST TO MINUS
                XCH     SUBLIST         # RETRIEVE A.
 +4             INCR    A
                TS      DNECADR         # SAVE DNADR
                TCF     SETWO +2        # GO USE COMMON CODING (PROLEMS WOULD
                                        # OCCUR IF THE PROGRAM ENCOUNTERED A
                                        # DNPTR NOW)

DNTMEXIT        EXTEND                  # DOWN-TELEMETRY EXIT
                WRITE   DNTM1           # TO SEND A + L TO CHANNELS 34 + 35
                CA      L               # RESPECTIVELY
TMEXITL         EXTEND
                WRITE   DNTM2
TMRESUME        TCF     RESUME          # EXIT TELEMETRY PROGRAM VIA RESUME.

MINB12          EQUALS  -1/8
DNECADR         EQUALS  TMINDEX
CTLIST          EQUALS  LDATALST
SUBLIST         EQUALS  DNQ
```

```
# SUBROUTINE NAME -- DNDUMP
#
# FUNCTIONAL DESCRIPTION -- TO SEND (DUMP) ALL ERASABLE STORAGE 'N' TIMES. (N=1 TO 4).  BANKS ARE SENT ONE AT A TIME
#       EACH BANK IS PRECEDED BY AN ID WORD, SYNCH BITS, ECADR AND TIME1 FOLLOWED BY THE 256D WORDS OF EACH
#       EBANK.  EBANKS ARE DUMPED IN ORDER (I.E., EBANK 0 FIRST, THEN EBANK1 ETC.)
#
# CALLING SEQUENCE -- THE GROUND OR ASTRONAUT BY KEYING V74E CAN INITIALIZE THE DUMP.
#       AFTER KEYING IN V74E THE CURRENT DOWNLIST WILL BE IMMEDIATELY TERMINATED AND THE DOWNLINK ERASABLE DUMP
#       WILL BEGIN.
#
#       ONCE INITIATED THE DOWNLINK ERASABLE DUMP CAN BE TERMINATED (AND INTERRUPTED DOWNLIST REINSTATED) ONLY
#       BY THE FOLLOWING:
#
#       1.      A FRESH START
#       2.      COMPLETION OF ALL DOWNLINK DUMPS REQUESTED (ACCORDING TO BITS SET IN DUMPCNT).  NOTE THAT DUMPCNT
#               CAN BE ALTERED BY A V21N01.
#       3.      AND INVOLUNTARILY BY A RESTART.
#
# NORMAL EXIT MODE -- TCF DNPHASE1
#
# ALARM OR ABORT MODE -- NONE
#
# *SUBROUTINES CALLED -- NONE
#
# ERASABLE INITIALIZATION REQUIRED --
#       DUMPCNT         OCT 20000       IF 4 COMPLETE ERASABLE DUMPS ARE DESIRED
#       DUMPCNT         OCT 10000       IF 2 COMPLETE ERASABLE DUMPS ARE DESIRED
#       DUMPCNT         OCT 04000       IF 1 COMPLETE ERASABLE DUMP  IS  DESIRED
#
# DEBRIS -- DUMPLOC, DUMPSW, DNTMGOTO, EBANK, AND CENTRAL REGISTERS
#
# TIMING --     TIME (IN SECS) = ((NO.DUMPS)*(NO.EBANKS)*(WDSPEREBANK + NO.IDWDS)) / NO.WDSPERSEC
#               TIME (IN SECS) =  (    4    )*(    8    )*(    256     +     4   )  /     100
#               THUS TIME (IN SECS TO SEND DUMP OF ERASABLE 4 TIMES VIA DOWNLINK) = 83.2 SECONDS
#
# STRUCTURE OF ONE EBANK AS IT IS SENT BY DOWNLINK PROGRAM --
#       (REMINDER -- THIS ONLY DESCRIBES ONE OF THE 8 EBANKS X 4 (DUMPS) = 32 EBANKS WHICH WILL BE SENT BY DNDUMP)
#
#       DOWNLIST                                W
#         WORD  TAKEN FROM CONTENTS OF  EXAMPLE O       COMMENTS
#           1   ERASID                    0177X  0       DOWNLIST I.D. FOR DOWNLINK ERASABLE DUMP (X=7 CSM, 6 LM)
#           2   LOWIDCOD                  77340  1       DOWNLINK SYNCH BITS.  (SAME ONE USED IN ALL OTHER DOWNLISTS)
#           3   DUMPLOC                   13400  1       (SEE NOTES ON DUMPLOC) 1 = 3RD ERAS DUMP, 3400=ECADR OF 5TH WD
#           4   TIME1                     14120  1       TIME IN CENTISECONDS
#           5   FIRST WORD OF EBANK X     03400  1       IN THIS EXAMPLE THIS WORD = CONTENTS OF E7,1400 (ECADR 3400)
#           6   2ND   WORD OF EBANK X     00142  1       IN THIS EXAMPLE THIS WORD = CONTENTS OF E7,1401 (ECADR 3401)
#           7   3RD   WORD OF EBANK X     00142  1       IN THIS EXAMPLE THIS WORD = CONTENTS OF E7,1402 (ECADR 3402)
#           .
#           .
#           .
#        260D   256TH WORD OF EBANK X     03777  1       IN THIS EXAMPLE THIS WORD = CONTENTS OF E7,1777 (ECADR 3777)
#
# NOTE --       DUMPLOC CONTAINS THE COUNTER AND ECADR FOR EACH WORD BEING SENT.
#               THE BIT STRUCTURE OF DUMPLOC IS FOLLOW --
#                                               X = NOT USED
#               X ABC EEE RRRRRRRR            ABC = ERASABLE DUMP COUNTER (I.E. ABC = 0,1,2, OR 3 WHICH MEANS THAT
#                                                   COMPLETE ERASABLE DUMP NUMBER 1,2,3, OR 4 RESPECTIVELY IS IN PROGRESS)
#                                             EEE = EBANK BITS
#                                        RRRRRRRR = RELATIVE ADDRESS WITHIN AN EBANK
```

1412THE

1

```
DNDUMPI         CA      ZERO            # INITIALIZE DOWNLINK
                TS      DUMPLOC         # ERASABLE DUMP
 +2             TC      SENDID          # GO SEND ID AND SYNCH BITS
                CA      LDNDUMP1        # SET DNTMGOTO
                TS      DNTMGOTO        # TO LOCATION FOR NEXT PASS
                CA      TIME1           # PLACE TIME1
                XCH     L               # INTO L
                CA      DUMPLOC         # AND ECADR OF THIS EBANK INTO A
                TCF     DNTMEXIT        # SEND DUMPLOC AND TIME1

LDNDUMP         ADRES   DNDUMP
LDNDUMP1        ADRES   DNDUMP1

DNDUMP          CA      TWO             # INCREMENT ECADR IN DUMPLOC
                ADS     DUMPLOC         # TO NEXT DP WORD TO BE
                MASK    LOW8            # DUMPED AND SAVE IT.
                CCS     A               # IS THIS THE BEGINNING OF A NEW EBANK
                TCF     DNDUMP2         # NO -- THEN CONTINUE DUMPING
                CA      DUMPLOC         # YES -- IS THIS THE END OF THE
                MASK    DUMPCNT         # N TH (N = 1 TO 4) COMPLETE ERASABLE
                MASK    PRIO34          # DUMP (BIT14 FOR 4, BIT13 FOR 2 OR BIT12
                CCS     A               # FOR 1 COMPLETE ERASABLE DUMP(S)).
                TCF     DNPHASE1        # YES -- START SENDING INTERRUPTED DOWNLIST
                                        # AGAIN
                TCF     DNDUMPI +2      # NO -- GO BACK AND INITIALIZE NEXT BANK

DNDUMP1         CA      LDNDUMP         # SET DNTMGOTO
                TS      DNTMGOTO        # FOR WORDS 3 TO 256D OF CURRENT EBANK

DNDUMP2         CA      DUMPLOC
                TS      EBANK           # SET EBANK
                MASK    LOW8            # ISOLATE RELATIVE ADDRESS.
                TS      Q               # (NOTE: MASK INSTRUCTION IS USED TO PICK
                CA      NEGO            # UP ERASABLE REGISTERS SO THAT EDITING
                TS      L               # REGISTERS 20-23 WILL NOT BE ALTERED.)
                INDEX   Q
                EBANK=  1400            # PICK UP LOW ORDER REGISTER OF PAIR
                MASK    1401            # OF ERASABLE REGISTERS.
                XCH     L
                INDEX   Q               # PICK UP HIGH ORDER REGISTER OF PAIR
                MASK    1400            # OF ERASABLE REGISTERS.
                EBANK=  DNTMBUFF
                TCF     DNTMEXIT        # GO SEND THEM

SENDID          EXTEND                  # ** ENTRANCE USED BY ERASABLE DUMP PROG. **
                QXCH    DNTMGOTO        # SET DNTMGOTO SO NEXT TIME PROG WILL GO
                CAF     ERASID          # TO LOCATION FOLLOWING `TC SENDID'

                TS      L               # ** ENTRANCE USED BY REGULAR DOWNLINK PG **
```

```
                TC      WOZERO          # GO SET WORD ORDER CODE TO ZERO
                CAF     LOWIDCOD        # PLACE SPECIAL ID CODE INTO L
                XCH     L               # AND ID BACK INTO A
                TCF     DNTMEXIT        # SEND DOWNLIST ID CODE(S).

WOTEST          CA      BIT7            # AT THE BEGINNING OF THE LIST THE WORD
                EXTEND                  # ORDER BIT WILL BE SET BACK TO ZERO
                RAND    CHAN13
                CCS     A
                TC      DNTMGOTO
                CA      BIT7
                TCF     WO1
```

```
#          THE FOLLOWING ROUTINE CAN BE USED TO CALL A SUBROUTINE IN ANOTHER BANK. IN THE BANKCALL VERSION, THE
# CADR OF THE SUBROUTINE IMMEDIATELY FOLLOWS THE  TC BANKCALL  INSTRUCTION, WITH C(A) AND C(L) PRESERVED.

                BLOCK   02
                COUNT*  $$/BANK
BANKCALL        DXCH    BUF2                    # SAVE INCOMING A,L.
                INDEX   Q                       # PICK UP CADR.
                CA      0
                INCR    Q                       # SO WE RETURN TO THE LOC. AFTER THE CADR.

#          SWCALL IS IDENTICAL TO BANKCALL, EXCEPT THAT THE CADR ARRIVES IN A.

SWCALL          TS      L
                LXCH    FBANK                   # SWITCH BANKS, SAVING RETURN.
                MASK    LOW10                   # GET SUB-ADDRESS OF CADR.
                XCH     Q                       # A,L NOW CONTAINS DP RETURN.
                DXCH    BUF2                    # RESTORING INPUTS IF THIS IS A BANKCALL.
                INDEX   Q
                TC      10000                   # SETTING Q TO SWRETURN.

SWRETURN        XCH     BUF2    +1              # COMES HERE TO RETURN TO CALLER. C(A,L)
                XCH     FBANK                   # ARE PRESERVED FOR RETURN.
                XCH     BUF2    +1
                TC      BUF2

#          THE FOLLOWING ROUTINE CAN BE USED AS A UNILATERAL JUMP WITH C(A,L) PRESERVED AND THE CADR IMMEDIATELY
# FOLLOWING THE TC POSTJUMP INSTRUCTION.

POSTJUMP        XCH     Q                       # SAVE INCOMING C(A).
                INDEX   A                       # GET CADR.
                CA      0

#          BANKJUMP IS THE SAME AS POSTJUMP, EXCEPT THAT THE CADR ARRIVES IN A.

BANKJUMP        TS      FBANK
                MASK    LOW10
                XCH     Q                       # RESTORING INPUT C(A) IF THIS WAS A
Q+10000         INDEX   Q                       # POSTJUMP.
PRIO12          TCF     10000                   # PRIO12 = TCF   10000 = 12000
```

```
#          THE FOLLOWING ROUTINE GETS THE RETURN CADR SAVED BY SWCALL OR BANKCALL AND LEAVES IT IN A.

MAKECADR        CAF     LOW10
                MASK    BUF2
                AD      BUF2    +1
                TC      Q

SUPDACAL        TS      MPTEMP
                XCH     FBANK           # SET FBANK FOR DATA.
                EXTEND
                ROR     SUPERBNK        # SAVE FBANK IN BITS 15-11, AND
                XCH     MPTEMP          #  SUPERBANK IN BITS  7-5.
                MASK    LOW10
                XCH     L               # SAVE REL. ADR. IN BANK, FETCH SUPERBITS.
                INHINT                  # BECAUSE RUPT DOES NOT SAVE SUPERBANK.
                EXTEND
                WRITE   SUPERBNK        # SET SUPERBANK FOR DATA.
                INDEX   L
                CA      10000           # PINBALL (FIX MEM DISP) PREVENTS DCA HERE
                XCH     MPTEMP          # SAVE 1ST WD, FETCH OLD FBANK AND SBANK.
                EXTEND
                WRITE   SUPERBNK        # RESTORE SUPERBANK.
                RELINT
                TS      FBANK           # RESTORE FBANK.
                CA      MPTEMP          # RECOVER FIRST WORD OF DATA.
                RETURN                  # 24 WDS. DATACALL 516 MU, SUPDACAL 432 MU
```

```
#          THE FOLLOWING ROUTINES ARE IDENTICAL TO BANKCALL AND SWCALL EXCEPT THAT THEY ARE USED IN INTERRUPT.

IBNKCALL        DXCH    RUPTREG3        # USES RUPTREG3,4 FOR DP RETURN ADDRESS.
                INDEX   Q
                CAF     0
                INCR    Q

ISWCALLL        TS      L
                LXCH    FBANK
                MASK    LOW10
                XCH     Q
                DXCH    RUPTREG3
                INDEX   Q
                TC      10000

ISWRETRN        XCH     RUPTREG4
                XCH     FBANK
                XCH     RUPTREG4
                TC      RUPTREG3

# 2. USPRCADR ACCESSES INTERPRETIVE CODING IN OTHER THAN THE USER'S FBANK.  THE CALLING SEQUENCE IS AS FOLLOWS:

# L             TC      USPRCADR
# L+1           CADR    INTPRETX          INTPRETX IS THE INTERPRETIVE CODING
#                                         RETURN IS TO L+2

USPRCADR        TS      LOC             # SAVE A
                CA      BIT8
                TS      EDOP            # EXIT INSTRUCTION TO EDOP
                CA      BBANK
                TS      BANKSET         # USER'S BBANK TO BANKSET
                INDEX   Q
                CA      0
                TS      FBANK           # INTERPRETIVE BANK TO FBANK
                MASK    LOW10           # YIELDS INTERPRETIVE RELATIVE ADDRESS
                XCH     Q               # INTERPRETIVE ADDRESS TO Q, FETCHING L+1
                XCH     LOC             # L+1 TO LOC, RETRIEVING ORIGINAL A
                TCF     Q+10000
```

```
# THERE ARE FOUR POSSIBLE SETTINGS FOR CHANNEL 07. (CHANNEL 07 CONTAINS SUPERBANK SETTING.)
#                                           PSEUDO-FIXED      OCTAL PSEUDO
# SUPERBANK      SETTING     S-REG. VALUE   BANK NUMBERS      ADDRESSES
# ----------     -------     ------------    ------------      ------------
#
# SUPERBANK 3      0XX       2000 - 3777       30 - 37        70000 - 107777    (WHERE XX CAN BE ANYTHING AND
#                                                                              WILL USUALLY BE SEEN AS 11)
# SUPERBANK 4      100       2000 - 3777       40 - 47        110000 - 127777   (AS FAR AS IT CAN BE SEEN,
#                                                                              ONLY BANKS 40-43 WILL EVER BE
#                                                                              AND ARE PRESENTLY AVAILABLE)
# SUPERBANK 5      101       2000 - 3777       50 - 57        130000 - 147777   (PRESENTLY NOT AVAILABLE TO
#                                                                              THE USER)
# SUPERBANK 6      110       2000 - 3777       60 - 67        150000 - 167777   (PRESENTLY NOT AVAILABLE TO
#                                                                              THE USER)
# ***  THIS ROUTINE MAYBE CALLED BY ANY PROGRAM LOCATED IN BANKS 00 - 27.  I.E., NO PROGRAM LIVING IN ANY
# SUPERBANK SHOULD USE SUPERSW.  ***
#
# SUPERSW MAYBE CALLED IN THIS FASHION:
#
#           CAF    ABBCON          WHERE  --  ABBCON   BBCON  SOMETHIN  --
#           TCR    SUPERSW         (THE SUPERBNK BITS ARE IN THE BBCON)
#           ...      ...
#            .        .
#            .        .
# OR IN THIS FASHION :
#           CAF    SUPERSET        WHERE SUPERSET IS ONE OF THE FOUR AVAILABLE
#           TCR    SUPERSW         SUPERBANK BIT CONSTANTS:
#           ...      ...                                  SUPER011 OCTAL  60
#            .        .                                   SUPER100 OCTAL 100
#            .        .                                   SUPER101 OCTAL 120
#                                                         SUPER110 OCTAL 140

SUPERSW         EXTEND
                WRITE   SUPERBNK        # WRITE BITS 7-6-5 OF THE ACCUMULATOR INTO
                                        # CHANNEL 07
                TC      Q               # TC TO INSTRUCTION FOLLOWING
                                        #   TC  SUPERSW
```

```
# SECTION 1:  DISPATCHER
#
# ENTRY TO THE INTERPRETER.  INTPRET SETS LOC TO THE FIRST INSTRUCTION, BANKSET TO THE BBANK OF THE
# OBJECT INTERPRETIVE PROGRAM, AND INTBIT15 TO THE BIT15 CONTENTS OF FBANK.  INTERPRETIVE PROGRAMS MAY BE IN
# VIRTUALLY ALL BANKS PRESENT UNDER ANY SUPER-BANK SETTING, WITH THE RESTRICTION THAT PROGRAMS IN HIGH BANKS
# (BIT15 OF FBANK = 1) DO NOT REFER TO LOWBANKS, AND VICE-VERSA.  THE INTERPRETER DOES NOT SWITCH SUPERBANKS.
# E-BANK SWITCHING OCCURS WHENEVER GENERAL ERASABLE (100-3777) IS ADDRESSED.

                BLOCK   03

                COUNT*  $$/INTER
INTPRET         RELINT
                EXTEND                          # SET LOC TO THE WORD FOLLOWING THE TC.
                QXCH    LOC

 +2             CA      BBANK                   # INTERPRETIVE BRANCHES FINISH HERE.
                TS      BANKSET
                MASK    BIT15                   # GET 15TH BIT FOR INDEXABLE ADDRESSES.
                TS      INTBIT15

                TS      EDOP                    # MAKE SURE NO INSTRUCTIONS LEFT OVER

                TCF     NEWOPS                  # PICK UP OP CODE PAIR AND BEGIN.

INTRSM          LXCH    BBANK                   # RESUME SUSPENDED INTERPRETIVE JOB
                TCF     INTPRET +3

# DLOAD LOADS MPAC, MPAC +1, LEAVING ZERO IN MPAC +2.

DLOAD           EXTEND
                INDEX   ADDRWD
                DCA     0                       # LOAD DP C(C(ADDRWD)) INT MPAC,MPAC +1
SLOAD2          DXCH    MPAC
                CAF     ZERO                    # ZERO MPAC +2
```

```
# AT THE END OF MOST INSTRUCTIONS, CONTROL IS GIVEN TO DANZIG TO DISPATCH THE NEXT OPERATION.

                TS      MPAC    +2              # AND DECLARE DP MODE

NEWMODE         TS      MODE                    # PROLOGUE FOR MODE-CHANGING INSTRUCTIONS.

DANZIG          CA      BANKSET                 # SET BBANK BEFORE TESTING NEWJOB SO THAT
                TS      BBANK                   # IT MAY BE SAVED DIRECTLY BY CHANJOB.

NOIBNKSW        CCS     EDOP                    # SEE IF AN ORDER CODE IS LEFT OVER FROM
                TCF     OPJUMP                  # THE LAST PAIR RETRIEVED.  IF SO, EXECUTE.
                                                # EDOP IS SET TO ZERO ON ITS RE-EDITIING.

                CCS     NEWJOB                  # SEE IF A JOB OF HIGHER PRIORITY IS
                TCF     CHANG2                  # PRESENT, AND IF SO, CHANGE JOBS.

                INCR    LOC                     # ADVANCE THE LOCATION COUNTER.

# ITRACE (1) REFERS TO "NEWOPS"
NEWOPS          INDEX   LOC                     # ENTRY TO BEGIN BY PICKING OP CODE PAIR.
                CA      0                       # MAY BE AN OPCODE PAIR OR A STORE CODE.
                CCS     A                       # TEST SIGN AND GET DABS(A).
                TCF     DOSTORE                 # PROCESS STORE CODE.

LOW7            OCT     177

                TS      EDOP                    # OP CODE PAIR.  LEAVE THE OTHER IN EDOP
                MASK    LOW7                    # WHERE CCS EDOP WILL HONOR IT NEXT.

OPJUMP          TS      CYR                     # LOWWD ENTERS HERE IF A RIGHT-HAND OP
                CCS     CYR                     # CODE IS TO BE PROCESSED.  TEST PREFICES.
                TCF     OPJUMP2                 # TEST SECOND PREFIX BIT.

                TCF     EXIT                    # +0 OP CODE IS EXIT
```

```
# PROCESS ADDRESSES WHICH MAY BE DIRECT, INDEXED, OR REFERENCE THE PUSHDOWN LIST.

ADDRESS         MASK    BIT1                    # SEE IF ADDRESS IS INDEXED.  CYR CONTAINED
                CCS     A                       # 400XX, SO BIT 1 IS NOW AS IT WAS IN CYR.
                TCF     INDEX                   # FORM INDEXED ADDRESS.

DIRADRES        INDEX   LOC                     # LOOK AHEAD TO NEXT WORD TO SEE IF
OCT40001        CS      1                       # ADDRESS IS GIVEN.
                CCS     A
                TCF     PUSHUP                  # IF NOT.

NEG4            DEC     -4

                INCR    LOC                     # IF SO, TO SHOW WE PICKED UP A WORD.
                TS      ADDRWD
```

```
# FINAL DIGESTION OF DIRECT ADDRESSES OF OP CODES WITH 01 PREFIX IS DONE HERE.   IN EACH CASE, THE
# REQUIRED 12-BIT SUB-ADDRESS IS LEFT IN ADDRWD, WITH ANY REQUIRED E OR F BANK SWITCHING DONE.  ADDRESSES LESS
# THAN 45D ARE TAKEN TO BE RELATIVE TO THE WORK AREA.  THE OP CODE IS NOW IN BITS 1-5 OF CYR WITH BIT 14 = 1.

                AD      -ENDVAC                 # SEE IF ADDRESS RELATIVE TO WORK AREA.
                CCS     A
                AD      -ENDERAS                # IF NOT, SEE IF IN GENERAL ERASABLE.
                TCF     IERASTST

NETZERO         CA      FIXLOC                  # IF SO, LEAVE THE MODIFIED ADDRESS IN
                ADS     ADDRWD                  # ADDRWD AND DISPATCH.
ITR15           INDEX   CYR                     # THIS INDEX MAKES THE NEXT INSTRUCTION
                7       INDJUMP -1              # TCF INDJUMP + OP, EDITING CYR.

IERASTST        EXTEND
                BZMF    GEADDR                  # GO PROCESS GENERAL-ERASABLE ADDRESS.

                MASK    LOW10                   # FIXED BANK ADDRESS. RESTORE AND ADD B15.
                AD      LOW10                   # SWITCH BANKS AND LEAVE SUBADDRESS IN
                XCH     ADDRWD                  # ADDRWD FOR OPERAND RETRIEVAL. (THIS
                AD      INTBIT15                # METHOD PRECLUDES USE OF THE LAST
                TS      FBANK                   # LOCATION IN EACH FBANK.)
ITR12           INDEX   CYR
                7       INDJUMP -1

GEADDR          MASK    LOW8
                AD      OCT1400
                XCH     ADDRWD
                TS      EBANK
ITR10           INDEX   CYR
                7       INDJUMP -1
```

```
# THE FOLLOWING ROUTINE PROCESSES INTERPRETIVE INDEXED ADDRESSES.  AN INTERPRETER INDEX REGISTER MAY
# CONTAIN THE ADDRESS OF ANY ERASABLE REGISTER (0-42 BEING RELATIVE TO THE VAC AREA) OR ANY INTERPRETIVE PROGRAM
# BANK, OR ANY INTEGER IN THAT RANGE.

DODLOAD*        CAF     DLOAD*                  # STODL* COMES HERE TO PROCESS LOAD ADR.
                TS      CYR                     # (STOVL* ENTERS HERE).

INDEX           CA      FIXLOC                  # SET UP INDEX LOCATION.
                TS      INDEXLOC
                INCR    LOC                     # (ADDRESS ALWAYS GIVEN).
                INDEX   LOC
                CS      0
                CCS     A                       # INDEX 2 IF ADDRESS STORED COMPLEMENTED.
                INCR    INDEXLOC
                NOOP

                TS      ADDRWD                  # 14 BIT ADDRESS TO ADDRWD.
                MASK    HIGH4                   # IF ADDRESS GREATER THAN 2K, ADD INTBIT15
                EXTEND
                BZF     INDEX2
                CA      INTBIT15
                ADS     ADDRWD

INDEX2          INDEX   INDEXLOC
                CS      X1
                ADS     ADDRWD                  # DO AUGMENT, IGNORING AND CORRECTING OVF.

                MASK    HIGH9                   # SEE IF ADDRESS IS IN WORK AREA.
                EXTEND
                BZF     INDWORK
                MASK    HIGH4                   # SEE IF IN FIXED BANK.
                EXTEND
                BZF     INDERASE

                CA      ADDRWD                  # IN FIXED -- SWITCH BANKS AND CREATE
                TS      FBANK                   # SUB-ADDRESS
                MASK    LOW10
                AD      2K
                TS      ADDRWD
ITR11           INDEX   CYR
                3       INDJUMP -1

INDWORK         CA      FIXLOC                  # MAKE ADDRWD RELATIVE TO WORK AREA.
                TCF     ITR13   -1

INDERASE        CA      OCT1400
                XCH     ADDRWD
                TS      EBANK
                MASK    LOW8
 -1             ADS     ADDRWD
```

```
ITR13           INDEX   CYR
                3       INDJUMP -1
```

```
# PUSH-UP ROUTINES.  WHEN NO OPERAND ADDRESS IS GIVEN, THE APPROPRIATE OPERAND IS TAKEN FROM THE PUSH-DOWN
# LIST.  IN MOST CASES THE MODE OF THE RESULT (VECTOR OR SCALAR) OF THE LAST ARITHMETIC OPERATION PERFORMED
# IS THE SAME AS THE TYPE OF OPERAND DESIRED (ALL ADD/SUBTRACT ETC.).  EXCEPTIONS TO THIS GENERAL RULE ARE LISTED
# BELOW (NOTE THAT IN EVERY CASE THE MODE REGISTER IS LEFT INTACT):
#
#       1.      VXSC AND V/SC WANT THE OPPOSITE TYPE OF OPERAND, E.G., IF THE LAST OPERATION YIELDED A VECTOR
#               RESULT, VXSC WANTS A SCALAR.
#
#       2.      THE LOAD CODES SHOULD LOAD THE ACCUMULATOR INDEPENDENT OF THE RESULT OF THE LAST OPERATION.  THIS
#               INCLUDES VLOAD, DLOAD, TLOAD, PDDL, AND PDVL (NO PUSHUP WITH SLOAD).
#
#       3.      SOME ARITHMETIC OPERATIONS REQUIRE A STANDARD TYPE OF OPERAND REGARDLESS OF THE PREVIOUS OPERATION.
#               THIS INCLUDES SIGN WANTING DP AND TAD REQUIRING TP.

PUSHUP          CAF     OCT23           # IF THE LOW 5 BITS OF CYR ARE LESS THAN
                MASK    CYR             # 20, THIS OP REQUIRES SPECIAL ATTENTION.
                AD      -OCT10          # (NO -0).
                CCS     A
                TCF     REGUP           # FOR ALL CODES GREATEER THAN OCT 7.

-OCT10          OCT     -10

                AD      NEG4            # WE NOW HAVE 7 -- OP CODE (MOD4).  SEE IF
                CCS     A               # THE OP CODE (MOD4) IS THREE (REVERSE).
                INDEX   A               # NO -- THE MODE IS DEFINITE.  PICK UP THE
                CS      NO.WDS
                TCF     REGUP   +2

                INDEX   MODE            # FOR VXSC AND V/SC WE WANT THE REQUIRED
                CS      REVCNT          # PUSHLOC DECREMENT WITHOUT CHANGING THE
                TCF     REGUP   +2      # MODE AT THIS TIME.

REGUP           INDEX   MODE            # MOST ALL OP CODES PUSHUP HERE.
                CS      NO.WDS
 +2             ADS     PUSHLOC
                TS      ADDRWD
ITR14           INDEX   CYR
                7       INDJUMP -1      # (THE INDEX MAKES THIS A TCF.)

                OCT     2               # REVERSE PUSHUP DECREMENT. VECTOR TAKES 2
REVCNT          OCT     6               # WORDS, SCALAR TAKES 6.
                OCT     6
NO.WDS          OCT     2               # CONVENTIONAL DECREMENT IS 6 WORDS VECTOR
OCTAL3          OCT     3               # 2 IN DP, AND 3 IN TP.
                OCT     6
```

```
# TEST THE SECOND PREFIX BIT TO SEE IF THIS IS A MISCELLANEOUS OR A UNARY/SHORT SHIFT OPERATION.

OPJUMP2         CCS     CYR             # TEST SECOND PREFIX BIT.
                TCF     OPJUMP3         # TEST THIRD BIT TO SEE IF UNARY OR SHIFT

-ENDVAC         DEC     -45

# THE FOLLOWING ROUTINE PROCESSES ADDRESSES OF SUFFIX CLASS 10.  THEY ARE BASICALLY WORK AREA ADDRESSES
# IN THE RANGE 0-52, ERASABLE ECADR CONSTANTS FROM 100-3777, AND FCADRS ABOVE THAT.  ALL 15 BITS ARE AVAILABLE
# IN CONTRAST TO SUFFIX 1, IN WHICH ONLY THE LOW ORDER 14 ARE AVAILABLE.

15BITADR        INCR    LOC             # (ENTRY HERE FROM STCALL).
                INDEX   LOC             # PICK UP ADDRESS WORD.
                CA      0
                TS      POLISH          # WE MAY NEED A SUBADDRESS LATER.

                CAF     LOW7+2K         # THESE INSTRUCTIONS ARE IN BANK 1.
                TS      FBANK
                MASK    CYR
ITR7            INDEX   A
                TCF     MISCJUMP
```

```
# COMPLETE THE DISPATCHING OF UNARY AND SHORT SHIFT OPERATIONS.

OPJUMP3         TS      FBANK           # CALL IN BANK 0 (BIT5S 11-15 OF A ARE 0.)
# ITRACE (6) REFERS TO "OPJUMP3"
                CCS     CYR             # TEST THIRD PREFIX BIT.
                INDEX   A               # THE DECREMENTED UNARY CODE IS IN BITS
                TCF     UNAJUMP         # 1-4 OF A (ZERO, EXIT, HAS BEEN DETECTED)

                CCS     MODE            # IT'S A SHORT SHIFT CODE.  SEE IF PRESENT
                TCF     SHORTT          # SCALAR OR VECTOR.
                TCF     SHORTT
                TCF     SHORTV          # CALLS THE APPROPRIATE ROUTINE.

FBANKMSK        EQUALS  BANKMASK
LVBUF           ADRES   VBUF
```

```
# THE FOLLOWING IS THE JUMP TABLE FOR OP CODES WHICH MAY HAVE INDEXABLE ADDRESSES OR MAY PUSH UP.

INDJUMP         TCF     VLOAD           # 00 -- LOAD MPAC WITH A VECTOR.
                TCF     TAD             # 01 -- TRIPLE PRECISION ADD TO MPAC.
                TCF     SIGN            # 02 -- COMPLEMENT MPAC (V OR SC) IF X NEG.
                TCF     VXSC            # 03 -- VECTOR TIMES SCALAR.
                TCF     CGOTO           # 04 -- COMPUTED GO TO.
                TCF     TLOAD           # 05 -- LOAD MPAC WITH TRIPLE PRECISION.
                TCF     DLOAD           # 06 -- LOAD MPAC WITH A DP SCALAR.
                TCF     V/SC            # 07 -- VECTOR DIVIDED BY A SCALAR.

                TCF     SLOAD           # 10 -- LOAD MPACIN SINGLE PRECISION.
                TCF     SSP             # 11 -- SET SINGLE PRECISION INTO X.
                TCF     PDDL            # 12 -- PUSH DOWN MPAC AND RE-LOAD IN DP.
                TCF     MXV             # 13 -- MATRIX POST-MULTIPLIED BY VECTOR.
                TCF     PDVL            # 14 -- PUSH DOWN AND VECTOR LOAD.
                TCF     CCALL           # 15 -- COMPUTED CALL.
                TCF     VXM             # 16 -- MATRIX PRE-MULTIPLIED BY VECTOR.
                TCF     TSLC            # 17 -- NORMALIZE MPAC (SCALAR ONLY).

                TCF     DMPR            # 20 -- DP MULTIPLY AND ROUND.
                TCF     DDV             # 21 -- DP DIVIDE BY.
                TCF     BDDV            # 22 -- DP DIVIDE INTO.
                TCF     GSHIFT          # 23 -- GENERAL SHIFT INSTRUCTION
                TCF     VAD             # 24 -- VECTOR ADD.
                TCF     VSU             # 25 -- VECTOR SUBTRACT.
                TCF     BVSU            # 26 -- VECTOR SUBTRACT FROM.
                TCF     DOT             # 27 -- VECTOR DOT PRODUCT.

                TCF     VXV             # 30 -- VECTOR CROSS PRODUCT.
                TCF     VPROJ           # 31 -- VECTOR PROJECTION.
                TCF     DSU             # 32 -- DP SUBTRACT.
                TCF     BDSU            # 33 -- DP SUBTRACT FROM.
                TCF     DAD             # 34 -- DP ADD.
                TCF                     # 35 -- AVAILABLE
                TCF     DMP1            # 36 -- DP MULTIPLY.
                TCF     SETPD           # 37 -- SET PUSH DOWN POINTER (DIRECT ONLY)

# CODES 10 AND 14 MUST NOT PUSH UP. CODE 04 MAY BE USED FOR VECTOR DECLARE BEFORE PUSHUP IF DESIRED.
```

```
# THE FOLLOWING JUMP TABLE APPLIES TO INDEX, BRANCH, AND MISCELLANEOUS INSTRUCTIONS.

MISCJUMP        TCF     AXT                 # 00 -- ADDRESS TO INDEX TRUE.
                TCF     AXC                 # 01 -- ADDRESS TO INDEX COMPLEMENTED.
                TCF     LXA                 # 02 -- LOAD INDEX FROM ERASABLE.
                TCF     LXC                 # 03 -- LOAD INDEX FROM COMPLEMENT OF ERAS.
                TCF     SXA                 # 04 -- STORE INDEX IN ERASABLE.
                TCF     XCHX                # 05 -- EXCHANGE INDEX WITH ERASABLE.
                TCF     INCR                # 06 -- INCREMENT INDEX REGISTER.
                TCF     TIX                 # 07 -- TRANSFER ON INDEX.

                TCF     XAD                 # 10 -- INDEX REGISTER ADD FROM ERASABLE.
                TCF     XSU                 # 11 -- INDEX SUBTRACT FROM ERASABLE.
                TCF     BZE/GOTO            # 12 -- BRANCH ZERO AND GOTO
                TCF     BPL/BMN             # 13 -- BRANCH PLUS AND BRANCH MINUS.
                TCF     RTB/BHIZ            # 14 -- RETURN TO BASIC AND BRANCH HI ZERO.
                TCF     CALL/ITA            # 15 -- CALL AND STORE QPRET.
                TCF     SW/                 # 16 -- SWITCH INSTRUCTIONS AND AVAILABLE.
                TCF     BOV(B)              # 17 -- BRANCH ON OVERFLOW TO BASIC OR INT.
```

```
# THE FOLLOWING JUMP TABLE APPLIES TO UNARY INSTRUCTIONS.

                COUNT*  $$/INTER
                BANK    0               # 00 -- EXIT -- DETECTED EARLIER.
UNAJUMP         TCF     SQRT            # 01 -- SQUARE ROOT.
                TCF     SINE            # 02 -- SIN.
                TCF     COSINE          # 03 -- COS.
                TCF     ARCSIN          # 04 -- ARC SIN.
                TCF     ARCCOS          # 05 -- ARC COS.
                TCF     DSQ             # 06 -- DP SQUARE.
                TCF     ROUND           # 07 -- ROUND TO DP.

                TCF     COMP            # 10 -- COMPLEMENT VECTOR OR SCALAR
                TCF     VDEF            # 11 -- VECTOR DEFINE.
                TCF     UNIT            # 12 -- UNIT VECTOR.
                TCF     ABVALABS        # 13 -- LENGTH OF VECTOR OR MAG OF SCALAR.
                TCF     VSQ             # 14 -- SQUARE OF LENGTH OF VECTOR.
                TCF     STADR           # 15 -- PUSH UP ON STORE CODE.
                TCF     RVQ             # 16 -- RETURN VIA QPRET.
                TCF     PUSH            # 17 -- PUSH MPAC DOWN.
```

```
# SECTION 2     LOAD AND STORE PACKAGE.
#
# A SET OF EIGHT STORE CODES IS PROVIDED AS THE PRIMARY METHOD OF STORING THE MULTI-PURPOSE
# ACCUMULATOR (MPAC).  IF IN THE DANZIG SECTION LOC REFERS TO AN ALGEBRAICALLY POSITIVE WORD, IT IS TAKEN AS A
# STORE CODE WITH A CORRESPONDING ERASABLE ADDRESS.  MOST OF THESE CODES ARE TWO ADDRESS, SPECIFYING THAT THE WORD
# FOLLOWING THE STORE CODE IS TO BE USED AS AN ADDRESS FROM WHICH TO RE-LOAD MPAC.  FOUR OPTIONS ARE AVAILABLE:
#
#       1. STORE        STORE MPAC.  THE E ADDRESS MAY BE INDEXED.
#       2. STODL        STORE MPAC AND RE-LOAD IT IN DP WITH THE NEXT ADDRESS (THE LOAD MAY BE INDEXED).
#       3. STOVL        STORE MPAC AND RE-LOAD A VECTOR (AS ABOVE).
#       4. STCALL       STORE AND DO A CALL (BOTH ADDRESES MUST BE DIRECT HERE).
#
# STODL AND STOVL WILL TAKE FROM THE PUSH-DOWN LIST IF NO LOAD ADDRESS IS GIVEN.

                BLOCK   3

                COUNT*  $$/INTER
STADR           CA      BANKSET         # THE STADR CODE (PUSHUP UP ON STORE
                TS      FBANK           # ADDRESS) ENTERS HERE.
                INCR    LOC
ITR1            INDEX   LOC             # THE STORECODE WAS STORED COMPLEMENTED TO
                CS      0               # MAKE IT LOOK LIKE AN OPCODE PAIR.
                AD      NEGONE          # (YUL CAN'T REMOVE 1 BECAUSE OF EARLY CCS)

DOSTORE         TS      ADDRWD
                MASK    LOW11           # ENTRY FROM DISPATCHER.  SAVE THE ERASABLE
                XCH     ADDRWD          # ADDRESS AND JUMP ON THE STORE CODE NO.
                MASK    B12T14
                EXTEND
                MP      BIT5            # EACH TRANSFER VECTOR ENTRY IS TWO WORDS.
                INDEX   A
                TCF     STORJUMP
```

```
# STORE CODE JUMP TABLE.  CALLS THE APPROPRIATE STORING ROUTINE AND EXITS TO DANZIG OR TO ADDRESS WITH
# A SUPPLIED OPERATION CODE.
#
# STORE STORE,1 AND STORE,2 RETURN TO DANZIG, THUS RESETTING THE EBANK TO ITS STATE AT INTPRET.

STORJUMP        TC      STORE           # STORE.
                TCF     DANZIG          # PICK UP NEW OP CODE(S).
                TC      STORE,1
                TCF     DANZIG
                TC      STORE,2
                TCF     DANZIG

                TC      STORE           # STODL.
                TCF     DODLOAD

                TC      STORE           # STODL WITH INDEXED LOAD ADDRESS.
                TCF     DODLOAD*

                TC      STORE           # STOVL.
                TCF     DOVLOAD

                TC      STORE           # STOVL WITH INDEXED LOAD ADDRESS.
                TCF     DOVLOAD*
                TC      STORE           # STOTC.
                CAF     CALLCODE
                TS      CYR
                TCF     15BITADR        # GET A 15 BIT ADDRESS.
```

```
# STORE CODE ADDRESS PROCESSOR.

STORE,1         INDEX   FIXLOC
                CS      X1
                TCF     PRESTORE

STORE,2         INDEX   FIXLOC
                CS      X2
PRESTORE        ADS     ADDRWD          # RESULTANT ADDRESS IS IN ERASABLE.

STORE           CS      ADDRWD
                AD      DEC45
                CCS     A               # DOES THE ADDRESS POINT TO THE WORK AREA?
                CA      FIXLOC          # YES.
                TCF     AHEAD5
                CA      OCT1400         # NO.  SET EBANK & MAKE UP SUBADDRESS.
                XCH     ADDRWD
                TS      EBANK
                MASK    LOW8
AHEAD5          ADS     ADDRWD
```

```
# STORING ROUTINES.  STORE DP, TP, OR VECTOR AS INDICATED BY MODE.

STARTSTO        EXTEND                          # MPAC,+1 MUST BE STORED IN ANY EVENT.
# ITRACE (5) REFERS TO "STARTSTO".
                DCA     MPAC
                INDEX   ADDRWD
                DXCH    0

                CCS     MODE
                TCF     TSTORE
                TC      Q

VSTORE          EXTEND
                DCA     MPAC     +3
                INDEX   ADDRWD
                DXCH    2

                EXTEND
                DCA     MPAC     +5
                INDEX   ADDRWD
                DXCH    4
                TC      Q

TSTORE          CA      MPAC    +2
                INDEX   ADDRWD
                TS      2
                TC      Q
```

```
# ROUTINES TO BEGIN PROCESSING OF THE SECOND ADDRES ASSOCIATED WITH ALL STORE-TYPE CODES EXCEPT STORE
# ITSELF.

DODLOAD         CAF     DLOADCOD
                TS      CYR
                TCF     DIRADRES        # GO GET A DIRECT ADDRESS.

DOVLOAD         CAF     VLOADCOD
                TS      CYR
                TCF     DIRADRES

DOVLOAD*        CAF     VLOAD*
                TCF     DODLOAD* +1     # PROLOGUE TO INDEX ROUTINE.
```

```
# THE FOLLOWING LOAD INSTRUCTIONS ARE PROVIDED FOR LOADING THE MULTI-PURPOSE ACCUMULATOR MPAC.

TLOAD           INDEX   ADDRWD
                CA      2               # LOAD A TRIPLE PRECISION ARGUMENT INTO
                TS      MPAC    +2      # THE FIRST THREE MPAC REGISTERS, WITH THE
                EXTEND                  # CONTENTS OF THE OTHER FOUR IRRELEVANT.
                INDEX   ADDRWD
                DCA     0
                DXCH    MPAC
TMODE           CAF     ONE
                TCF     NEWMODE         # DECLARE TRIPLE PRECISION MODE.

SLOAD           ZL                      # LOAD A SINGLE PRECISION NUMBER INTO
                INDEX   ADDRWD          # MPAC, SETTING MPAC+1,2 TO ZERO.  THE
                CA      0               # CONTENTS OF THE REMAINING MPAC REGISTERS
                TCF     SLOAD2          # ARE IRRELEVANT.

VLOAD           EXTEND                  # LOAD A DOUBLE PRECISION VECTOR INTO
                INDEX   ADDRWD          # MPAC,+1, MPAC+3,4, AND MPAC+5,6.  THE
                DCA     0               # CONTENTS OF MPAC +2 ARE IRRELEVANT.
                DXCH    MPAC

ENDVLOAD        EXTEND                  # PDVL COMES HERE TO FINISH UP FOR DP, TP.
                INDEX   ADDRWD
                DCA     2
                DXCH    MPAC    +3

 +4             EXTEND                  # TPDVL FINISHES HERE.
                INDEX   ADDRWD
                DCA     4
                DXCH    MPAC    +5

VMODE           CS      ONE             # DECLARE VECTOR MODE.
                TCF     NEWMODE
```

```
# THE FOLLOWING INSTRUCTIONS ARE PROVIDED FOR STORING OPERANDS IN THE PUSHDOWN LIST:
#        1.      PUSH            PUSHDOWN AND NO LOAD.
#        2.      PDDL            PUSHDOWN AND DOUBLE PRECISION LOAD.
#        3.      PDVL            PUSHDOWN AND VECTOR LOAD.

PDDL            EXTEND
                INDEX   ADDRWD          # LOAD MPAC,+1, PUSHING THE FORMER
                DCA     0               # CONTENTS DOWN.
                DXCH    MPAC
                INDEX   PUSHLOC
                DXCH    0

                INDEX   MODE            # ADVANCE THE PUSHDOWN POINTER APPRO-
                CAF     NO.WDS          # PRIATELY.
                ADS     PUSHLOC

                CCS     MODE
                TCF     ENDTPUSH
                TCF     ENDDPUSH

                TS      MODE            # NOW DP.
ENDVPUSH        TS      MPAC    +2
                DXCH    MPAC    +3      # PUSH DOWN THE REST OF THE VECTOR HERE.
                INDEX   PUSHLOC
                DXCH    0       -4

                DXCH    MPAC    +5
                INDEX   PUSHLOC
                DXCH    0       -2

                TCF     DANZIG

ENDDPUSH        TS      MPAC    +2      # SET MPAC +2 TO ZERO AND EXIT ON DP.
                TCF     DANZIG

ENDTPUSH        TS      MODE
                XCH     MPAC    +2      # ON TRIPLE, SET MPAC +2 TO ZERO, PUSHING
 +2             INDEX   PUSHLOC         # DOWN THE OLD CONTENTS
                TS      0       -1
                TCF     DANZIG
```

```
# PDVL -- PUSHDOWN AND VECTOR LOAD

PDVL            EXTEND                          # RELOAD MPAC AND PUSH DOWN ITS CONTENTS.
                INDEX   ADDRWD
                DCA     0
                DXCH    MPAC
                INDEX   PUSHLOC
                DXCH    0

                INDEX   MODE                    # ADVANCE THE PUSHDOWN POINTER.
                CAF     NO.WDS
                ADS     PUSHLOC

                CCS     MODE                    # TEST PAST MODE.
                TCF     TPDVL
                TCF     ENDVLOAD                # JUST LOAD LAST FOUR REGISTERS ON DP.

VPDVL           EXTEND                          # PUSHDOWN AND RE-LOAD LAST TWO COMPONENTS
                INDEX   ADDRWD
                DCA     2
                DXCH    MPAC    +3
                INDEX   PUSHLOC
                DXCH    0       -4

                EXTEND
                INDEX   ADDRWD
                DCA     4
                DXCH    MPAC    +5
                INDEX   PUSHLOC
                DXCH    0       -2

                TCF     DANZIG

TPDVL           EXTEND                          # ON TP, WE MUST LOAD THE Y COMPONENT
                INDEX   ADDRWD                  # BEFORE STORING MPAC +2 IN CASE THIS IS A
                DCA     2                       # PUSHUP.
                DXCH    MPAC    +3

                CA      MPAC    +2
                INDEX   PUSHLOC                 # IN DP.
                TS      0       -1
                TCF     ENDVLOAD +4

# SSP (STORE SINGLE PRECISION) IS EXECUTED HERE.

SSP             INCR    LOC                     # PICK UP THE WORD FOLLOWING THE GIVEN
                INDEX   LOC                     # ADDRESS AND STORE IT AT X.
                CA      0
STORE1          INDEX   ADDRWD                  # SOME INDEX AND MISCELLANEOUS OPS END
                TS      0                       # HERE.
```

TCF DANZIG

```
# SEQUENCE CHANGING AND SUBROUTINE CALLING OPTIONS.
#
# THE FOLLOWING OPERATIONS ARE AVAILABLE FOR SEQUENCING CHANGING, BRANCHING, AND CALLING SUBROUTINES:
#       1.      GOTO            GO TO.
#       2.      CALL            CALL SUBROUTINE SETTING QPRET.
#       3.      CGOTO           COMPUTED GO TO.
#       4.      CCALL           COMPUTED CALL.
#       7.      BPL             BRANCH IF MPAC POSITIVE OR ZERO.
#       8.      BZE             BRANCH IF MPAC ZERO.
#       9.      BMN             BRANCH IF MPAC NEGATIVE NON-ZERO.

CCALL           INCR    LOC             # MAINTAIN LOC FOR QPRET COMPUTATION
                INDEX   LOC
                CAF     0               # GET BASE ADDRESS OF CADR LIST.
                INDEX   ADDRWD
                AD      0               # ADD INCREMENT.
                TS      FBANK           # SELECT DESIRED CADR.
                MASK    LOW10
                INDEX   A
                CAF     10000
                TS      POLISH

CALL            CA      BANKSET         # FOR ANY OF THE CALL OPTIONS, MAKE UP THE
                MASK    BANKMASK        # ADDRESS OF THE NEXT OP-CODE PAIR/STORE
                AD      BANKMASK        # CODE AND LEAVE IT IN QPRET.  NOTE THAT
                AD      LOC             # BANKMASK = -(2000 - 1).
                INDEX   FIXLOC
                TS      QPRET

GOTO            CA      POLISH          # BASIC BRANCHING SEQUENCE.
 +1             MASK    HIGH4
                EXTEND
                BZF     GOTOERS         # SEE IF ADDRESS POINTS TO FIXED OR ERAS.
 +4             CA      BANKSET         # SET EBANK PART OF BBANK.  NEXT, SET UP
                TS      BBANK           # FBANK.  THE COMBINATION IS PICKED UP &
                CA      POLISH          # PUT INTO BANKSET AT INTPRET +2.
                TS      FBANK
                MASK    LOW10
                AD      2K
                TS      LOC
                TCF     INTPRET +3

                EBANK=  1400            # SO YUL DOESN'T CUSS THE "CA 1400" BELOW.

GOTOERS         CA      POLISH          # THE GIVEN ADDRESS IS IN ERASABLE -- SEE
                AD      -ENDVAC         # IF RELATIVE TO THE WORK ARA.
                CCS     A
                CA      POLISH          # GENERAL ERASABLE.
                TCF     GOTOGE
```

```
                CA      FIXLOC          # WORK AREA.
                AD      POLISH
                INDEX   A               # USE THE GIVEN ADDRESS AS THE ADDRESS OF
                CA      0               # THE BRANCH ADDRESS.
                TS      POLISH
                TCF     GOTO    +1      # ALLOWS ARBITRARY INDIRECTNESS LEVELS.

GOTOGE          TS      EBANK
                MASK    LOW8
                INDEX   A               # USE THE GIVEN ADDRESS AS THE ADDRESS OF
                CA      1400            # THE BRANCH ADDRESS.
                TS      POLISH
                TCF     GOTO    +1

CGOTO           INDEX   LOC             # COMPUTED GO TO.  PICK UP ADDRESS OF CADR
                CA      1               # LIST
                INDEX   ADDRWD          # ADD MODIFIER.
                AD      0
                TS      FBANK           # SELECT GOTO ADDRESS
                MASK    LOW10
                INDEX   A
                CA      10000
                TS      POLISH
                TCF     GOTO    +1      # WITH ADDRESS IN A.

SWBRANCH        CA      BANKSET         # SWITCH INSTRUCTIONS WHICH ELECT TO
                TS      FBANK           # BRANCH COME HERE TO DO SO.
                INDEX   LOC
                CA      1
                TS      POLISH
                TCF     GOTO    +1
```

```
# TRIPLE PRECISION BRANCHING ROUTINE.  IF CALLING TC IS AT L, RETURN IS AS FOLLOWS:
#        L+1      IF MPAC IS GREATER THAN ZERO.
#        L+2      IF MPAC IS EQUAL TO +0 OR -0.
#        L+3      IF MPAC IS LESS THAN ZERO.

BRANCH          CCS     MPAC
                TC      Q
                TCF     +2              # ON ZERO.
                TCF     NEG

                CCS     MPAC    +1
                TC      Q
                TCF     +2
                TCF     NEG

                CCS     MPAC    +2
                TC      Q
                TCF     +2
                TCF     NEG

Q+1             INDEX   Q
                TC      1

NEG             INDEX   Q               # IF FIRST NON-ZERO REGISTER WAS NEGATIVE.
                TC      2

Q+2             =       NEG

# ITRACE (3) REFERS TO "EXIT".

EXIT            CA      BANKSET         # RESTORE USER'S BANK SETTING, AND LEAVE
                TS      BBANK           # INTERPRETIVE MODE.
                INDEX   LOC
                TC      1
```

```
# SECTION 3 -- ADD/SUBTRACT PACKAGE.
#
# THE FOLLOWING OPERATIONS ARE PROVIDED FOR ADDING TO AND SUBTRACTING FROM THE MULTI-PURPOSE ACCUMULATOR
# MPAC:
#       1.      DAD     DOUBLE PRECISION ADD.
#       2.      DSU     DOUBLE PRECISION SUBTRACT.
#       3.      BDSU    DOUBLE PRECISION SUBTRACT FROM.
#       4.      TAD     TRIPLE PRECISION ADD.
#       5.      VAD     VECTOR ADD.
#       6.      VSU     VECTOR SUBTRACT.
#       7.      BVSU    VECTOR SUBTRACT FROM.
# THE INTERPRETIVE OVERFLOW INDICATOR OVFIND IS SET NON-ZERO IF OVERFLOW OCCURS IN ANY OF THE ABOVE.

VSU             CAF     BIT15           # CHANGES 0 TO DCS.
                TCF     +2

VAD             CAF     PRIO30          # CHANGES 0 TO DCA.
                ADS     ADDRWD
                EXTEND
                INDEX   ADDRWD
                READ    HISCALAR        # DCA 2 OR DCS 2
                DAS     MPAC    +3
                EXTEND                  # CHECK OVERFLOW.
                BZF     +2
                TC      OVERFLWY

                EXTEND
                INDEX   ADDRWD
                READ    CHAN5           # DCA 4 OR DCS 4
                DAS     MPAC    +5
                EXTEND
                BZF     +2
                TC      OVERFLWZ

                EXTEND
                INDEX   ADDRWD
                READ    LCHAN           # DCA 0 OR DCS 0
                TCF     ENDVXV

DAD             EXTEND
                INDEX   ADDRWD
                DCA     0
ENDVXV          DAS     MPAC            # VXV FINISHES HERE.
                EXTEND
                BZF     DANZIG
```

```
SETOVF          TC      OVERFLOW
                TCF     DANZIG
```

```
DSU             EXTEND
                INDEX   ADDRWD
                DCS     0
                TCF     ENDVXV

OVERFLWZ        TS      L                 # ENTRY FOR THIRD COMPONENT.
                CAF     FIVE
                TCF     +3

OVERFLWY        TS      L                 # ENTRY FOR SECOND COMPONENT.
                CAF     THREE
                XCH     L

OVERFLOW        INDEX   A                 # ENTRY FOR 1ST COMP OR DP (L=0).
                CS      LIMITS            # PICK UP POSMAX OR NEGMAX.
                TS      BUF
                EXTEND
                AUG     A                 # FORCE OVERFLOW.
                INDEX   L
                ADS     MPAC    +1
                TS      7
                CAF     ZERO
                AD      BUF
                INDEX   L
                ADS     MPAC
                TS      7
                TC      Q                 # NO OVERFLOW EXIT.
                TCF     SETOVF2           # SET OVFIND AND EXIT.

BVSU            EXTEND
                INDEX   ADDRWD
                DCA     2
                DXCH    MPAC    +3
                EXTEND
                DCOM
                DAS     MPAC    +3
                EXTEND
                BZF     +2
                TC      OVERFLWY

                EXTEND
                INDEX   ADDRWD
                DCA     4
                DXCH    MPAC    +5
                EXTEND
                DCOM
                DAS     MPAC    +5
                EXTEND
                BZF     +2
                TC      OVERFLWZ
```

```
BDSU            EXTEND
                INDEX   ADDRWD
                DCA     0
                DXCH    MPAC
                EXTEND
                DCOM
                TCF     ENDVXV
```

```
# TRIPLE PRECISION ADD ROUTINE.

TAD             EXTEND
                INDEX   ADDRWD
                DCA     1               # ADD MINOR PARTS FIRST.
                DAS     MPAC    +1
                INDEX   ADDRWD
                AD      0
                AD      MPAC
                TS      MPAC
                TCF     DANZIG

                TCF     SETOVF          # SET OVFIND IF SUCH OCCURS.
```

```
# ARITHMETIC SUBROUTINES REQUIRED IN FIXED-FIXED.
#       1.  DMPSUB      DOUBLE PRECISION MULTIPLY, MULTIPLY THE CONTENTS OF MPAC,+1 BY THE DP WORD WHOSE ADDRESS
#                       IS IN ADDRWD AND LEAVE A TRIPLE-PRECISION RESULT IN MPAC.
#       2.  ROUNDSUB    ROUND THE TRIPLE PRECISION CONTENTS OF MPAC TO DOUBLE PRECISION.
#       3.  DOTSUB      TAKE THE DOT PRODUCT OF THE VECTOR IN MPAC AND THE VECTOR WHOSE ADDRESS IS IN ADDRWD
#                       AND LEAVE THE TRIPLE PRECISION RESULT IN MPAC.
#       4.  POLY        USING THE CONTENTS OF MPAC AS A DP ARGUMENT, EVALUATE THE POLYNOMIAL WHOSE DEGREE AND
#                       COEFFICIENTS IMMEDIATELY FOLLOW THE TC POLY INSTRUCTION (SEE ROUTINE FOR DETAILS).

DMP             INDEX   Q               # BASIC SUBROUTINE FOR USE BY PINBALL, ETC
                CAF     0               # ADRES OF ARGUMENT FOLLOWS  TC DMP .
                INCR    Q
 -1             TS      ADDRWD          # (PROLOGUE FOR SETTING ADDRWD.)

DMPSUB          INDEX   ADDRWD          # GET MINOR PART OF OPERAND AT C(ADDRWD).
                CA      1
                TS      MPAC    +2      # THIS WORKS FOR SQUARING MPAC AS WELL.
                CAF     ZERO            # SET MPAC +1 TO ZERO SO WE CAN ACCUMULATE
                XCH     MPAC    +1      # THE PARTIAL PRODUCTS WITH DAS
                TS      MPTEMP          # INSTRUCTIONS.
                EXTEND
                MP      MPAC    +2      # MINOR OF MPAC X MINOR OF C(ADDRWD).

                XCH     MPAC    +2      # DISCARD MINOR PART OF ABOVE RESULT AND
                EXTEND                  # FORM MAJOR OF MPAC X MINOR OF C(ADDRWD).
                MP      MPAC
                DAS     MPAC    +1      # GUARANTEED NO OVERFLOW.

                INDEX   ADDRWD          # GET MAJOR PART OF ARGUMENT AT C(ADDRWD).
                CA      0
                XCH     MPTEMP          # SAVE AND BRING OUT MINOR OF MPAC.
DMPSUB2         EXTEND
                MP      MPTEMP          # MAJOR OF C(ADDRWD) X MINOR OF MPAC.
                DAS     MPAC    +1      # ACCUMULATE, SETTING A TO NET OVERFLOW.

                XCH     MPAC            # SETTING MPAC TO 0 OR +-1.
                EXTEND
                MP      MPTEMP          # MAJOR OF MPAC X MAJOR OF C(ADDRWD).
                DAS     MPAC            # GUARANTEED NO OVERFLOW.
                TC      Q               # 49 MCT = .573 MS. INCLUDING RETURN.
```

```
# ROUND MPAC TO DOUBLE PRECISION, SETTING OVFIND ON THE RARE EVENT OF OVERFLOW.

ROUNDSUB        CAF     ZERO            # SET MPAC +2 = 0 FOR SCALARS AND CHANGE
 +1             TS      MODE            # MODE TO DP.

VROUND          XCH     MPAC    +2      # BUT WE NEEDN'T TAKE THE TIME FOR VECTORS.
                DOUBLE
                TS      L
                TC      Q

                AD      MPAC    +1      # ADD ROUDING BIT IF MPAC +2 WAS GREATER
                TS      MPAC    +1      # THAN .5 IN MAGNITUDE.
                TC      Q

                AD      MPAC            # PROPAGATE INTERFLOW.
                TS      MPAC
                TC      Q

SETOVF2         TS      OVFIND          # (RARE).
                TC      Q
```

```
# THE DOT PRODUCT SUBROUTINE USUALLY FORMS THE DOT PRODUCT OF THE VECTOR IN MPAC WITH A STANDARD SIX
# REGISTER VECTOR WHOSE ADDRESS IS IN ADDRWD.  IN THIS CASE C(DOTINC) ARE SET TO 2.   VXM, HOWEVER, SETS C(DOTINC) TO
# 6 SO THAT DOTSUB DOTS MPAC WITH A COLUMN VECTOR OF THE MATRIX IN QUESTION IN THIS CASE.

PREDOT          CAF     TWO             # PROLOGUE TO SET DOTINC TO 2.
                TS      DOTINC

DOTSUB          EXTEND
                QXCH    DOTRET          # SAVE RETURN
                TC      DMPSUB          # DOT X COMPONENTS.
                DXCH    MPAC    +3      # POSITION Y COMPONENT OF MPAC FOR
                DXCH    MPAC            # MULTIPLICATION WHILE SAVING RESULT IN
                DXCH    BUF             # THREE WORD BUFFER, BUF.
                CA      MPAC    +2
                TS      BUF     +2

                CA      DOTINC          # ADVANCE ADDRWD TO Y COMPONENT OF
                ADS     ADDRWD          # OTHER ARGUMENT.
                TC      DMPSUB
                DXCH    MPAC    +1      # ACCUMULATE PARTIAL PRODUCTS.
                DAS     BUF     +1
                AD      MPAC
                AD      BUF
                TS      BUF
                TCF     +2
                TS      OVFIND          # IF OVERFLOW OCCURS.

                DXCH    MPAC    +5      # MULTIPLY Z COMPONENTS.
                DXCH    MPAC
                CA      DOTINC
                ADS     ADDRWD
                TC      DMPSUB
ENDDOT          DXCH    BUF     +1      # LEAVE FINAL ACCUMULATION IN MPAC.
                DAS     MPAC    +1
                AD      MPAC
                AD      BUF
                TS      MPAC
                TC      DOTRET

                TC      OVERFLOW        # ON OVERFLOW HERE.
                TC      DOTRET
```

```
# DOUBLE PRECISION POLYNOMIAL EVALUATOR
#                                  N        N-1
#       THIS ROUTINE EVALUATES A X  + A    X     + ... + A  X + A   LEAVING THE DP RESULT IN MPAC ON EXIT.
#                                N      N-1              1      0
#
# THE ROUTINE HAS TWO ENTRIES
#
#       1       ENTRY THRU POWRSERS.  THE COEFFICIENTS MAY BE EITHER IN FIXED OR ERASABLE E.  THE CALL IS BY
#               TC POWRSERS, AND THE RETURN IS TO LOC(TC POWRSERS)+1.  THE ENTERING DATA MUST BE AS FOLLOWS:
#
#                       A               SP      LOC-3           # ADDRESS FOR REFERENCING COEF TABLE
#                       L               SP      N-1             # N IS THE DEGREE OF THE POWER SERIES
#                       MPAC            DP      X               # ARGUMENT
#                       LOC-2N          DP      A(0)
#                                       ...
#                       LOC             DP      A(N)
#
#       2.      ENTRY THRU POLY.  THE CALL TO POLY AND THE ENTERING DATA MUST BE AS FOLLOWS
#
#                       MPAC            DP      X               # ARGUMENT
#                       LOC             TC      POLY
#                       LOC+1           SP      N-1
#                       LOC+2           DP      A(0)
#                                       ...
#                       LOC+2N+2        DP      A(N)            # RETURN IS TO LOC+2N+4

POWRSERS        EXTEND
                QXCH    POLYRET         # RETURN ADDRESS
                TS      POLISH          # POWER SERIES ADDRESS
                LXCH    POLYCNT         # N-1 TO COUNTER
                TCF     POLYCOM         # SKIP SET UP BY POLY

POLY            INDEX   Q
                CAF     0
                TS      POLYCNT         # N-1 TO COUNTER
                DOUBLE
                AD      Q
                TS      POLISH          # L(A(N))-3 TO POLISH
                AD      FIVE
                TS      POLYRET         # STORE RETURN ADDRESS

POLYCOM         CAF     LVBUF           # INCOMING X WILL BE MOVED TO VBUF, SO
                TS      ADDRWD          # SET ADDRWD SO DMPSUB WILL MPY BY VBUF.

                EXTEND
                INDEX   POLISH
                DCA     3
```

```
                DXCH    MPAC            # LOAD A(N) INTO MPAC
                DXCH    VBUF            # SAVING X IN VBUF
                TCF     POLY2

POLYLOOP        TS      POLYCNT         # SAVE DECREMENTD LOOP COUNTER
                CS      TWO
                ADS     POLISH          # REGRESS COEFFICIENT POINTER

POLY2           TC      DMPSUB          # MULTIPLY BY X
                EXTEND
                INDEX   POLISH
                DCA     1               # ADD IN NEXT COEFFICIENT
                DAS     MPAC            # USER'S RESPONSIBILITY TO ASSURE NO OVFLOW

                CCS     POLYCNT
                TCF     POLYLOOP
                TC      POLYRET         # RETURN CALLER
```

```
# MISCELLANEOUS MULTI-PRECISION ROUTINES REQUIRED IN FIXED-FIXED BUT NOT USED BY THE INTERPRETER.

DPAGREE         CAF     ZERO            # DOUBLE PRECISION ENTRY --
                TS      MPAC    +2      # ZERO LOW-ORDER WORD

TPAGREE         LXCH    Q               # FORCE SIGN AGREEMENT AMONG THE TRIPLE
                TC      BRANCH          # PRECISION CONTENTS OF MPAC.  RETURNING
                TCF     ARG+            # WITH SIGNUM OF THE INPUT IN A.
                TCF     ARGZERO

                CS      POSMAX          # IF NEGATIVE.
                TCF             +2

ARG+            CAF     POSMAX
                TS      Q
                EXTEND
                AUG     A               # FORMS +-1.0.
                AD      MPAC    +2
                TS      MPAC    +2
                CAF     ZERO
                AD      Q
                AD      MPAC    +1
                TS      MPAC    +1
                CAF     ZERO
                AD      Q               # Q STILL HAS POSMAX OR NEGMAX IN IT.
                AD      MPAC
ARGZERO2        TS      MPAC            # ALWAYS SKIPPING UNLESS ARGZERO.
                TS      MPAC    +1
                TC      L               # RETURN VIA L.

ARGZERO         TS      MPAC    +2      # SET ALL THREE MPAC REGISTERS TO ZERO.
                TCF     ARGZERO2

# SHORTMP MULTIPLIES THE TP CONTENTS OF MPAC BY THE SINGLE PRECISION NUMBER ARRIVING IN A.

SHORTMP         TS      MPTEMP
                EXTEND
                MP      MPAC    +2
                TS      MPAC    +2
SHORTMP2        CAF     ZERO            # SO SUBSEQUENT DAS WILL WORK.
                XCH     MPAC    +1
                TCF     DMPSUB2
```

```
# DMPNSUB MULTIPLIES THE DP FRACTION ARRIVING IN MPAC BY THE SP
# INTEGER ARRIVING IN A.  THE DP PRODUCT DEPARTS BOTH IN MPAC AND IN
# A AND L.  NOTE THAT DMPNSUB NORMALLY INCREASES THE MAGNITUDE OF THE
# CONTENTS OF MPAC.  THE CUSTOMER MUST INSURE THAT B(A) X B(MPAC,MPAC+1)
# AND B(A) X B(MPAC) ARE LESS THAN 1 IN MAGNITUDE, WHERE B, AS IS OBVIOUS,
# INDICATES THE ARRIVING CONTENTS.

DMPNSUB         TS      DMPNTEMP
                EXTEND
                MP      MPAC    +1
                DXCH    MPAC            # LOW PRODUCT TO MPAC, HIGH FACTOR TO A
                EXTEND
                MP      DMPNTEMP
                CA      L
                ADS     MPAC            # COMPLETING THE PRODUCT IN MPAC
                EXTEND
                DCA     MPAC            # BRINGING THE PRODUCT INTO A AND L
                TC      Q
```

```
# MISCELLANEOUS VECTOR OPERATIONS.  INCLUDED HERE ARE THE FOLLOWING.
#       1.      DOT     DP VECTOR DOT PRODUCT.
#       2.      VXV     DP VECTOR CROSS PRODUCT.
#       3.      VXSC    DP VECTOR TIMES SCALAR.
#       4.      V/SC    DP VECTOR DIVIDED BY SCALAR.
#       5.      VPROJ   DP VECTOR PROJECTION.  ( (MPAC.X)MPAC ).
#       6.      VXM     DP VECTOR POST-MULTIPLIED BY MATRIX.
#       7.      MXV     DP VECTOR PRE-MULTIPLIED BY MATRIX.

DOT             TC      PREDOT          # DO THE DOT PRODUCT AND EXIT, CHANGING
DMODE           CAF     ZERO            # THE MODE TO DP SCALAR.
                TCF     NEWMODE

MXV             CAF     TWO             # SET UP MATINC AND DOTINC FOR ROW
                TS      MATINC          # VECTORS.
                TCF     VXM/MXV         # GO TO COMMON PORTION.

VXM             CS      TEN             # SET MATINC AND DOTINC TO REFER TO MATRIX
                TS      MATINC          # AS THREE COLUMN VECTORS.
                CAF     SIX
```

```
# COMMON PORTION OF MXV AND VXM.

VXM/MXV          TS       DOTINC
# ITRACE (2) REFERS TO "VXM/MXV".
                 TC       MPACVBUF        # SAVE VECTOR IN MPAC FOR FURTHER USE.

                 TC       DOTSUB          # GO DOT TO GET X COMPONENT OF ANSWER.
                 EXTEND
                 DCA      VBUF            # MOVE MPAC VECTOR BACK INTO MPAC, SAVING
                 DXCH     MPAC            # NEW X COMPONENT IN BUF2.
                 DXCH     BUF2
                 EXTEND
                 DCA      VBUF    +2
                 DXCH     MPAC    +3
                 EXTEND
                 DCA      VBUF    +4
                 DXCH     MPAC    +5
                 CA       MATINC          # INITIALIZE ADDRWD FOR NEXT DOT PRODUCT.
                 ADS      ADDRWD          # FORMS HAS ADDRESS OF NEXT COLUMN(ROW).

                 TC       DOTSUB
                 DXCH     VBUF            # MORE GIVEN VECTOR BACK TO MPAC, SAVING Y
                 DXCH     MPAC            # COMPONENT OF ANSWER IN VBUF +2.
                 DXCH     VBUF    +2
                 DXCH     MPAC    +3
                 DXCH     VBUF    +4
                 DXCH     MPAC    +5
                 CA       MATINC          # FORM ADDRESS OF LAST COLUMN OR ROW.
                 ADS      ADDRWD

                 TC       DOTSUB
                 DXCH     BUF2            # ANSWER NOW COMPLETE. PUT COMPONENTS INTO
                 DXCH     MPAC            # PROPER MPAC REGISTERS.
                 DXCH     MPAC    +5
                 DXCH     VBUF    +2
                 DXCH     MPAC    +3
                 TCF      DANZIG          # EXIT.
```

```
# VXSC -- VECTOR TIMES SCALAR.

VXSC            CCS     MODE            # TEST PRESENT MODE.
                TCF     DVXSC           # SEPARATE ROUTINE WHEN SCALAR IS IN MPAC.
                TCF     DVXSC

VVXSC           TC      DMPSUB          # COMPUTE X COMPONENT
                TC      VROUND          # AND ROUND IT.
                DXCH    MPAC    +3      # PUT Y COMPONENT INTO MPAC SAVING MPAC IN
                DXCH    MPAC            # MPAC +3.
                DXCH    MPAC    +3

                TC      DMPSUB          # DO SAME FOR Y AND Z COMPONENTS.
                TC      VROUND
                DXCH    MPAC    +5
                DXCH    MPAC
                DXCH    MPAC    +5

                TC      DMPSUB
                TC      VROUND
VROTATEX        DXCH    MPAC            # EXIT USED TO RESTORE MPAC AFTER THIS
                DXCH    MPAC    +5      # TYPE OF ROTATION.  CALLED BY VECTOR SHIFT
                DXCH    MPAC    +3      # RIGHT, V/SC, ETC.
                DXCH    MPAC
                TCF     DANZIG
```

```
# DP VECTOR PROJECTION ROUTINE.

VPROJ           TC      PREDOT          # (MPAC.X)MPAC IS COMPUTED AND LEFT IN
                CS      FOUR            # MPAC.  DO DOT AND FALL INTO DVXSC.
                ADS     ADDRWD

# VXSC WHEN SCALAR ARRIVES IN MPAC AND VECTOR IS AT X.

DVXSC           EXTEND                  # SAVE SCALAR IN MPAC +3 AND GET X
                DCA     MPAC            # COMPONENT OF ANSWER.
                DXCH    MPAC    +3
                TC      DMPSUB
                TC      VROUND

                CAF     TWO             # ADVANCE ADDRWD TO Y COMPONENT OF X.
                ADS     ADDRWD
                EXTEND
                DCA     MPAC    +3      # PUT SCALAR BACK INTO MPAC AND SAVE
                DXCH    MPAC            # X RESULT IN MPAC +5.
                DXCH    MPAC    +5
                TC      DMPSUB
                TC      VROUND

                CAF     TWO
                ADS     ADDRWD          # TO Z COMPONENT.
                DXCH    MPAC    +3      # BRING SCALAR BACK, PUTTING Y RESULT IN
                DXCH    MPAC            # THE PROPER PLACE.
                DXCH    MPAC    +3
                TC      DMPSUB
                TC      VROUND

                DXCH    MPAC            # PUT Z COMPONENT IN PROPER PLACE, ALSO
                DXCH    MPAC    +5      # POSITIONING X.
                DXCH    MPAC

                TCF     VMODE           # MODE HAS CHANGED TO VECTOR.
```

```
# VECTOR CROSS PRODUCT ROUTINE CALCULATES (X M -X M ,X M -X M ,X M -X M ) WHERE M IS THE VECTOR IN
#                                           3 2  2 3  1 3  3 1  2 1  1 2
# MPAC AND X THE VECTOR AT THE GIVEN ADDRESS.

VXV             EXTEND
                DCA     MPAC    +5      # FORM UP M3X1, LEAVING M1 IN VBUF.
                DXCH    MPAC
                DXCH    VBUF
                TC      DMPSUB          # BY X1.

                EXTEND
                DCS     MPAC    +3      # CALCULATE -X1M2, SAVING X1M3 IN VBUF +2.
                DXCH    MPAC
                DXCH    VBUF    +2
                TC      DMPSUB

                CAF     TWO             # ADVANCE ADDRWD TO X2.
                ADS     ADDRWD
                EXTEND
                DCS     MPAC    +5      # PREPARE TO GET -X2M3, SAVING -X1M2 IN
                DXCH    MPAC            # MPAC +5.
                DXCH    MPAC    +5
                TC      DMPSUB

                EXTEND
                DCA     VBUF            # GET X2M1, SAVING -X2M3 IN VBUF +4.
                DXCH    MPAC
                DXCH    VBUF    +4
                TC      DMPSUB

                CAF     TWO             # ADVANCE ADDRWD TO X3.
                ADS     ADDRWD
                EXTEND
                DCS     VBUF            # GET -X3M1, ADDING X2M1 TO MPAC +5 TO
                DXCH    MPAC            # COMPLETE THE Z COMPONENT OF THE ANSWER.
                DAS     MPAC    +5

                EXTEND
                BZF     +2
                TC      OVERFLWZ

                TC      DMPSUB
                DXCH    VBUF    +2      # MOVE X1M3 TO MPAC +3 SETTING UP FOR X3M2
                DXCH    MPAC    +3      # AND ADD -X3M1 TO MPAC +3 TO COMPLETE THE
                DXCH    MPAC            # Y COMPONENT OF THE RESULT.
                DAS     MPAC    +3

                EXTEND
                BZF             +2
```

```
                TC OVERFLWY

                TC      DMPSUB
                DXCH    VBUF    +4      # GO ADD -X2M3 TO X3M2 TO COMPLETE THE X
                TCF     ENDVXV          # COMPONENT (TAIL END OF DAD).

# THE MPACVBUF SUBROUTINE SAVES THE VECTOR IN MPAC IN VBUF WITHOUT CLOBBERING MPAC.

MPACVBUF        EXTEND                  # CALLED BY MXV, VXM, AND UNIT.
                DCA     MPAC
                DXCH    VBUF
                EXTEND
                DCA     MPAC    +3
                DXCH    VBUF    +2
                EXTEND
                DCA     MPAC    +5
                DXCH    VBUF    +4
                TC      Q               # RETURN TO CALLER.

# DOUBLE PRECISION SIGN AGREE ROUTINE.  ARRIVE WITH INPUT IN A+L.  OUTPUT IS IN A + L.

ALSIGNAG        CCS     A               # TEST UPPER PART.
                TCF     UPPOS           # IT IS POSITIVE
                TC      Q               # ZERO
                TCF     UPNEG           # NEGATIVE
                TC      Q               # ZERO

UPPOS           XCH     L               # SAVE DECREMENTED UPPER PART.
                AD      HALF
                AD      HALF
                TS      A               # SKIPS ON OVERFLOW
                TCF     +2
                INCR    L               # RESTORE UPPER TO ORIGINAL VALUE
                XCH     L               # SWAP A + L BANCK.
                TC      Q

UPNEG           XCH     L               # SAVE COMPLEMENTED + DECREMENTED UPPER PT
                AD      NEGMAX
                AD      NEGONE
                TS      A
                TCF     +2              # DON'T INCREMENT IF NO OVERFLOW.
                INCR    L
                XCH     L
                COM                     # MAKE NEGATIVE AGAIN.
                TC      Q
```

```
# INTERPRETIVE INSTRUCTIONS WHOSE EXECUTION CONSISTS OF PRINCIPALLY CALLING SUBROUTINES.

DMP1            TC      DMPSUB          # DMP INSTRUCTIONS
                TCF     DANZIG

DMPR            TC      DMPSUB
                TC      ROUNDSUB +1     # (C(A) = +0).
                TCF     DANZIG

DDV             EXTEND
                INDEX   ADDRWD          # MOVE DIVIDEND INTO BUF.
                DCA     0
                TCF     BDDV    +4

BDDV            EXTEND                  # MOVE DIVISOR INTO MPAC SAVING MPAC, THE
                INDEX   ADDRWD          # DIVIDEND, IN BUF.
                DCA     0
                DXCH    MPAC
 +4             DXCH    BUF
                CAF     ZERO            # DIVIDE ROUTINES IN BANK 0.
                TS      FBANK
                TCF     DDV/BDDV

SETPD           CA      ADDRWD          # MUST SET TO WORK AREA, OR EBANK TROUBLE.
                TS      PUSHLOC
                TCF     NOIBNKSW        # NO FBANK SWITCH REQUIRED.

TSLC            CAF     ZERO            # SHIFTING ROUTINES LOCATED IN BANK 00.
                TS      FBANK
                TCF     TSLC2

GSHIFT          CAF     LOW7            # USED AS MASK AT GENSHIFT. THIS PROCESSES
                TS      FBANK           # ANY SHIFT INSTRUCTION (EXCEPT TSLC) WITH
                TCF     GENSHIFT        # AN ADDRESS (ROUTINES IN BANK 0).
```

```
# THE FOLLOWING IS THE PROLOGUE TO V/SC.  IF THE PRESENT MODE IS VECTOR, IT SAVES THE SCALAR AT X IN BUF
# AND CALLS THE V/SC ROUTINE IN BANK 0.  IF THE PRESENT MODE IS SCALAR, IT MOVES THE VECTOR AT X INTO MPAC, SAVING
# THE SCALAR IN MPAC IN BUF BEFORE CALLING THE V/SC ROUTINE IN BANK 0.

V/SC            CCS     MODE
                TCF     DV/SC           # MOVE VECTOR INTO MPAC.
                TCF     DV/SC

VV/SC           EXTEND
                INDEX   ADDRWD
                DCA     0
V/SC1           DXCH    BUF             # IN BOTH CASES, VECTOR IS NOW IN MPAC AND
                CAF     ZERO            # SCALAR IN BUF.
                TS      FBANK
                TCF     V/SC2

DV/SC           EXTEND
                INDEX   ADDRWD
                DCA     2
                DXCH    MPAC    +3
                EXTEND
                INDEX   ADDRWD
                DCA     4
                DXCH    MPAC    +5

                CS      ONE             # CHANGE MODE TO VECTOR.
                TS      MODE

                EXTEND
                INDEX   ADDRWD
                DCA     0
                DXCH    MPAC
                TCF     V/SC1           # FINISH PROLOGUE AT COMMON SECTION.
```

```
# SIGN AND COMPLEMENT INSTRUCTIONS.

SIGN            INDEX   ADDRWD          # CALL COMP INSTRUCTION IF WORD AT X IS
                CCS     0               # NEGATIVE NON-ZERO.
                TCF     DANZIG
                TCF     +2
                TCF     COMP            # DO THE COMPLEMENT.

                INDEX   ADDRWD
CCSL            CCS     1
                TCF     DANZIG
                TCF     DANZIG
                TCF     COMP
                TCF     DANZIG
COMP            EXTEND                  # COMPLEMENT DP MPAC IN EVERY CASE.
                DCS     MPAC
                DXCH    MPAC

                CCS     MODE            # EITHER COMPLEMENT MPAC +3 OR THE REST OF
                TCF     DCOMP           # THE VECTOR ACCUMULATOR.
                TCF     DCOMP

                EXTEND                  # VECTOR COMPLEMENT.
                DCS     MPAC    +3
                DXCH    MPAC    +3
                EXTEND
                DCS     MPAC    +5
                DXCH    MPAC    +5
                TCF     DANZIG

DCOMP           CS      MPAC    +2
                TS      MPAC    +2
                TCF     DANZIG
```

```
# THE FOLLOWING SHORT SHIFT CODES REQUIRE NO ADDRESS WORD:
#       1.      SR1 TO SR4      SCALAR SHIFT RIGHT.
#       2.      SR1R TO SR4R    SCALAR SHIFT RIGHT AND ROUND.
#       3.      SL1 TO SL4      SCALAR SHIFT LEFT.
#       4.      SL1R TO SL4R    SCALAR SHIFT LEFT AND ROUND.
#       5.      VSR1 TO VSR8    VECTOR SHIFT RIGHT (ALWAYS ROUNDS).
#       6.      VSL1 TO VSL8    VECTOR SHIFT LEFT (NEVER ROUNDS).
# THE FOLLOWING CODES REQUIRE AND ADDRESS WHICH MAY BE INDEXED:*
#       1.      SR              SCALAR SHIFT RIGHT.
#       2.      SRR             SCALAR SHIFT RIGHT AND ROUND.
#       3.      SL              SCALAR SHIFT LEFT.
#       4.      SLR             SCALAR SHIFT LEFT AND ROUND.
#       5.      VSR             VECTOR SHIFT RIGHT.
#       6.      VSL             VECTOR SHIFT LEFT.
# * IF THE ADDRESS IS INDEXED, AND THE INDEX MODIFICATION RESULTS IN A NEGATIVE SHIFT COUNT, A SHIFT OF THE
# ABSOLUTE VALUE OF THE COUNT IS DONE IN THE OPPOSITE DIRECTION.

                BANK    00

                COUNT*  $$/INTER
SHORTT          CAF     SIX             # SCALAR SHORT SHIFTS COME HERE.  THE SHIFT
                MASK    CYR             # COUNT-1 IS NOW IN BITS 2-3 OF CYR.  THE
                TS      SR              # ROUNDING BIT IS IN BIT1 AT THIS POINT.

                CCS     CYR             # SEE IF RIGHT OR LEFT SHIFT DESIRED.
                TCF     TSSL            # SHIFT LEFT.

SRDDV           DEC     20              # MPTEMP SETTING FOR SR BEFORE DDV.

TSSR            INDEX   SR              # GET SHIFTING BIT.
                CAF     BIT14
                TS      MPTEMP

                CCS     CYR             # SEE IF A ROUND IS DESIRED.
RIGHTR          TC      MPACSRND        # YES -- SHIFT RIGHT AND ROUND.
                TCF     NEWMODE         # SET MODE TO DP (C(A) = 0).
MPACSHR         CA      MPTEMP          # DO A TRIPLE PRECISION SHIFT RIGHT.
                EXTEND
                MP      MPAC    +2
 +3             TS      MPAC    +2      # (EXIT FROM SQRT AND ABVAL).
                CA      MPTEMP
                EXTEND
                MP      MPAC            # SHIFT MAJOR PART INTO A,L AND PLACE IN
```

```
                DXCH    MPAC            # MPAC,+1.
                CA      MPTEMP
                EXTEND
                MP      L               # ORIGINAL C(MPAC +1).
                DAS     MPAC    +1      # GUARANTEED NO OVERFLOW.
                TCF     DANZIG

# MPAC SHIFT RIGHT AND ROUND SUBROUTINES

MPACSRND        CA      MPAC    +2      # WE HAVE TO DO ALL THREE MULTIPLIES SINCE
                EXTEND                  # MPAC +1 AND MPAC +2 MIGHT HAVE SIGN
                MP      MPTEMP          # DISAGREEMENT WITH A SHIFT RIGHT OF L.
                XCH     MPAC    +1
                EXTEND
                MP      MPTEMP
                XCH     MPAC    +1      # TRIAL MINOR PART.
                AD      L

VSHR2           DOUBLE                  # (FINISH VECTOR COMPONENT SHIFT RIGHT
                TS      MPAC    +2      # AND ROUND.)
                TCF     +2
                ADS     MPAC    +1      # GUARANTEED NO OVERFLOW.

                CAF     ZERO
                TS      MPAC    +2
                XCH     MPAC            # SETTING TO ZERO SO FOLLOWING DAS WORKS.
                EXTEND
                MP      MPTEMP
                DAS     MPAC            # AGAIN NO OVERFLOW.
                TC      Q

VSHRRND         CA      MPTEMP          # ENTRY TO SHIFT RIGHT AND ROUND MPAC WHEN
                EXTEND                  # MPAC CONTAINS A VECTOR COMPONENT.
                MP      MPAC    +1
                TS      MPAC    +1
                XCH     L
                TCF     VSHR2           # GO ADD ONE IF NECESSARY AND FINISH.
```

```
# ROUTINE FOR SHORT SCALAR SHIFT LEFT (AND MAYBE ROUND).

TSSL            CA      SR              # GET SHIFT COUNT FOR SR.
 +1             TS      MPTEMP

 +2             EXTEND                  # ENTRY HERE FROM SL FOR SCALARS.
                DCA     MPAC    +1      # SHIFTING LEFT ONE PLACE AT A TIME IS
                DAS     MPAC    +1      # FASTER THAN DOING THE WHOLE SHIFT WITH
                AD      MPAC            # MULTIPLIES ASSUMING THAT FREQUENCY OF
                AD      MPAC            # SHIFT COUNTS GOES DOWN RAPIDLY AS A
                TS      MPAC            # FUNCTION OF THEIR MAGNITUDE.
                TCF     +2
                TS      OVFIND          # OVERFLOW.  (LEAVES OVERFLOW-CORRECTED
                                        # RESULT ANYWAY).
                CCS     MPTEMP          # LOOP ON DECREMENTED SHIFT COUNT.
                TCF     TSSL    +1

                CCS     CYR             # SEE IF ROUND WANTED.
ROUND           TC      ROUNDSUB        # YES -- ROUND AND EXIT.
                TCF     DANZIG          # SL LEAVES A ZERO IN CYR FOR NO ROUND.
                TCF     DANZIG          # NO -- EXIT IMMEDIATELY
```

```
# VECTOR SHIFTING ROUTINES.

SHORTV          CAF     LOW3            # SAVE 3 BIT SHIFT COUNT -- 1 WITHOUT
                MASK    CYR             # EDITING CYR.
                TS      MPTEMP
                CCS     CYR             # SEE IF LEFT OR RIGHT SHIFT.
                TCF     VSSL            # VECTOR SHIFT LEFT.
OCT176          OCT     176             # USED IN PROCESSED SHIFTS WITH - COUNT.

VSSR            INDEX   MPTEMP          # (ENTRY FROM SR).  PICK UP SHIFTING BIT.
                CAF     BIT14           # MPTEMP CONTAINS THE SHIFT COUNT - 1.
                TS      MPTEMP
                TC      VSHRRND         # SHIFT X COMPONENT.

                DXCH    MPAC            # SWAP X AND Y COMPONENTS.
                DXCH    MPAC    +3
                DXCH    MPAC
                TC      VSHRRND         # SHIFT Y COMPONENT.

                DXCH    MPAC            # SWAP Y AND Z COMPONENTS.
                DXCH    MPAC    +5
                DXCH    MPAC
                TC      VSHRRND         # SHIFT Z COMPONENT.

                TCF     VROTATEX        # RESTORE COMPONENTS TO PROPER PLACES.
```

```
# VECTOR SHIFT LEFT -- DONE ONE PLACE AT A TIME.

 -1             TS      MPTEMP          # SHIFTING LOOP.

VSSL            EXTEND
                DCA     MPAC
                DAS     MPAC
                EXTEND
                BZF     +2
                TC      OVERFLOW

                EXTEND
                DCA     MPAC    +3
                DAS     MPAC    +3
                EXTEND
                BZF     +2
                TC      OVERFLWY

                EXTEND
                DCA     MPAC    +5
                DAS     MPAC    +5
                EXTEND
                BZF     +2
                TC      OVERFLWZ

                CCS     MPTEMP          # LOOP ON DECREMENTED SHIFT COUNTER.
                TCF     VSSL    -1
                TCF     DANZIG          # EXIT.
```

```
# TSLC -- TRIPLE SHIFT LEFT AND COUNT.  SHIFTS MPAC LEFT UNTIL GREATER THAN .5 IN MAGNITUDE, LEAVING
# THE COMPLEMENT OF THE NUMBER OF SHIFTS REQUIRED IN X.

TSLC2           TS      MPTEMP          # START BY ZEROING SHIFT COUNT (IN A NOW).
                TC      BRANCH          # EXIT WITH NO SHIFTING IF ARGUMENT ZERO.
                TCF     +2
                TCF     ENDTSLC         # STORES ZERO SHIFT COUNT IN THIS CASE.

                TC      TPAGREE         # MAY CAUSE UPSHIFT OF ONE EXTRA PLACE.

                CA      MPAC            # BEGIN NORMALIZATION LOOP.
                TCF     TSLCTEST

TSLCLOOP        INCR    MPTEMP          # INCREMENT SHIFT COUNTER.
                EXTEND
                DCA     MPAC    +1
                DAS     MPAC    +1
                AD      MPAC
                ADS     MPAC
TSLCTEST        DOUBLE                  # SEE IF (ANOTHER) SHIFT IS REQUIRED
                OVSK
                TCF     TSLCLOOP        # YES -- INCREMENT COUNT AND SHIFT AGAIN.

ENDTSLC         CS      MPTEMP
                TCF     STORE1          # STORE SHIFT COUNT AND RETURN TO DANZIG.
```

```
# THE FOLLOWING ROUTINE PROCESSES THE GENERAL SHIFT INSTRUCTIONS SR, SRR, SL, AND SLR.
# THE GIVEN ADDRESS IS DECODED AS FOLLOWS:
#       BITS 1-7        SHIFT COUNT (SUBADDRESS) LESS THAN 125 DECIMAL.
#       BIT 8           PSEUDO SIGN BIT (DETECTS CHANGE IN SIGN IN INDEXED SHIFTS).
#       BIT 9           0 FOR LEFT SHIFT, AND 1 FOR RIGHT SHIFT.
#       BIT 10          1 FOR TERMINAL ROUND ON SCALAR SHIFTS, 0 OTHERWISE
#       BITS 11-13      0.
#       BIT 14          1.
#       BIT 15          0.
# THE ABOVE ENCODING IS DONE BY THE YUL SYSTEM.

GENSHIFT        MASK    ADDRWD          # GET SHIFT COUNT, TESTING FOR ZERO.
                CCS     A               # (ARRIVES WITH C(A) = LOW7).
                TCF     GENSHFT2        # IF NON-ZERO, PROCEED WITH DECREMENTED CT

                CAF     BIT10           # ZERO SHIFT COUNT.  NO SHIFTS NEEDED BUT
                MASK    ADDRWD          # WE MIGHT HAVE TO ROUND MPAC ON SLR AND
                CCS     A               # SRR (SCALAR ONLY).
                TC      ROUNDSUB
                TCF     DANZIG

GENSHFT2        TS      MPTEMP          # DECREMENTED SHIFT COUNT TO MPTEMP.
                CAF     BIT8            # TEST MEANING OF LOW SEVEN BIT COUNT IN
                EXTEND                  # MPTEMP NOW.
                MP      ADDRWD
                MASK    LOW2            # JUMPS ON SHIFT DIRECTION (BIT8) AND
                INDEX   A
                TCF     +1              # ORIGINAL SHIFT DIRECTION (BIT 9)
                TCF     RIGHT-          # NEGATIVE SHIFT COUNT FOR SL OR SLR.
                TCF     LEFT            # SL OR SLR.
                TCF     LEFT-           # NEGATIVE SHIFT COUNT WITH SR OR SRR.
```

```
# GENERAL SHIFT RIGHT

RIGHT           CCS     MODE            # SET IF VECTOR OR SCALAR.
                TCF     GENSCR
                TCF     GENSCR

                CA      MPTEMP          # SEE IF SHIFT COUNT LESS THAN 14D.
VRIGHT2         AD      NEG12
                EXTEND
                BZMF    VSSR            # IF SO, BRANCH AND SHIFT IMMEDIATELY.

                AD      NEGONE          # IF NOT, REDUCE MPTEMP BY A TOTAL OF 14.
                TS      MPTEMP          # AND DO A SHIFT RIGHT AND ROUND BY 14.
                CAF     ZERO            # THE ROUND AT THIS STAGE MAY INTRODUCE A
                TS      L               # ONE BIT ERROR IN A SHIFT RIGHT 15D.
                XCH     MPAC
                XCH     MPAC    +1
                TC      SETROUND        # X COMPONENT NOW SHIFTED, SO MAKE UP THE
                DAS     MPAC            # ROUNDING QUANTITY (0 IN A AND 0 OR +-1
                                        # IN L).
                XCH     MPAC    +3      # REPEAT THE ABOVE PROCESS FOR Y AND Z.
                XCH     MPAC    +4
                TC      SETROUND
                DAS     MPAC    +3      # NO OVERFLOW ON THESE ADDS.

                XCH     MPAC    +5
                XCH     MPAC    +6
                TC      SETROUND
                DAS     MPAC    +5

                CCS     MPTEMP          # SEE IF DONE, DOING FINAL DECREMENT.
                TS      MPTEMP
                TCF     VRIGHT2
BIASLO          DEC     .2974   B-1     # SQRT CONSTANT

                TCF     DANZIG

SETROUND        DOUBLE                  # MAKES UP ROUNDING QUANTITY FROM ARRIVING
                TS      MPAC    +2      # C(A).   L IS ZERO INITIALLY.
                CAF     ZERO
                XCH     L
                TC      Q               # RETURN AND DO THE DAS, RESETTING L TO 0.
```

```
# PROCESS SR AND SRR FOR SCALARS.

GENSCR          CA      MPTEMP          # SEE IF THE ORIGINAL SHIFT COUNT WAS LESS
 +1             AD      NEG12           # THAN 14D.
                EXTEND
                BZMF    DOSSHFT         # DO THE SHIFT IMMEDIATELY IF SO.

 +4             AD      NEGONE          # IF NOT, DECREMENT SHIFT COUNT BY 14D AND
                TS      MPTEMP          # SHIFT MPAC RIGHT 14 PLACES.
                CAF     ZERO
                XCH     MPAC
                XCH     MPAC    +1
                TS      MPAC    +2
                CCS     MPTEMP          # SEE IF FINISHED, DO FINAL DECREMENT.
                TS      MPTEMP
                TC      GENSCR  +1
SLOPEHI         DEC     .5884           # SQRT CONSTANT.
                CAF     BIT10           # FINISHED WITH SHIFT.  SEE IF ROUND
                MASK    ADDRWD          # WANTED.
                CCS     A
                TC      ROUNDSUB
                TCF     DANZIG          # DO SO AND/OR EXIT.

DOSSHFT         INDEX   MPTEMP          # PICK UP SHIFTING BIT.
                CAF     BIT14
                TS      MPTEMP
                CAF     BIT10           # SEE IF TERMINAL ROUND DESIRED.
                MASK    ADDRWD
                CCS     A
                TCF     RIGHTR          # YES.
                TCF     MPACSHR         # JUST SHIFT RIGHT.
```

```
# PROCESS THE RIGHT- (SL(R) WITH A NEGATIVE COUNT), LEFT-, AND LEFT OPTIONS.

RIGHT-          CS      MPTEMP          # GET ABSOLUTE VALUE - 1 OF SHIFT COUNT
                AD      OCT176          # UNDERSTANDING THAT BIT8 (PSEUDO-SIGN)
                TS      MPTEMP          # WAS 1 INITIALLY.
                TCF     RIGHT           # DO NORMAL SHIFT RIGHT.

LEFT-           CS      OCT176          # SAME PROLOGUE TO LEFT FOR INDEXED RIGHT
                AD      MPTEMP          # SHIFT WHOSE NET SHIFT COUNT IS NEGATIVE
                COM
                TS      MPTEMP

LEFT            CCS     MODE            # SINCE LEFT SHIFTING IS DONE ONE PLACE AT
                TCF     GENSCL          # A TIME, NO COMPARISON WITH 14 NEED BE
                TCF     GENSCL          # DONE.  FOR SCALARS, SEE IF TERMINAL ROUND
                TCF     VSSL            # DESIRED.  FOR VECTORS, SHIFT IMMEDIATELY.

GENSCL          CS      ADDRWD          # PUT ROUNDING BIT (BIT 10 OF ADDRWD) INTO
                EXTEND                  # BIT 15 OF CYR WHERE THE ROUNDING BIT OF
                MP      BIT6            # A SHORT SHIFT LEFT WOULD BE
                TS      CYR
                TCF     TSSL    +2      # DO THE SHIFT.
```

```
# SCALAR DIVISION INSTRUCTIONS, DDV AND BDDV, ARE EXECUTED HERE.  AT THIS POINT, THE DIVIDEND IS IN MPAC
# AND THE DIVISOR IS IN BUF.

DDV/BDDV        CS      ONE             # INITIALIZATION
                TS      DVSIGN          # +-1 FOR POSITIVE QUOTIENT -- -0 FOR NEG.
                TS      DVNORMCT        # DIVIDENT NORMALIZATION COUNT.
                TS      MAXDVSW         # NEAR-ONE DIVIDE FLAG.

                CCS     BUF             # FORCE BUF POSITIVE WITH THE MAJOR PART
                TCF     BUFPOS          # NON-ZERO.
                TCF     +2
                TCF     BUFNEG

BUFZERO         TS      MPAC    +2      # ZERO THIS.
                TC      TPAGREE         # FORCE SIGN AGREEMENT BEFORE OVERFLOW

                CCS     MPAC            # TEST TO SEE IF MPAC NON-ZERO.  (TOO BIG)
                TCF     OVF+            # MAJOR PART OF DIVIDEND IS POSITIVE NON-0
                TCF     +2
                TCF     OVF+    -1      # MAJOR PART OF DIVIDEND IS NEG. NON-ZERO

                XCH     BUF     +1      # SHIFT DIVIDEND AND DIVISOR LEFT 14
                XCH     BUF
                XCH     MPAC    +1
                XCH     MPAC
                CCS     BUF             # TRY AGAIN ON FORMER MINOR PART.
                TCF     BUF+
                TCF     +2              # OVERFLOW ON ZERO DIVISOR.
                TCF     BUF-

                CS      MPAC            # SIGN OF MPAC DETERMINES SIGN OF RESULT.
SGNDVOVF        EXTEND
                BZMF    +2
                INCR    DVSIGN          # NEGMAX IN MPAC PERHAPS.
DVOVF           CAF     POSMAX          # ON DIVISION OVERFLOW OF ANY SORT, SET
                TS      MPAC            # SET DP MPAC TO +-POSMAX.
                TC      FINALDV +3
                CAF     ONE             # SET OVERFLOW INDICATOR AND EXIT.
                TS      OVFIND
                TC      DANZIG

 -1             INCR    DVSIGN
OVF+            CS      BUF     +1      # LOAD LOWER ORDER PART OF DIVISOR.
                TCF     SGNDVOVF        # GET SIGN OF RESULT.

BUF-            EXTEND                  # IF BUF IS NEGATIVE, COMPLEMENT IT AND
                DCS     BUF             # MAINTAIN DVSIGN FOR FINAL QUOTIENT SIGN.
                DXCH    BUF
                INCR    DVSIGN          # NOW -0.
```

```
BUF+            CCS     MPAC            # FORCE MPAC POSITIVE, CHECKING FOR ZERO
                TCF     MPAC+           # DIVIDEND IN THE PROCESS.
                TCF     +2
                TCF     MPAC-
                CCS     MPAC    +1
                TCF     MPAC+
                TCF     DANZIG          # EXIT IMMEDIATELY ON ZERO DIVIDEND.
                TCF     MPAC-
                TCF     DANZIG

MPAC-           EXTEND                  # FORCE MPAC POSITIVE AS BUF IN BUF-.
                DCS     MPAC
                DXCH    MPAC
                INCR    DVSIGN          # NOW +1 OR -0.
```

```
MPAC+           CS      MPAC            # CHECK FOR DIVISION OVERFLOW.  IF THE
                AD      NEGONE          # MAJOR PART OF THE DIVIDEND IS LESS THAN
                AD      BUF             # THE MAJOR PART OF THE DIVISOR BY AT
                CCS     A               # LEAST TWO, WE CAN PROCEED IMMEDIATELY
                TCF     DVNORM          # WITHOUT NORMALIZATION PRODUCING A DVMAX.
-1/2+2          OCT     60001           # USED IN SQRTSUB.

                TCF     +1              # IF THE ABOVE DOES NOT HOLD, FORCE SIGN
                CAF     HALF            # AGREEMENT IN NUMERATOR AND DENOMINATOR
                DOUBLE                  # TO FACILITATE OVERFLOW AND NEAR-ONE
                AD      MPAC    +1      # CHECKING.
                TS      MPAC    +1
                CAF     ZERO
                AD      POSMAX
                ADS     MPAC

                CAF     HALF            # SAME FOR BUF.
                DOUBLE
                AD      BUF     +1
                TS      BUF     +1
                CAF     ZERO
                AD      POSMAX
                ADS     BUF

                CS      MPAC            # CHECK MAGNITUDE OF SIGN-CORRECTED
                AD      BUF             # OPERANDS.
                CCS     A
                TCF     DVNORM          # DIVIDE OK -- WILL NOT BECOME MAXOV CASE.
LBUF2           ADRES   BUF2
                TCF     DVOVF           # DIVISOR NOT LESS THAN DIVIDEND -- OVF.

                TS      MAXDVSW         # IF THE MAJOR PARTS OF THE DIVIDEND AND
                CS      MPAC    +1      # DIVISOR ARE EQUAL, A SPECIAL APPROXIMA-
                AD      BUF     +1      # TION IS USED (PROVIDED THE DIVISION IS
                EXTEND                  # POSSIBLE, OF COURSE).
                BZMF    DVOVF
                TCF     DVNORM          # IF NO OVERFLOW.
```

```
BUFNORM         EXTEND                          # ADD -1 TO AUGMENT SHIFT COUNT AND SHIFT
                AUG     DVNORMCT                # LEFT ONE PLACE.
                EXTEND
                DCA     BUF
                DAS     BUF

DVNORM          CA      BUF                     # SEE IF DIVISOR NORMALIZED YET.
                DOUBLE
                OVSK
                TCF     BUFNORM                 # NO -- SHIFT LEFT ONE AND TRY AGAIN.

                DXCH    MPAC                    # CALL DIVIDEND NORMALIZATION SEQUENCE
                INDEX   DVNORMCT                # PRIOR TO DOING THE DIVIDE.
                TC      MAXTEST

                TS      MPAC    +2              # RETURNS WITH DIVISION DONE AND C(A) = 0.
                TCF     DANZIG

BUFPOS          CCS     A
                TCF     BUF+                    # TO BUF+ IF BUF IS GREATER THAN +1.

                CS      BUF     +1              # IF BUF IS +1, FORCING SIGN AGREEMENT
                EXTEND                          # MAY CAUSE BUF TO BECOME ZERO.
                BZMF    BUF+                    # BRANCH IF SIGNS AGREE.

                CA      HALF                    # SIGNS DISAGREE.  FORCE AGREEMENT.
 +6             DOUBLE
                ADS     BUF     +1
                CA      ZERO
                TS      BUF
                TCF     BUFZERO

BUFNEG          CCS     A
                TCF     BUF-                    # TO BUF- IF BUF IS LESS THAN -1.

                CA      BUF     +1              # IF BUF IS -1, FORCING SIGN AGREEMENT
                EXTEND                          # MAY CAUSE BUF TO BECOME ZERO.
                BZMF    BUF-                    # BRANCH IF SIGNS AGREE.

                CS      HALF                    # SIGNS DISAGREE.  FORCE AGREEMENT.
                TCF     BUFPOS +6
```

```
# THE FOLLOWING ARE PROLOGUES TO SHIFT THE DIVIDEND ARRIVING IN A AND L BEFORE THE DIVIDE.

 -21D           LXCH    SR              # SPECIAL PROLOGUE FOR UNIT WHEN THE
                EXTEND                  # LENGTH OF THE ARGUMENT WAS NOT LESS THAN
                MP      HALF            # .5.  IN THIS CASE, EACH COMPONENT MUST BE
                XCH     L               # SHIFTED RIGHT ONE TO PRODUCE A HALF-UNIT
                AD      SR              # VECTOR.
                XCH     L
                TCF     GENDDV  +1      # WITH DP DIVIDEND IN A,L.

                DDOUBL                  # PROLOGUE WHICH NORMALIZES THE DIVIDEND
                DDOUBL                  # WHEN IT IS KNOWN THAT NO DIVISION
                DDOUBL                  # OVEFLOW WILL OCCUR.
                DDOUBL
                DDOUBL
                DDOUBL
                DDOUBL
                DDOUBL
                DDOUBL
                DDOUBL
                DDOUBL
                DDOUBL
                DDOUBL
                DXCH    MPAC

MAXTEST         CCS     MAXDVSW         # 0 IF MAJORS MIGHT BE =, -1 OTHERWISE.
BIASHI          DEC     .4192   B-1     # SQRT CONSTANTS.

                TCF     MAXDV           # CHECK TO SEE IF THAY ARE NOW EQUAL.
```

```
# THE FOLLOWING IS A GENERAL PURPOSE DOUBLE PRECISION DIVISION ROUTINE.  IT DIVIDES MPAC BY BUF AND LEAVES
# THE RESULT IN MPAC.  THE FOLLOWING CONDITIONS MUST BE SATISFIED:
#
#       1.      THE DIVISOR (BUF) MUST BE POSITIVE AND NOT LESS THAN .5.
#
#       2.      THE DIVIDEND (MPAC) MUST BE POSITIVE WITH THE MAJOR PART OF MPAC STRICTLY LESS THAN THAT OF BUF
#               (A SPECIAL APPROXIMATION, MAXDV, IS USED WHEN THE MAJOR PARTS ARE EQUAL).
#
# UNDERSTANDING THAT A/B = Q + S(R/B) WHERE S = 2(-14) AND Q AND R ARE QUOTIENT AND REMAINDER, RESPEC-
# TIVELY, THE FOLLOWING APPROXIMATION IS OBTAINED BY MULTIPLYING ABOVE AND BELOW BY C - SD AND NEGLECTING TERMS OF
# ORDER S-SQUARED (POSSIBLY INTRODUCING ERROR INTO THE LOW TWO BITS OF THE RESULT).  SIGN AGREEMENT IS UNNECESSARY.
#
#       A + SB .      (R - QD)                                                  A + SB
#       ------ = Q + S(------) WHERE Q AND R ARE QUOTIENT AND REMAINDER OF ------ RESPECTIVELY.
#       C + SD         (  C   }                                                    C

GENDDV          DXCH    MPAC            # WE NEED A AND B ONLY FOR FIRST DV.
 +1             EXTEND                  # (SPECIAL UNIT PROLOGUE ENTERS HERE).
                DV      BUF             # A NOW CONTAINS Q AND L, R.
                DXCH    MPAC

                CS      MPAC            # FORM DIVIDEND FOR MINOR PART OF RESULT.
                EXTEND
                MP      BUF     +1
                AD      MPAC    +1      # OVERFLOW AT THIS POINT IS POSITIVE SINCE
                OVSK                    # R IS POSITIVE IN EVERY CASE.
                TCF     +5

                EXTEND                  # OVERFLOW CAN BE REMOVED BY SUBTRACTING C
                SU      BUF             # (BUF) ONCE SINCE R IS ALWAYS LESS THAN C
                INCR    MPAC            # IN THIS CASE.  INCR COMPENSATES SUBTRACT.
                TCF     +DOWN           # (SINCE C(A) IS STILL POSITIVE).

 +5             EXTEND                  # C(A) CAN BE MADE LESS THAN C IN MAGNI-
                BZMF    -UP             # TUDE BY DIMINISHING IT BY C (SINCE C IS
                                        # NOT LESS THAN .5) UNLESS C(A) = 0.
```

```
+DOWN           EXTEND
                SU      BUF             # IF POSITIVE, REDUCE ONLY IF NECESSARY
                EXTEND                  # SINCE THE COMPENSATING INCR MIGHT CAUSE
                BZF     +3              # OVERFLOW.
                EXTEND                  # DON'T SUBTRACT UNLESS RESULT IS POSITIVE
                BZMF    ENDMAXDV        # OR ZERO.

 +3             INCR    MPAC            # KEEP SUBTRACT HERE AND COMPENSATE.
                TCF     FINALDV

-UP             EXTEND                  # IF ZERO, SET MINOR PART OF RESULT TO
                BZF     FINALDV +3      # ZERO.

                EXTEND                  # IF NEGATIVE, ADD C TO A, SUBTRACTING ONE
                DIM     MPAC            # TO COMPENSATE.  DIM IS OK HERE SINCE THE
ENDMAXDV        AD      BUF             # MAJOR PART NEVER GOES NEGATIVE.
```

```
FINALDV         ZL                              # DO DV TO OBTAIN MINOR PART OF RESULT.
                EXTEND
                DV      BUF
 +3             TS      MPAC    +1

                CCS     DVSIGN                  # LEAVE RESULT POSITIVE UNLESS C(DVSIGN)=
                TC      Q                       # -0.
                TC      Q
                TC      Q

                EXTEND
                DCS     MPAC
                DXCH    MPAC
                CAF     ZERO                    # SO WE ALWAYS RETURN WITH C(A) = 0.
                TC      Q
```

```
# IF THE MAJOR PARTS OF THE DIVISOR AND DIVIDEND ARE EQUAL, BUT THE MINOR PARTS ARE SUCH THAT THE
# DIVIDEND IS STRICTLY LESS THAN THE DIVISOR IN MAGNITUDE, THE FOLLOWING APPROXIMATION IS USED.  THE ASSUMPTIONS
# ARE THE SAME AS THE GENERAL ROUTINE WITH THE ADDITION THAT SIGN AGREEMENT IS NECESSARY (B, C, & D POSITIVE).
#
#       C + SB .          (C + B - D)
#       ------ = 37777 + S(---------)
#       C + SD            (    C    )
#
# THE DIVISION MAY BE PERFORMED IMMEDIATELY SINCE B IS STRICTLY LESS THAN D AND C IS NOT LESS THAN .5.

MAXDV           CS      MPAC            # SEE IF MAXDV CASE STILL HOLDS AFTER
                AD      BUF             # NORMALIZATION.
                EXTEND
                BZF     +2
                TCF     GENDDV          # MPAC NOW LESS THAN BUFF -- DIVIDE AS USUAL.

 +2             CAF     POSMAX          # SET MAJOR PART OF RESULT.
                TS      MPAC

                CS      BUF     +1      # FORM DIVIDEND OF MINOR PART OF RESULT.
                AD      MPAC    +1
                TCF     ENDMAXDV        # GO ADD C AND DO DIVIDE, ATTACHING SIGN
                                        # BEFORE EXITING.
```

```
# VECTOR DIVIDED BY SCALAR, V/SC, IS EXECUTED HERE.  THE VECTOR IS NOW IN MPAC WITH SCALAR IN BUF.

V/SC2           CS      ONE             # INITIALIZE DIVIDEND NORMALIZATION COUNT
                TS      DVNORMCT        # AND DIVISION SIGN REGISTER.
                TS      VBUF    +5

                TC      VECAGREE        # FORCE SIGN AGREEMENT IN VECTOR

                DXCH    BUF
                TC      ALSIGNAG        # SIGN AGREE BUF
                DXCH    BUF
                CCS     BUF             # FORCE DIVISOR POSITIVE WITH MAJOR PART
                TCF     /BUF+           # NON-ZERO (IF POSSIBLE).
                TCF     +2
                TCF     /BUF-

                XCH     BUF     +1      # SHIFT VECTOR AND SCALAR LEFT 14.
                XCH     BUF
                XCH     MPAC    +1
                XCH     MPAC
                EXTEND                  # CHECK FOR OVERFLOW IN EACH CASE.
                BZF     +2
                TCF     DVOVF

                XCH     MPAC    +4
                XCH     MPAC    +3
                EXTEND
                BZF     +2
                TCF     DVOVF

                XCH     MPAC    +6
                XCH     MPAC    +5
                EXTEND
                BZF     +2
                TCF     DVOVF

                CCS     BUF
                TCF     /BUF+
                TCF     DVOVF           # ZERO DIVISOR - OVERFLOW.
                TCF     /BUF-
                TCF     DVOVF

/BUF-           EXTEND                  # ON NEGATIVE, COMPLEMENT BUF AND MAINTAIN
                DCS     BUF             # DVSIGN IN VBUF +5.
                DXCH    BUF
                INCR    VBUF    +5
```

```
/BUF+           EXTEND
                DCA     BUF             # LEAVE ABS(ORIG DIVISOR) IN BUF2
                DXCH    BUF2            # FOR OVERFLOW TESTING
                TCF     /NORM           # NORMALIZE DIVISOR IN BUF.

/NORM2          EXTEND                  # IF LESS THAN .5, AUGMENT DVNORMCT AND
                AUG     DVNORMCT        # DOUBLE DIVISOR.
                EXTEND
                DCA     BUF
                DAS     BUF

/NORM           CA      BUF             # SEE IF DIVISOR NORMALIZED.
                DOUBLE
                OVSK
                TCF     /NORM2          # DOUBLE AND TRY AGAIN IF NOT.

                TC      V/SCDV          # DO X COMPONENT DIVIDE.
                DXCH    MPAC    +3      # SUPPLY ARGUMENTS IN USUAL SEQUENCE.
                DXCH    MPAC
                DXCH    MPAC    +3

                TC      V/SCDV          # Y COMPONENT.
                DXCH    MPAC    +5
                DXCH    MPAC
                DXCH    MPAC    +5

                TC      V/SCDV          # Z COMPONENT.
                TCF     VROTATEX        # GO RE-ARRANGE COMPONENTS BEFORE EXIT.
```

```
# SUBROUTINE USED BY V/SC TO DIVIDE VECTOR COMPONENT IN MPAC,+1 BY THE SCALAR GIVEN IN BUF.

V/SCDV          CA      VBUF    +5      # REFLECTS SIGN OF SCALAR.
                TS      DVSIGN

                CCS     MPAC            # FORCE MPAC POSITIVE, EXITING ON ZERO.
                TCF     /MPAC+
                TCF     +2
                TCF     /MPAC-

                CCS     MPAC    +1
                TCF     /MPAC+
                TC      Q
                TCF     /MPAC-
                TC      Q

/MPAC-          EXTEND                  # USUAL COMPLEMENTING AND SETTING OF SIGN.
                DCS     MPAC
                DXCH    MPAC
                INCR    DVSIGN

/MPAC+          CS      ONE             # INITIALIZE NEAR-ONE SWITCH.
                TS      MAXDVSW

                CS      MPAC            # CHECK POSSIBLE OVERFLOW.
                AD      BUF2            # UNNORMALIZED INPUT DIVISOR.
                CCS     A
                TCF     DDVCALL         # NOT NEAR-ONE
                TCF     +2              # +0 IS JUST POSSIBLE
                TCF     DVOVF           # NO HOPE
                TS      MAXDVSW         # SIGNAL POSSIBLE NEAR-ONE CASE
                CS      MPAC    +1      # SEE IF DIVISION CAN BE DONE
                AD      BUF2    +1
                EXTEND
                BZMF    DVOVF

DDVCALL         DXCH    MPAC            # CALL PRE-DIVIDE NORMALIZATION.
                INDEX   DVNORMCT
                TCF     MAXTEST
```

```
SLOPELO         DEC     .8324

VECAGREE        XCH     Q               # SAVE Q IN A
                DXCH    MPAC
                TC      ALSIGNAG        # SIGNAGREE MPAC
                DXCH    MPAC
                DXCH    MPAC    +3
                TC      ALSIGNAG        # SIGN AGREE MPAC +3
                DXCH    MPAC    +3
                DXCH    MPAC    +5
                TC      ALSIGNAG        # SIGNAGREE MPAC +5
                DXCH    MPAC    +5
                TC      A
```

```
# THE FOLLOWING ROUTINE EXECUTES THE UNIT INSTRUCTION, WHICH TAKES THE UNIT OF THE VECTOR IN MPAC.

UNIT            TC      VECAGREE        # FORCE SIGN AGREEMENT IN VECTOR
                TC      MPACVBUF        # SAVE ARGUMENT IN VBUF
                CAF     ZERO            # MUST SENSE OVERFLOW IN FOLLOWING DOT.
                XCH     OVFIND
                TS      TEM1
                TC      VSQSUB          # DOT MPAC WITH ITSELF.
                CA      TEM1
                XCH     OVFIND
                EXTEND
                BZF     +2
                TCF     DVOVF
                EXTEND
                DCA     MPAC            # LEAVE THE SQUARE OF THE LENGTH OF THE
                INDEX   FIXLOC          # ARGUMENT IN LVSQUARE.
                DXCH    LVSQUARE

                TC      SQRTSUB         # GO TAKE THE NORMALIZED SQUARE ROOT.

                CCS     MPAC            # CHECK FOR UNIT OVERFLOW.
                TCF     +5              # MPAC IS NOT LESS THAN .5 UNLESS
                TS      L
                INDEX   FIXLOC
                DXCH    LV
                TCF     DVOVF           # INPUT TO SQRTSUB WAS 0.

                CS      FOURTEEN        # SEE IF THE INPUT WAS SO SMALL THAT THE
                AD      MPTEMP          # FIRST TWO REGISTERS OF THE SQUARE WERE 0
                CCS     A
                COM                     # IF SO, SAVE THE NEGATIVE OF THE SHIFT
                TCF     SMALL           # COUNT -15D.

                TCF     LARGE           # (THIS IS USUALLY THE CASE.)

                CS      THIRTEEN        # IF THE SHIFT COUNT WAS EXACTLY 14, SET
                TS      MPTEMP          # THE PRE-DIVIDE NORM COUNT TO -13D.

                CA      MPAC            # SHIFT THE LENGTH RIGHT 14 BEFORE STORING
SMALL2          TS      L               # (SMALL EXITS TO THIS POINT).
                CAF     ZERO
                TCF     LARGE2          # GO TO STORE LENGTH AND PROCEED.

LARGE           CCS     MPTEMP          # MOST ALL CASES COME HERE.
                TCF     LARGE3          # SEE IF NO NORMALIZATION WAS REQUIRED BY

                CS      SRDDV           # SQRT, AND IF SO, SET UP FOR A SHIFT
                TS      MPTEMP          # RIGHT 1 BEFORE DIVIDING TO PRODUCE
                EXTEND                  # THE DESIRED HALF UNIT VECTOR.
                DCA     MPAC
```

TCF LARGE2

```
LARGE3          COM                             # LEAVE NEGATIVE OF SHIFT COUNT-1 FOR
                TS      MPTEMP                  # PREDIVIDE LEFT SHIFT.

                COM                             # PICK UP REQUIRED SHIFTING BIT TO UNNORM-
                INDEX   A                       # ALIZE THE SQRT RESULT.
                CAF     BIT14
                TS      BUF
                EXTEND
                MP      MPAC    +1
                XCH     BUF
                EXTEND                          # (UNNORMALIZE THE SQRT FOR LV).
                MP      MPAC
                XCH     L
                AD      BUF
                XCH     L

LARGE2          INDEX   FIXLOC
                DXCH    LV                      # LENGTH NOW STORED IN WORK AREA.

                CS      ONE
                TS      MAXDVSW                 # NO MAXDV CASES IN UNIT.

                DXCH    VBUF                    # PREPARE X COMPONENT FOR DIVIDE, SETTING
                DXCH    MPAC                    # LENGTH OF VECTOR AS DIVISOR IN BUF.
                DXCH    BUF
                TC      UNITDV

                DXCH    VBUF    +2              # DO Y AND Z IN USUAL FASHION SO WE CAN
                DXCH    MPAC                    # EXIT THROUGH VROTATEX.
                DXCH    MPAC    +3
                TC      UNITDV

                DXCH    VBUF    +4
                DXCH    MPAC
                DXCH    MPAC    +5
                TC      UNITDV
                TCF     VROTATEX                # AND EXIT.
```

```
# IF THE LENGTH OF THE ARGUMENT VECTOR WAS LESS THAN 2(-28), EACH COMPONENT MUST BE SHIFTED LEFT AT LEAST
# 14 PLACES BEFORE THE DIVIDE. NOTE THAT IN THIS CASE, THE MAJOR PART OF EACH COMPONENT IS ZERO.

SMALL           TS      MPTEMP          # NEGATIVE OF PRE-DIVIDE SHIFT COUNT.

                CAF     ZERO            # SHIFT EACH COMPONENT LEFT 14.
                XCH     VBUF    +1
                XCH     VBUF
                XCH     VBUF    +3
                XCH     VBUF    +2
                XCH     VBUF    +5
                XCH     VBUF    +4

                CS      MPTEMP
                INDEX   A
                CAF     BIT14
                EXTEND
                MP      MPAC
                TCF     SMALL2

THIRTEEN        =       OCT15
FOURTEEN        =       OCT16
OCT16           =       R1D1
```

```
# THE FOLLOWING ROUTINE SETS UP THE CALL TO THE DIVIDE ROUTINES.

UNITDV          CCS     MPAC            # FORCE MPAC POSITIVE IF POSSIBLE, SETTING
                TCF     UMPAC+          # DVSIGN ACCORDING TO THE SIGN OF MPAC
                TCF     +2              # SINCE THE DIVISOR IS ALWAYS POSITIVE
                TCF     UMPAC-          # HERE.

                CCS     MPAC    +1
                TCF     UMPAC+
                TC      Q               # EXIT IMMEDIATELY ON ZERO.
                TCF     UMPAC-
                TC      Q

UMPAC-          CS      ZERO            # IF NEGATIVE, SET -0 IN DVSIGN FOR FINAL
                TS      DVSIGN          # COMPLEMENT.
                EXTEND
                DCS     MPAC            # PICK UP ABSOLUTE VALUE OF ARG AND JUMP.
                INDEX   MPTEMP
                TCF     MAXTEST -1

UMPAC+          TS      DVSIGN          # SET DVSIGN FOR POSITIVE QUOTIENT.
                DXCH    MPAC
                INDEX   MPTEMP
                TCF     MAXTEST -1
```

```
# MISCELLANEOUS UNARY OPERATIONS.

DSQ             TC      DSQSUB          # SQUARE THE DP CONTENTS OF MPAC.
                TCF     DANZIG

ABVALABS        CCS     MODE            # ABVAL OR ABS INSTRUCTION.
                TCF     ABS             # DO ABS ON SCALAR.
                TCF     ABS

ABVAL           TC      VSQSUB          # DOT MPAC WITH ITSELF.
                LXCH    MODE            # MODE IS NOW DP (L ZERO AFTER DAS).

                EXTEND                  # STORE SQUARE OF LENGTH IN WORK AREA.
                DCA     MPAC
                INDEX   FIXLOC
                DXCH    LVSQUARE
```

```
# PROGRAM DESCRIPTION -- SUBROUTINE SQRT
#
# FUNCTIONAL DESCRIPTION -- DOUBLE PRECISION SQUARE ROOT ROUTINE
#       THIS PROGRAM TAKES THE SQUARE ROOT OF THE 27 OR 28 MOST SIGNIFICANT BITS IN THE TRIPLE PRECISION SET OF
#       NUMBERS -- MPAC, MPAC+1, AND MPAC+2.  THE ROOT IS RETURNED DOUBLE PRECISION IN MPAC AND MPAC+1.
#
# WARNING -- THIS SUBROUTINE USES A TRIPLE PRECISION INPUT.  THE PROGRAMMER MUST ASSURE THE CONTENTS OF MPAC+2
#       ESPECIALLY IF THE CONTENTS OF MPAC IS SMALL OR ZERO.  FOR DETAILS SEE STG MEMO NO.949.
#
# CALLING SEQUENCE -- IN INTERPRETIVE MODE, I.E., FOLLOWING `TC INTPRET', `SQRT', NO ADDRESS IS ALLOWED.
#       INPUT SCALING: THE BINARY POINT IS ASSUMED TO THE RIGHT OF BIT 15.  THE ANSWER IS RETURNED WITH THE SAME SCALING.
#
# SUBROUTINES -- GENSCR, MPACSHR, SQRTSUB, ABORT
#
# ABORT EXIT MODE -- ABORTS ON NEGATIVE INPUT -1.2X10E-4 (77775 OCTAL) OR LESS.
#       DISPLAYS ERROR CODE 1302
#               TC      ABORT
#               OCT     1302
#
# DEBRIS -- LOCATIONS BUF, MPTEMP, ADDRWD ARE USED

SQRT            TC      SQRTSUB         # TAKE THE SQUARE ROOT OF MPAC.
                CCS     MPTEMP          # RETURNED NORMALIZED SQUARE ROOT.  SEE IF
                TCF     +2              # ANY UN-NORMALIZATION REQUIRED AND EXIT
                TCF     DANZIG          # IF NOT.

                AD      NEG12           # A RIGHT SHIFT OF MORE THAN 13 COULD BE
                EXTEND                  # REQUIRED IF INPUT WAS ZERO IN MPAC,+1.
                BZMF    SQRTSHFT        # GOES HERE IN MOST CASES.
                ZL                      # IF A LONG SHIFT IS REQUIRED, GO TO
                LXCH    ADDRWD          # GENERAL RIGHT SHIFT ROUTINES.
                TCF     GENSCR  +4      # ADDRWD WAS ZERO TO PREVENT ROUND.

SQRTSHFT        INDEX   MPTEMP          # SELECT SHIFTING BIT AND EXIT THROUGH
                CAF     BIT15           # SHIFT ROUTINES.
                TS      MPTEMP
                CAF     ZERO            # TO ZERO MPAC +2 IN THE PROCESS.
                TCF     MPACSHR +3

ABS             TC      BRANCH          # TEST SIGN OF MPAC AND COMPLEMENT IF
                TCF     DANZIG
                TCF     DANZIG
                TCF     COMP
```

```
VDEF            CS      FOUR            # VECTOR DEFINE -- ESSENTIALLY TREATS
                ADS     PUSHLOC         # SCALAR IN MPAC AS X COMPONENT, PUSHES UP
                EXTEND                  # FOR Y AND THEN AGAIN FOR Z.
                INDEX   A
                DCA     2
                DXCH    MPAC    +3
                EXTEND
                INDEX   PUSHLOC
                DCA     0
                DXCH    MPAC    +5
                TCF     VMODE           # MODE IS NON VECTOR.

VSQ             TC      VSQSUB          # DOT MPAC WITH ITSELF.
                TCF     DMODE           # MODE IS NOW DP.

PUSH            EXTEND                  # PUSH DOWN MPAC LEAVING IT LOADED.
                DCA     MPAC
                INDEX   PUSHLOC         # PUSH DOWN FIRST TWO REGISTERS IN EACH
                DXCH    0

                INDEX   MODE            # INCREMENT PUSHDOWN POINTER.
                CAF     NO.WDS
                ADS     PUSHLOC

                CCS     MODE
                TCF     TPUSH           # PUSH DOWN MPAC +2.
                TCF     DANZIG          # DONE FOR DP.

                EXTEND                  # ON VECTOR, PUSH DOWN Y AND Z COMPONENTS.
                DCA     MPAC    +3
                INDEX   PUSHLOC
                DXCH    0       -4
                EXTEND
                DCA     MPAC    +5
                INDEX   PUSHLOC
                DXCH    0       -2
                TCF     DANZIG

TPUSH           CA      MPAC    +2
                TCF     ENDTPUSH +2

RVQ             INDEX   FIXLOC          # RVQ -- RETURN IVA QPRET.
                CA      QPRET
                TS      POLISH
                TCF     GOTO    +4      # (ASSUME QPRET POINTS TO FIXED ONLY.)
```

```
# THE FOLLOWING SUBROUTINES ARE USED IN SQUARING MPAC, IN BOTH THE SCALAR AND VECTOR SENSE.  THEY ARE
# SPECIAL CASES OF DMPSUB AND DOTSUB, PUT IN TO SAVE SOME TIME.

DSQSUB          CA      MPAC    +1      # SQUARES THE SCALAR CONTENTS OF MPAC.
                EXTEND
                SQUARE
                TS      MPAC    +2
                CAF     ZERO            # FORM 2(CROSS TERM).
                XCH     MPAC    +1
                EXTEND
                MP      MPAC
                DDOUBL                  # AND MAYBE OVEFLOW.
                DAS     MPAC    +1      # AND SET A TO NET OVERFLOW.
                XCH     MPAC
                EXTEND
                SQUARE
                DAS     MPAC
                TC      Q

VSQSUB          EXTEND                  # DOTS THE VECTOR IN MPAC WITH ITSELF.
                QXCH    DOTRET
                TC      DSQSUB          # SQUARE THE X COMPONENT.
                DXCH    MPAC    +3
                DXCH    MPAC
                DXCH    BUF             # SO WE CAN END IN DOTSUB.
                CA      MPAC    +2
                TS      BUF     +2

                TC      DSQSUB          # SQUARE Y COMPONENT.
                DXCH    MPAC    +1
                DAS     BUF     +1
                AD      MPAC
                AD      BUF
                TS      BUF
                TCF     +2
                TS      OVFIND          # IF OVERFLOW.

                DXCH    MPAC    +5
                DXCH    MPAC
                TC      DSQSUB          # SQUARE Z COMPONENT.
                TCF     ENDDOT          # END AS IN DOTSUB.
```

```
# DOUBLE PRECISION SQUARE ROOT ROUTINE.  TAKE THE SQUARE ROOT OF THE TRIPLE PRECISION (MPAC +2 USED ONLY
# IN NORMALIZATION) CONTENTS OF MPAC AND LEAVE THE NORMALIZED RESULT IN MPAC (C(MPAC) GREATER THAN OR EQUAL TO
# .5).  THE RIGHT SHIFT COUNT (TC UNNORMALIZE) IS LEFT IN MPTEMP.

SQRTSUB         CAF     ZERO            # START BY ZEROING RIGHT SHIFT COUNT.
                TS      MPTEMP

                CCS     MPAC            # CHECK FOR POSITIVE ARGUMENT, SHIFTING
                TCF     SMPAC+          # FIRST SIGNIFICANT MPAC REGISTER INTO
                TCF     +2              # MPAC ITSELF.
                TCF     SQRTNEG         # SEE IF MAG OF ARGUMENT LESS THAN 10(-4).

                XCH     MPAC    +2      # MPAC IS ZERO -- SHIFT LEFT 14.
                XCH     MPAC    +1
                TS      MPAC
                CAF     SEVEN           # AUGMENT RIGHT SHIFT COUNTER.
                TS      MPTEMP

                CCS     MPAC            # SEE IF MPAC NOW PNZ.
                TCF     SMPAC+
                TCF     +2
                TCF     ZEROANS         # NEGATIVE BUT LESS THAN 10(-4) IN MAG.

                XCH     MPAC    +1      # XERO -- SHIFT LEFT 14 AGAIN.
                TS      MPAC
                CAF     SEVEN           # AUGMENT RIGHT SHIFT COUNTER.
                ADS     MPTEMP

                CCS     MPAC
                TCF     SMPAC+
                TC      Q               # SQRT(0) = 0.
                TCF     ZEROANS
                TCF     FIXROOT         # DO NOT LEAVE SQRTSUB WITH -0 IN MPAC.

SQRTNEG         CCS     A               # ARGUMENT IS NEGATIVE, BUT SEE IF SIGN-
                TCF     SQRTABRT        # CORRECTED ARGUMENT IS LESS THAN 10(-4)

                CCS     MPAC    +1      # IN MAGNITUDE.  IF SO, CALL ANSWER ZERO.
ZEROANS         CAF     ZERO            # FORCE ANSWER TO ZERO HERE.
                TCF     FIXROOT
                TCF     SQRTABRT
                TCF     FIXROOT

SQRTABRT        DXCH    LOC
                TC      POODOO1
                OCT     1302
```

```
SMPAC+          AD      -1/2+2          # SEE IF ARGUMENT GREATER THAN OR EQUAL TO
                EXTEND                  # .5.
                BZMF    SRTEST          # IF SO, SEE IF LESS THAN .25.

                DXCH    MPAC            # WE WILL TAKE THE SQUARE ROOT OF MPAC/2.
                LXCH    SR              # SHIFT RIGHT 1 AND GO TO THE SQRT ROUTINE
                EXTEND
                MP      HALF
                DXCH    MPAC
                XCH     SR
                ADS     MPAC    +1      # GUARANTEED NO OVERFLOW.

ARGHI           CAF     SLOPEHI         # ARGUMENT BETWEEN .25 AND .5, GET A
                EXTEND                  # LINEAR APPROXIMATION FOR THIS RANGE.
                MP      MPAC
                AD      BIASHI          # X0/2 = (MPAC/2)(SLOPHI) + BIASHI/2.

 +4             TS      BUF             # X0/2 (ARGLO ENTERS HERE).
                CA      MPAC            # SINGLE-PRECISION THROUGHOUT.
                ZL
                EXTEND
                DV      BUF             # (MPAC/2)/(X0/2)
                EXTEND
                MP      HALF
                ADS     BUF             # X1 = X0/2 + .5(MPAX/2)/(X0/2)

                EXTEND
                MP      HALF            # FORM UP X1/2.
                DXCH    MPAC            # SAVE AND BRING OUT ARGUMENT.
                EXTEND                  # TAKE DP QUOTIENT WITH X1.
                DV      BUF
                TS      BUF     +1      # SAVE MAJOR PART OF QUOTIENT.
                CAF     ZERO            # FORM MINOR PART OF QUOTIENT USING
                XCH     L               # (REMAINDER,0).
                EXTEND
                DV      BUF
                TS      L               # IN PREPARATION FOR DAS.
                CA      BUF     +1
                DAS     MPAC            # X2 = X1/2 + (MPAC/2)X1

                EXTEND                  # OVERFLOWS IF ARG. NEAR POSMAX.
                BZF     TCQBNK00
                CAF     POSMAX
FIXROOT         TS      MPAC
                TS      MPAC    +1
TCQBNK00        TC      Q               # RETURN TO CALLER TO UNNORMALIZE, ETC.
```

```
SRTEST          AD      QUARTER         # ARGUMENT WAS LESS THAN .5, SEE IF LESS
                EXTEND                  # THAN .25.
                BZMF    SQRTNORM        # IF SO, BEGIN NORMALIZATION.

                DXCH    MPAC            # IF BETWEEN .5 AND .25, SHIFT RIGHT 1 AND
                LXCH    SR              # START AT ARGLO.
                EXTEND
                MP      HALF
                DXCH    MPAC
                XCH     SR
                ADS     MPAC    +1      # NO OVERFLOW.

ARGLO           CAF     SLOPELO         # (NORMALIZED) ARGUMENT BETWEEN .125 AND
                EXTEND                  # .25
                MP      MPAC
                AD      BIASLO
                TCF     ARGHI   +4      # BEGIN SQUARE ROOT.

SQRTNM2         EXTEND                  # SHIFT LEFT 2 AND INCREMENT RIGHT SHIFT
                DCA     MPAC    +1      # COUNT (FOR TERMINAL UNNORMALIZATION).
                DAS     MPAC    +1
                AD      MPAC
                ADS     MPAC            # (NO OVERFLOW).

SQRTNORM        INCR    MPTEMP          # FIRST TIME THROUGH, JUST SHIFT LEFT 1
                EXTEND                  # (PUTS IN EFFECTIVE RIGHT SHIFT SINCE
                DCA     MPAC    +1      # WE WANT MPAC/2).
                DAS     MPAC    +1
                AD      MPAC
                ADS     MPAC            # (AGAIN NO OVERFLOW).
                DOUBLE
                TS      CYL

NORMTEST        CCS     CYL             # SEE IF ARGUMENT NOW NORMALIZED AT
                CCS     CYL             # GREATER THAN .125.
                TCF     SQRTNM2         # NO -- SHIFT LEFT 2 MORE AND TRY AGAIN.
                TCF     ARGHI           # YES -- NOW BETWEEN .5 AND .25.
                TCF     ARGLO           # ARGUMENT NOW BETWEEN .25 AND .125.
```

```
# TRIGONOMETRIC FUNCTION PACKAGE.
#       THE FOLLOWING TRIGONOMETRIC FUNCTIONS ARE AVAIALABLE AS INTERPRETIVE OPERATIONS:
#       1.      SIN             COMPUTES (1/2)SINE(2 PI MPAC).
#       2.      COS             COMPUTES (1/2)COSINE(2 PI MPAC).
#       3.      ASIN            COMPUTES (1/2PI)ARCSINE(2 MPAC).
#       4.      ACOS            COMPUTES (1/2PI)ARCCOSINE(2 MPAC).
#
# SIN-ASIN AND COS-ACOS ARE MUTUALLY INVERSE, I.E., SIN(ASIN(X)) = X.

COSINE          TC      BRANCH          # FINDS COSINE USING THE IDENTITY
                TCF     +3              # COS(X) = SIN(PI/2 - ABS(X)).
                TCF     PRESINE
                TCF     PRESINE

 +3             EXTEND
                DCS     MPAC
                DXCH    MPAC

PRESINE         CAF     QUARTER         # PI/2 SCALED.
                ADS     MPAC

SINE            DXCH    MPAC            # DOUBLE ARGUMENT.
                DDOUBL
                OVSK                    # SEE IF OVERFLOW PRESENT.
                TCF     +3              # IF NOT, ARGUMENT OK AS IS.

                EXTEND                  # IF SO, WE LOST (OR GAINED) PI, SO
                DCOM                    # COMPLEMENT MPAC USING THE IDENTITY
                                        # SIN(X-(+)PI) = SIN(-X).
 +3             DXCH    MPAC
                CA      MPAC            # SEE IF ARGUMENT GREATER THAN .5 IN
                DOUBLE                  # MAGNITUDE.  IF SO, REDUCE IT TO LESS THAN
                TS      L               # .5 (+-PI/2 SCALED) AS FOLLOWS:
                TCF     SN1

                INDEX   A               # IF POSITIVE, FORM PI - X, IF NEGATIVE
                CAF     NEG1/2 +1       # USE -PI -X.
                DOUBLE
                EXTEND
                SU      MPAC            # GUARANTEED NO OVERFLOW.
                TS      MPAC
                CS      MPAC    +1
                TS      MPAC    +1
```

```
SN1             EXTEND                          # SET UP TO EVALUATE HASTINGS POLYNOMIAL
                DCA     MPAC
                DXCH    BUF2
                TC      DSQSUB                  # SQUARE MPAC.

                TC      POLY                    # EVALUATE FOURTH ORDER POLYNOMIAL.
                DEC     3
                2DEC    +.3926990796

                2DEC    -.6459637111

                2DEC    +.318758717

                2DEC    -.074780249

                2DEC    +.009694988

                CAF     LBUF2                   # MULTIPLY BY ARGUMENT AND SHIFT LEFT 2.
                TC      DMPSUB  -1

                EXTEND
                DCA     MPAC    +1
                DAS     MPAC    +1
                AD      MPAC
                ADS     MPAC                    # NEITHER SHIFT OVERFLOWS.
                EXTEND
                DCA     MPAC    +1
                DAS     MPAC    +1
                AD      MPAC
                ADS     MPAC
                TCF     DANZIG
```

```
# ARCSIN/ARCCOS ROUTINE.

ARCSIN          CAF     LASINEX         # COMPUTE ARCSIN BY USING THE IDENTITY
                TCF     +2              # ARCSIN(X) = PI/2 - ARCCOS(X).

ARCCOS          CAF     LDANZIG         # (EXITS IMMEDIATELY).
                TS      ESCAPE
                TC      BRANCH          # TEST SIGN OF INPUT.
                TCF     ACOSST          # START IMMEDIATELY IF POSITIVE.
                TCF     ACOSZERO        # ARCCOS(0) = PI/2 = .25.
                EXTEND                  # IF NEGATIVE, USE THE IDENTITY
                DCS     MPAC            # ARCCOS(X) = PI - ARCCOS(-X), FORCING
                DXCH    MPAC            # ARGUMENT POSITIVE.
                CAF     TCSUBTR         # SET EXIT TO DO ABOVE BEFORE
                XCH     ESCAPE          # ARCSIN/ARCCOS CONSIDERATIONS.
                TS      ESCAPE2

ACOSST          CS      HALF            # TEST MAGNITUDE OF INPUT.
                AD      MPAC
                CCS     A
                TCF     ACOSOVF         # THIS IS PROBABLY AN OVERFLOW CASE.

LASINEX         TCF     ASINEX

                TCF     ACOSST2         # NO OVERFLOW -- PROCEED.

                CCS     MPAC    +1      # IF MAJOR PART IS .5, CALL ANSWER 0
                CAF     ZERO            # UNLESS MINOR PART NEGATIVE.
                TCF     ACOS=0

                TCF     ACOSST2

ACOS=0          TS      MPAC    +1
                TS      MPAC
                TC      ESCAPE

ACOSST2         EXTEND                  # NOW THAT ARGUMENT IS IN PROPER RANGE,
                DCS     MPAC            # BEGIN COMPUTATION.  USE HASTINGS
                AD      HALF            # APPROXIMATION ARCCOS(X) = SQRT(1-X)P(X)
                DXCH    MPAC            # IN A SCALED VERSION WHERE P(X) IS A
                DXCH    BUF2            # SEVENTH ORDER POLYNOMIAL.

                TC      SQRTSUB         # RETURNS WITH NORMALIZED SQUARE ROOT.

                CCS     MPTEMP          # SEE IF UN-NORMALIZATION REQUIRED.
                TCF     ACOSSHR         # IF SO.
```

```
ACOS3           DXCH    MPAC            # SET UP FOR POLYNOMIAL EVALUATION.
                DXCH    BUF2
                DXCH    MPAC

                TC      POLY
                DEC     6
                2DEC    +.353553385     # COEFFICIENTS ARE C 2(+I)/PISQRT(2) WHERE

                2DEC*   -.0483017006 B+1*       # I

                2DEC*   +.0200273085 B+2*       # WHERE C STANDS FOR ORIGINAL COEFFS.

                2DEC*   -.0112931863 B+3*

                2DEC*   +.00695311612 B+4*

                2DEC*   -.00384617957 B+5*

                2DEC*   +.001501297736 B+6*

                2DEC*   -.000284160334 B+7*


                CAF     LBUF2           # DO FINAL MULTIPLY AND GO TO ANY
                TC      DMPSUB  -1      # EPILOGUE SEQUENCES.
                TC      ESCAPE

SUBTR           EXTEND                  # EPILOGUE FOR NEGATIVE INPUTS TO ARCCOS.
                DCS     MPAC
                AD      HALF            # FORMS PI - ARCCOS(-X) = ARCCOS(X).
                DXCH    MPAC
                TC      ESCAPE2         # GO TO POSSIBLE ARCSIN EPILOGUE.

ASINEX          EXTEND
                DCS     MPAC            # ARCSIN EPILOGUE -- GET ARCSIN(X)
                AD      QUARTER         # = PI/2 - ARCCOS(X).
                DXCH    MPAC
LDANZIG         TCF     DANZIG
```

```
ACOSSHR         INDEX   A               # THE SHIFT RIGHT IS LESS THAN 14 SINCE
                CAF     BIT14           # THE INPUT WAS NON-ZERO DP.
                TS      MPTEMP
                TC      VSHRRND         # DP SHIFT RIGHT AND ROUND.
                TCF     ACOS3           # PROCEED.

ACOSOVF         EXTEND                  # IF MAJOR PART WAS ONLY 1 MORE THAN .5,
                BZF     ACOS=0          # CALL ANSWER ZERO.

ACOSABRT        EXTEND                  # IF OVERFLOW, CALL ANSWER ZERO BUT
                DCA     LOC             # SOUND AN ALARM.
                TC      ALARM1
                OCT     1301

                CAF     ZERO
                TCF     ACOS=0

ACOSZERO        CAF     QUARTER         # ACOS(0) = PI/2.
                TCF     ACOS=0  +1      # SET MPAC AND EXIT VIA ESCAPE.

NEG12           DEC     -12
TCSUBTR         TCF     SUBTR
```

```
# THE FOLLOWING INSTRUCTIONS ARE AVAILABLE FOR SETTING, MODIFYING, AND BRANCHING ON INDEX REGISTERS:
#        1.      AXT     ADDRESS TO INDEX TRUE.
#        2.      AXC     ADDRESS TO INDEX COMPLEMENTED.
#        3.      LXA     LOAD INDEX FROM ERASABLE.
#        4.      LXC     LOAD INDEX COMPLEMENTED FROM ERASABLE.
#        5.      SXA     STORE INDEX IN ERASABLE.
#        6.      XCHX    EXCHANGE INDEX REGISTER WITH ERASABLE.
#        7.      INCR    INCREMENT INDEX REGISTER.
#        8.      XAD     ERASABLE ADD TO INDEX REGISTER.
#        9.      XSU     ERASABLE SUBTRACT FROM INDEX REGISTER.
#        10.     TIX     BRANCH ON INDEX REGISTER AND DECREMENT.

                BANK    01

                COUNT*  $$/INTER
AXT             TC      TAGSUB          # SELECT APPROPRIATE INDEX REGISTER.
                CA      POLISH
XSTORE          INDEX   INDEXLOC        # CONTAINS C(FIXLOC) OR C(FIXLOC)+1
                TS      X1
                TCF     DANZIG

AXC             TC      TAGSUB
                CS      POLISH
                TC      XSTORE

LXA             TC      15ADRERS        # LOAD INDEX REGISTER FROM ERASABLE.
                INDEX   POLISH
                CA      0
                TCF     XSTORE

LXC             TC      15ADRERS        # LOAD NDX REG FROM ERASABLE COMPLEMENTED.
                INDEX   POLISH
                CS      0
                TCF     XSTORE

SXA             TC      15ADRERS        # STORE INDEX REGISTER IN ERASABLE.
                INDEX   INDEXLOC
                CA      X1
MSTORE1         INDEX   POLISH
                TS      0
                TCF     DANZIG
```

```
XCHX            TC      15ADRERS        # EXCHANGE INDEX REGISTER WITH ERASABLE.
                INDEX   POLISH
                CA      0
                INDEX   INDEXLOC
                XCH     X1
                TCF     MSTORE1

XAD             TC      15ADRERS        # ADD ERASABLE TO INDEX REGISTER.
                INDEX   POLISH
                CA      0
XAD2            INDEX   INDEXLOC
                ADS     X1              # IGNORING OVERFLOWS.
                TCF     DANZIG

INCR            TC      TAGSUB          # INCREMENT INDEX REGISTER.
                CA      POLISH
                TCF     XAD2

XSU             TC      15ADRERS        # SUBTRACT ERASABLE FROM INDEX REGISTER.
                INDEX   POLISH
                CS      0
                TCF     XAD2

TIX             TC      TAGSUB          # BRANCH AND DECREMENT ON INDEX.
                INDEX   INDEXLOC
                CS      S1
                INDEX   INDEXLOC
                AD      X1
                EXTEND                  # NO OPERATION IF DECREMENTED INDEX IS
                BZMF    DANZIG          # NEGATIVE OR ZERO.

DOTIXBR         INDEX   INDEXLOC
                XCH     X1              # IGNORING OVERFLOWS.

                TCF     GOTO            # DO THE BRANCH USING THE CADR IN POLISH.
```

```
# SUBROUTINE TO CONVERT AN ERASABLE ADDRESS (11 BITS) TO AN EBANK SETTING AND SUBADDRESS.

15ADRERS        CS      POLISH
                AD      DEC45
                CCS     A               # DOES THE ADDRESS POINT TO THE WORK AREA?
                CA      FIXLOC          # YES.  ADD FIXLOC.  EBANK OK AS IS.
                TCF     +5

                CA      OCT1400         # NO. SET EBANK & MAKE UP SUBADDRESS.
                XCH     POLISH
                TS      EBANK
                MASK    LOW8
 +5             ADS     POLISH          # FALL INTO TAGSUB, AND RETURN VIA Q.

# SUBROUTINE WHICH SETS THE ADDRESS OF THE SPECIFIED INDEX IN INDEXLOC.  (ACTUALLY, THE ADDRESS -38D.)

TAGSUB          CA      FIXLOC
                TS      INDEXLOC

                CCS     CYR             # BIT 15 SPECIFIES INDEX.
                INCR    INDEXLOC        # 0 MEANS USE X2.
                TC      Q
                TC      Q               # 1 FOR X1.
```

```
# MISCELLANEOUS OPERATION CODES WITH DIRECT ADDRESSES.  INCLUDED HERE ARE:
#        1.      ITA     STORE CPRET (RETURN ADDRESS) IN ERASABLE.
#        2.      CALL    CALL A SUBROUTINE, LEAVING RETURN IN QPRET.
#        3.      RTB     RETURN TO BASIC LANGUAGE AT THE GIVEN ADDRESS.
#        4.      BHIZ    BRANCH IF THE HIGH ORDER OF MPAC IS ZERO (SINGLE PRECISION).
#        5.      BOV     BRANCH ON OVERFLOW.
#        6.      GOTO    SIMPLE SEQUENCE CHANGE.

RTB/BHIZ        CCS     CYR
RTB             CA      POLISH
                TC      SWCALL  -1      # SO A "TC Q" FROM ROUTINE LEADS TO DANZIG

BHIZ            CCS     MPAC
                TCF     DANZIG
                TCF     GOTO
                TCF     DANZIG
                TCF     GOTO

BOV(B)          CCS     OVFIND          # BRANCH ON OVERFLOW TO BASIC OR INTERP.
                TCF     +2
                TCF     DANZIG
                TS      OVFIND
                CCS     CYR
                TCF     RTB             # IF BASIC.
B5TOBB          OCT     360
                TCF     GOTO
```

```
BZE/GOTO        CCS     CYR             # SEE WHICH OP-CODE IS DESIRED.
                TC      BRANCH          # DO BZE.
                TCF     DANZIG
                TCF     GOTO            # DO GOTO.
                TCF     DANZIG

BPL/BMN         CCS     CYR
                TCF     BPL
5B10            DEC     5 B+10          # SHIFTS OP CODE IN SWITCH INSTRUCTION ADR

                TC      BRANCH          # DO BMN
                TCF     DANZIG
                TCF     DANZIG
                TCF     GOTO            # ONLY IF NNZ.

BPL             TC      BRANCH
                TCF     GOTO            # IF POSITIVE OR ZERO.
                TCF     GOTO
                TCF     DANZIG

CALL/ITA        CCS     CYR
                TCF     CALL

                TC      CCSHOLE
                TC      15ADRERS        # STORE QPRET.  (TAGSUB AFTER 15ADRERS IS
                INDEX   FIXLOC          # SLOW IN THIS CASE, BUT SAVES STORAGE.)
                CA      QPRET
                TCF     MSTORE1
```

```
# THE FOLLOWING OPERATIONS ARE AVAILABLE FOR ALTERING AND TESTING INTERPRETATIVE SWITCHES:

#       00      BONSET          SET A SWITCH AND DO A GOTO IF IT WAS ON.
#       01      SETGO           SET A SWITCH AND DO A GOTO.
#       02      BOFSET          SET A SWITCH AND DOA GOTO IF IT WAS OFF
#       03      SET             SET A SWITCH.

#       04      BONINV          INVERT A SWITCH AND BRANCH IF IT WAS ON.
#       05      INVGO           INVERT A SWITCH AND DO A GOTO.
#       06      BOFINV          INVERT A SWITCH AND BRANCH IF IT WAS OFF
#       07      INVERT          INVERT A SWITCH.

#       10      BONCLR          CLEAR A SWITCH AND BRANCH IF IT WAS ON.
#       11      CLRGO           CLEAR A SWITCH AND DO A GOTO.
#       12      BOFCLR          CLEAR A SWITCH AND BRANCH IF IT WAS OFF.
#       13      CLEAR           CLEAR A SWITCH.

#       14      BON             BRANCH IF A SWITCH WAS ON.
#       16      BOFF            BRANCH IF A SWITCH WAS OFF.

# THE ADDRESS SUPPLIED WITH THE SWITCH INSTRUCTION IS INTERPRETED AS FOLLOWS:

#       BITS 1-4        SWITCH BIT NUMBER (1-15).
#       BITS 5-8        SWITCH OPERATION NUMBER
#       BITS 9-         SWITCH WORD NUMBER (UP TO 64 SWITCH WORDS).

# THE ADDRESS ITSELF IS MADE UP BY THE YUL SYSTEM ASSEMBLER.  THE BRANCH INSTRUCTIONS REQUIRE TWO
# ADDRESSES, THE SECOND TAKEN AS THE DIRECT (OR INDIRECT IF IN ERASABLE) ADDRESS OF THE BRANCH.

SWITCHES        CAF     LOW4            # LEAVE THE SWITCH BIT IN SWBIT.
                MASK    POLISH
                INDEX   A
                CAF     BIT15           # (NUMBER FROM LEFT TO RIGHT.)
                TS      SWBIT

                CAF     BIT7            # LEAVE THE SWITCH NUMBER IN SWWORD.
                EXTEND
                MP      POLISH
                TS      SWWORD

                INHINT                  # DURING SWITCH CHANGE SO RUPT CAN USE TOO
                INDEX   A               # LEAVE THE SWITCH WORD ITSELF IN L.
                CA      STATE
                TS      Q               # Q WILL BE USED AS A CHANNEL.
```

```
                CAF     BIT11
                EXTEND                  # DISPATCH SWITCH BIT OPERATION AS IN BITS
                MP      POLISH          # 7-8 OF POLISH.
                MASK    B3TOB4          # GETS 4X2-BIT CODE.
                INDEX   A
                TCF     +1

 +1             CA      SWBIT           # 00 -- SET SWITCH IN QUESTION.
                EXTEND
                ROR     QCHAN
                TCF     SWSTORE

 +5             CA      SWBIT           # 01 -- INVERT SWITCH.
                EXTEND
                RXOR    QCHAN
                TCF     SWSTORE

 +9D            CS      SWBIT           # 10 -- CLEAR.
                MASK    Q
SWSTORE         INDEX   SWWORD
                TS      STATE           # NEW SWITCH WORD.
```

```
 +13D           RELINT                  # 11 -- NOOP.
                CAF     BIT13
                EXTEND                  # DISPATCH SEQUENCE CHANGING OR BRANCING
                MP      POLISH          # CODE.
                MASK    B3TOB4
                INDEX   A
                TCF     +1              # ORIGINALLY STORED IN BITS 5-6

 +1             CS      Q               # 00 -- BRANCH IF ON.
TEST            MASK    SWBIT
                CCS     A
                TCF     SWSKIP

 +5             TCF     SWBRANCH        # 01 -- GO TO.

                TCF     SWSKIP          # HERE ONLY ON BIT 15.

                TC      CCSHOLE
                TC      CCSHOLE

 +9D            CA      Q               # 10 -- BRANCH IF OFF.
                TCF     TEST

B3TOB4          OCT     0014
SWSKIP          INCR    LOC

SW/             EQUALS  SWITCHES

 +13D           TCF     DANZIG          # 11 -- NOOP.
```

```
                BLOCK   02

                COUNT*  $$/FCONS
# THE FOLLOWING TABLE OF 18 VALUES IS INDEXED. DO NOT INSERT OR REMOVE ANY QUANTITIES

DPOSMAX         OCT     37777           # MUST PRECEDE POSMAX
POSMAX          OCT     37777

LIMITS          =       NEG1/2

NEG1/2          OCT     -20000          # USED BY SIN ROUTINE (MUST BE TWO
                                        # LOCATIONS IN FRONT OF BIT14)

# BIT TABLE

BIT15           OCT     40000
BIT14           OCT     20000
BIT13           OCT     10000
BIT12           OCT     04000
BIT11           OCT     02000
BIT10           OCT     01000
BIT9            OCT     00400
BIT8            OCT     00200
BIT7            OCT     00100
BIT6            OCT     00040
BIT5            OCT     00020
BIT4            OCT     00010
BIT3            OCT     00004
BIT2            OCT     00002
BIT1            OCT     00001

# DO NOT DESTROY THIS COMBINATION, SINCE IT IS USED IN DOUBLE PRECISION INSTRUCTIONS.
NEG0            OCT     -0              # MUST PRECEDE ZERO
ZERO            OCT     0               # MUST FOLLOW NEG0
# BIT1          OCT     00001
# NO.WDS        OCT     2               # INTERPRETER
# OCTAL3        OCT     3               # INTERPRETER
# R3D1          OCT     4               # PINBALL
FIVE            OCT     5
# REVCNT        OCT     6               # INTERPRETER
SEVEN           OCT     7
# BIT4          OCT     00010
# R2D1          OCT     11              # PINBALL
OCT11           =       R2D1            # P20S
# BINCON        DEC     10              # PINBALL           (OCTAL 12)
ELEVEN          DEC     11
# OCT14         OCT     14              # ALARM AND ABORT (FILLER)
OCT15           OCT     15
# R1D1          OCT     16              # PINBALL
LOW4            OCT     17
```

```
# BIT5          OCT     00020
# ND1           OCT     21              # PINBALL
# VD1           OCT     23              # PINBALL
# OCT24         OCT     24              # SERVICE ROUTINES
# MD1           OCT     25              # PINBALL
BITS4&5         OCT     30
# OCT31         OCT     31              # SERVICE ROUTINES
OCT33           OCT     33
DEC27           =       OCT33
OCT35           OCT     35
DEC29           =       OCT35
CALLCODE        OCT     00032
# LOW5          OCT     37              # PINBALL
# 33DEC         DEC     33              # PINBALL             (OCTAL 41)
# 34DEC         DEC     34              # PINBALL             (OCTAL 42)
TBUILDFX        DEC     37              # BUILDUP FOR CONVIENCE IN DAPTESTING
TDECAYFX        DEC     38              # CONVENIENCE FOR DAPTESTING
# BIT6          OCT     00040
OCT50           OCT     50
DEC45           DEC     45
SUPER011        OCT     60              # BITS FOR SUPERBNK SETTING 011.
.5SEC           DEC     50
# BIT7          OCT     00100

SUPER100        =       BIT7            # BITS FOR SUPERBNK SETTING 100
                                        # (LAST 4K OF ROPE)
SUPER101        OCT     120             # BITS FOR SUPERBNK SETTING 101
# OCT121        OCT     121             # SERVICE ROUTINES
                                        # (FIRST 8K OF ACM)
SUPER110        OCT     140             # BITS FOR SUPERBNK SETTING 110.
                                        # (LAST 8K OF ACM)
1SEC            DEC     100
# LOW7          OCT     177             # INTERPRETER
# BIT8          OCT     00200
# OT215         OCT     215             # ALARM AND ABORT
# 8,5           OCT     00220           # P20-P25 SUNDANCE
2SECS           DEC     200
# LOW8          OCT     377             # PINBALL
# BIT9          OCT     00400
GN/CCODE        OCT     00401           # SET S/C CONTROL SWITCH TO G/N
3SECS           DEC     300
4SECS           DEC     400
LOW9            OCT     777
# BIT10         OCT     01000
# 5.5DEGS       DEC     .03056          # P20-P25 SUNDANCE    (OCTAL 00765)
# OCT1103       OCT     1103            # ALARM AND ABORT
C5/2            DEC     .0363551        #                     (OCTAL 01124)
V05N09          VN      0509            # (SAME AS OCTAL 1211)
OCT1400         OCT     01400
V06N22          VN      0622
```

```
# MID5          OCT     1740            # PINBALL
BITS2-10        OCT     1776
LOW10           OCT     1777
# BIT11         OCT     02000
# 2K+3          OCT     2003            # PINBALL
LOW7+2K         OCT     2177            # OP CODE MASK + BANK 1 FBANK SETTING.
EBANK5          OCT     02400
PRIO3           OCT     03000
EBANK7          OCT     03400
# LOW11         OCT     3777            # PINBALL
# BIT12         OCT     04000
# RELTAB        OCT     04025           # T4RUPT
PRIO5           OCT     05000
PRIO6           OCT     06000
PRIO7           OCT     07000

# BIT13         OCT     10000
#               OCT     10003           # T4RUPT     RELTAB +1D
# 13,7,2        OCT     10102           # P20-P25 SUNDANCE
PRIO11          OCT     11000
# PRIO12        OCT     12000           # BANKCALL
PRIO13          OCT     13000
PRIO14          OCT     14000
#               OCT     14031           # T4RUPT     RELTAB +2D
PRIO15          OCT     15000
PRIO16          OCT     16000
# 85DEGS        DEC     .45556          # P20-P25 SUNDANCE   (OCTAL 16450)
PRIO17          OCT     17000
OCT17770        OCT     17770
# BIT14         OCT     20000
#               OCT     20033           # T4RUPT     RELTAB +3D
PRIO21          OCT     21000
                BLOCK   03
                COUNT*  $$/FCONS
PRIO22          OCT     22000           # SERVICE ROUTINES
PRIO23          OCT     23000
PRIO24          OCT     24000
# 5/8+1         OCT     24001           # SINGLE PRECISION SUBROUTINES
#               OCT     24017           # T4RUPT     RELTAB +4D
PRIO25          OCT     25000
PRIO26          OCT     26000
PRIO27          OCT     27000
# CHRPRIO       OCT     30000           # PINBALL
#               OCT     30036           # T4RUPT     RELTAB +5D
PRIO31          OCT     31000
C1/2            DEC     .7853134        #                    (OCTAL 31103)
PRIO32          OCT     32000
PRIO33          OCT     33000
PRIO34          OCT     34000
#               OCT     34034           # T4RUPT     RELTAB +6D
```

```
PRIO35          OCT     35000
PRIO36          OCT     36000
PRIO37          OCT     37000
63/64+1         OCT     37401
# MID7          OCT     37600           # PINBALL
OCT37766        OCT     37766
OCT37774        OCT     37774
OCT37776        OCT     37776
# DPOSMAX       OCT     37777
# BIT15         OCT     40000
# OCT40001      OCT     40001           # INTERPRETER  ( CS   1   INSTRUCTION)
DLOADCOD        OCT     40014
DLOAD*          OCT     40015
#               OCT     40023           # T4RUPT     RELTAB +7D
BIT15+6         OCT     40040
OCT40200        OCT     40200
#               OCT     44035           # T4RUPT    RELTAB +8D
#               OCT     50037           # T4RUPT     RELTAB +9D
#               OCT     54000           # T4RUPT     RELTAB +10D
-BIT14          OCT     57777
# RELTAB11      OCT     60000           # T4RUPT
C3/2            DEC     -.3216147       #                     (OCTAL 65552)
13,14,15        OCT     70000
-1/8            OCT     73777
HIGH4           OCT     74000
-ENDERAS        DEC     -2001           #                     (OCTAL 74056)
# HI5           OCT     76000           # PINBALL
HIGH9           OCT     77700
# -ENDVAC       DEC     -45             # INTERPRETER          (OCTAL 77722)
# -OCT10        OCT     -10             #                      (OCT 77767)
# NEG4          DEC     -4              #                     (OCTAL 77773)
NEG3            DEC     -3
NEG2            OCT     77775
NEGONE          DEC     -1
```

```
# DEFINED BY EQUALS

# IT WOULD BE TO THE USERS ADVANTAGE TO OCCASIONALLY CHECK ANY OF THESE SYMBOLS IN ORDER TO PREVENT ANY
# ACCIDENTAL DEFINITION CHANGES.

MINUS1          =       NEG1
NEG1            =       NEGONE
ONE             =       BIT1
TWO             =       BIT2
THREE           =       OCTAL3
LOW2            =       THREE
FOUR            =       BIT3
SIX             =       REVCNT
LOW3            =       SEVEN
EIGHT           =       BIT4
NINE            =       R2D1
TEN             =       BINCON
NOUTCON         =       ELEVEN
OCT23           =       VD1
OCT25           =       MD1
PRIO1           =       BIT10
EBANK3          =       OCT1400
PRIO2           =       BIT11
OCT120          =       SUPER101
OCT140          =       SUPER110
2K              =       BIT11
EBANK4          =       BIT11
PRIO4           =       BIT12
EBANK6          =       PRIO3
QUARTER         =       BIT13
PRIO10          =       BIT13
OCT10001        =       CCSL
POS1/2          =       HALF
PRIO20          =       BIT14
HALF            =       BIT14
PRIO30          =       CHRPRIO
BIT13-14        =       PRIO30          # INTERPRETER USES IN PROCESSING STORECODE
OCT30002        =       TLOAD +1
B12T14          =       PRIO34
NEGMAX          =       BIT15
VLOADCOD        =       BIT15
VLOAD*          =       OCT40001
OCT60000        =       RELTAB11
BANKMASK        =       HI5
```

```
                SETLOC  INTPRET1
                BANK

                COUNT*  $$/ICONS
DP1/4TH         2DEC    .25

UNITZ           2DEC    0

UNITY           2DEC    0

UNITX           2DEC    .5

ZEROVECS        2DEC    0

                2DEC    0

                2DEC    0

DPHALF          =       UNITX
DPPOSMAX        OCT     37777
                OCT     37777
```

```
# INTERPRETIVE CONSTANTS IN THE OTHER HALF-MEMORY

                SETLOC  INTPRET2
                BANK

                COUNT*  $$/ICONS
ZUNIT           2DEC    0

YUNIT           2DEC    0

XUNIT           2DEC    .5

ZEROVEC         2DEC    0

                2DEC    0

                2DEC    0

                OCT     77777           # -0, -6, -12 MUST REMAIN IN THIS ORDER
DFC-6           DEC     -6
DFC-12          DEC     -12
LODPMAX         2OCT    3777737777      # THESE TWO CONSTANTS MUST REMAIN

LODPMAX1        2OCT    3777737777      # ADJACENT AND THE SAME FOR INTEGRATION

ZERODP          =       ZEROVEC
HALFDP          =       XUNIT
```

```
                BLOCK   02

# SINGLE PRECISION SINE AND COSINE

                COUNT*  $$/INTER
SPCOS           AD      HALF            # ARGUMENTS SCALED AT PI
SPSIN           TS      TEMK
                TCF     SPT
                CS      TEMK
SPT             DOUBLE
                TS      TEMK
                TCF     POLLEY
                XCH     TEMK
                INDEX   TEMK
                AD      LIMITS
                COM
                AD      TEMK
                TS      TEMK
                TCF     POLLEY
                TCF     ARG90
POLLEY          EXTEND
                MP      TEMK
                TS      SQ
                EXTEND
                MP      C5/2
                AD      C3/2
                EXTEND
                MP      SQ
                AD      C1/2
                EXTEND
                MP      TEMK
                DDOUBL
                TS      TEMK
                TC      Q
ARG90           INDEX   A
                CS      LIMITS
                TC      Q               # RESULT SCALED AT 1
```

```
                BLOCK   02

# TO ENTER A JOB REQUEST REQUIRING NO VAC AREA:

                COUNT*  $$/EXEC
NOVAC           INHINT
                AD      FAKEPRET        # LOC(MPAC +6) - LOC(QPRET)
                TS      NEWPRIO         # PRIORITY OF NEW JOB + NOVAC C(FIXLOC)

                EXTEND
                INDEX   Q               # Q WILL BE UNDISTURBED THROUGHOUT.
                DCA     0               # 2CADR OF JOB ENTERED.
                DXCH    NEWLOC
                CAF     EXECBANK
                XCH     FBANK
                TS      EXECTEM1
                TCF     NOVAC2          # ENTER EXECUTIVE BANK.

# TO ENTER A JOB REQUEST REQUIREING A VAC AREA -- E.G., ALL (PARTIALLY) INTERPRETIVE JOBS.

FINDVAC         INHINT
                TS      NEWPRIO
                EXTEND
                INDEX   Q
                DCA     0
SPVACIN         DXCH    NEWLOC
                CAF     EXECBANK
                XCH     FBANK
                TCF     FINDVAC2        # OFF TO EXECUTIVE SWITCHED-BANK.

# TO ENTER A FINDVAC WITH THE PRIORITY IN NEWPRIO TO THE 2CADR ARRIVING IN A AND L:
# USERS OF SPVAC MUST INHINT BEFORE STORING IN NEWPRIO.

SPVAC           XCH     Q
                AD      NEG2
                XCH     Q
                TCF     SPVACIN

# TO SUSPEND A BASIC JOB SO A HIGHER PRIORITY JOB MAY BE SERVICED:

CHANG1          LXCH    Q
                CAF     EXECBANK
                XCH     BBANK
                TCF     CHANJOB

# TO SUSPEND AN INTERPRETIVE JOB:

CHANG2          CS      LOC             # NEGATIVE LOC SHOWS JOB = INTERPRETIVE.
# ITRACE (4) REFERS TO "CHANG2"
                TS      L
```

```
+2              CAF     EXECBANK
                TS      BBANK
                TCF     CHANJOB -1
```

```
# TO VOLUNTARILY SUSPEND A JOB UNTIL THE COMPLETION OF SOME ANTICIPATED EVENT (I/O EVENT ETC.):

JOBSLEEP        TS      LOC
                CAF     EXECBANK
                TS      FBANK
                TCF     JOBSLP1

# TO AWAKEN A JOB PUT TO SLEEP IN THE ABOVE FASHION:

JOBWAKE         INHINT
                TS      NEWLOC
                CS      TWO             # EXIT IS VIA FINDVAC/NOVAC PROCEDURES.
                ADS     Q
                CAF     EXECBANK
                XCH     FBANK
                TCF     JOBWAKE2

# TO CHANGE THE PRIORITY OF A JOB CURRENTLY UNDER EXECUTION:

PRIOCHNG        INHINT                  # NEW PRIORITY ARRIVES IN A.  RETURNS TO
                TS      NEWPRIO         # CALLER AS SOON AS NEW JOB PRIORITY IS
                CAF     EXECBANK        # HIGHEST.  PREPARE FOR POSSIBLE BASIC-
                XCH     BBANK           # STYLE CHANGE-JOB.
                TS      BANKSET
                CA      Q
                TCF     PRIOCH2

# TO REMOVE A JOB FROM EXECUTIVE CONSIDERATIONS:

ENDOFJOB        CAF     EXECBANK
                TS      FBANK
                TCF     ENDJOB1

ENDFIND         CA      EXECTEM1        # RETURN TO CALLER AFTER JOB ENTRY
                TS      FBANK           # COMPLETE.
                TCF     Q+2
EXECBANK        CADR    FINDVAC2

FAKEPRET        ADRES   MPAC -36D       # LOC(MPAC +6) - LOC(QPRET)
```

```
# LOCATE AN AVAILABLE VAC AREA

                BANK    01
                COUNT*  $$/EXEC
FINDVAC2        TS      EXECTEM1        # (SAVE CALLER'S BANK FIRST.)
                CCS     VAC1USE
                TCF     VACFOUND
                CCS     VAC2USE
                TCF     VACFOUND
                CCS     VAC3USE
                TCF     VACFOUND
                CCS     VAC4USE
                TCF     VACFOUND
                CCS     VAC5USE
                TCF     VACFOUND
                LXCH    EXECTEM1
                CA      Q
                TC      BAILOUT1
                OCT     1201            # NO VAC AREAS.

VACFOUND        AD      TWO             # RESERVE THIS VAC AREA BY STORING A ZERO
                ZL                      # IN ITS VAC USE REGISTER AND STORE THE
                INDEX   A               # ADDRESS OF THE FIRST WORD OF IT IN THE
                LXCH    0       -1      # LOW NINE BITS OF THE PRIORITY WORD.
                ADS     NEWPRIO

NOVAC2          CAF     ZERO            # NOVAC ENTERS HERE.  FIND A CORE SET.
                TS      LOCCTR
                CAF     NO.CORES        # SEVEN SETS OF ELEVEN REGISTERS EACH.
NOVAC3          TS      EXECTEM2
                INDEX   LOCCTR
                CCS     PRIORITY        # EACH PRIORITY REGISTER CONTAINS -0 IF
                TCF     NEXTCORE        # THE CORESPONDING CORE SET IS AVAILABLE.
NO.CORES        DEC     7
                TCF     NEXTCORE        # AN ACTIVE JOB HAS A POSITIVE PRIORITY
                                        # BUT A DORMANT JOB'S PRIORITY IS NEGATIVE
```

```
CORFOUND        CA      NEWPRIO         # SET THE PRIORITY OF THIS JOB IN THE CORE
                INDEX   LOCCTR          # SET'S PRIORITY REGISTER AND SET THE
                TS      PRIORITY        # JOB'S PUSH-DOWN POINTER AT THE BEGINNING
                MASK    LOW9            # OF THE WORK AREA AND OVERFLOW INDICATOR.
                INDEX   LOCCTR
                TS      PUSHLOC         # OFF TO PREPARE FOR INTERPRETIVE PROGRAMS.

                CCS     LOCCTR          # IF CORE SET ZERO IS BEING LOADED, SET UP
                TCF     SETLOC          # OVFIND AND FIXLOC IMMEDIATELY.
                TS      OVFIND
                CA      PUSHLOC
                TS      FIXLOC

SPECTEST        CCS     NEWJOB          # SEE IF ANY ACTIVE JOBS WAITING (RARE).
                TCF     SETLOC          # MUST BE AWAKENED OUT UNCHANGED JOB.
                TC      CCSHOLE
                TC      CCSHOLE
                TS      NEWJOB          # +0 SHOWS ACTIVE JOB ALREADY SET.
                DXCH    NEWLOC
                DXCH    LOC
                TCF     ENDFIND

SETLOC          DXCH    NEWLOC          # SET UP THE LOCATION REGISTERS FOR THIS
                INDEX   LOCCTR
                DXCH    LOC
                INDEX   NEWJOB          # THIS INDEX INSTRUCTION INSURES THAT THE
                CS      PRIORITY        # HIGHEST ACTIVE PRIORITY WILL BE COMPARED
                AD      NEWPRIO         # WITH THE NEW PRIORITY TO SEE IF NEWJOB
                EXTEND                  # SHOULD BE SET TO SIGNAL A SWITCH.
                BZMF    ENDFIND
                CA      LOCCTR          # LOCCTR IS LEFT SET AT THIS CORE SET IF
                TS      NEWJOB          # THE CALLER WANTS TO LOAD ANY MPAC
                TCF     ENDFIND         # REGISTERS, ETC.

NEXTCORE        CAF     COREINC
                ADS     LOCCTR
                CCS     EXECTEM2
                TCF     NOVAC3
                LXCH    EXECTEM1
                CA      Q
                TC      BAILOUT1        # NO CORE SETS AVAILABLE.
                OCT     1202
```

```
# THE FOLLOWING ROUTINE SWAPS CORE SET 0 WITH THAT WHOSE RELATIVE ADDRESS IS IN NEWJOB.

 -2             LXCH    LOC
 -1             CAE     BANKSET         # BANKSET, NOT BBANK, HAS RIGHT CONTENTS.
CHANJOB         INHINT
                EXTEND
                ROR     SUPERBNK        # PICK UP CURRENT SBANK FOR BBCON
                XCH     L               # LOC IN A AND BBCON IN L.
 +4             INDEX   NEWJOB          # SWAP LOC AND BANKSET.
                DXCH    LOC
                DXCH    LOC

                CAE     BANKSET
                EXTEND
                WRITE   SUPERBNK        # SET SBANK FOR NEW JOB.
                DXCH    MPAC            # SWAP MULTI-PURPOSE ACCUMULATOR AREAS.
                INDEX   NEWJOB
                DXCH    MPAC
                DXCH    MPAC
                DXCH    MPAC    +2
                INDEX   NEWJOB
                DXCH    MPAC    +2
                DXCH    MPAC    +2
                DXCH    MPAC    +4
                INDEX   NEWJOB
                DXCH    MPAC    +4
                DXCH    MPAC    +4
                DXCH    MPAC    +6
                INDEX   NEWJOB
                DXCH    MPAC    +6
                DXCH    MPAC    +6

                CAF     ZERO
                XCH     OVFIND          # MAKE PUSHLOC NEGATIVE IF OVFIND NZ.
                EXTEND
                BZF     +3
                CS      PUSHLOC
                TS      PUSHLOC

                DXCH    PUSHLOC
                INDEX   NEWJOB
                DXCH    PUSHLOC
                DXCH    PUSHLOC         # SWAPS PUSHLOC AND PRIORITY.
                CAF     LOW9            # SET FIXLOC TO BASE OF VAC AREA.
                MASK    PRIORITY
                TS      FIXLOC

                CCS     PUSHLOC         # SET OVERFLOW INDICATOR ACCORDING TO
                CAF     ZERO
                TCF     ENDPRCHG -1
```

```
                CS      PUSHLOC
                TS      PUSHLOC
                CAF     ONE
                XCH     OVFIND
                TS      NEWJOB

ENDPRCHG        RELINT
                DXCH    LOC             # BASIC JOBS HAVE POSITIVE ADDRESSES, SO
                EXTEND                  # DISPATCH WITH A DTCB.
                BZMF    +2              # IF INTERPRETIVE, SET UP EBANK, ETC.
                DTCB
```

```
                COM                         # EPILOGUE TO JOB CHANGE FOR INTERPRETIVE
                AD      ONE
                TS      LOC                 # RESUME
                TCF     INTRSM

# COMPLETE JOBSLEEP PREPARATIONS.

JOBSLP1         INHINT
                CS      PRIORITY            # NNZ PRIORITY SHOWS JOB ASLEEP.
                TS      PRIORITY
                CAF     LOW7
                MASK    BBANK
                EXTEND
                ROR     SUPERBNK            # SAVE OLD SUPERBANK VALUE.
                TS      BANKSET
                CS      ZERO
JOBSLP2         TS      BUF     +1          # HOLDS -- HIGHEST PRIORITY.
                TCF     EJSCAN              # SCAN FOR HIGHEST PRIORITY ALA ENDOFJOB.

NUCHANG2        INHINT                      # QUICK... DON'T LET NEWJOB CHANGE TO +0.
                CCS     NEWJOB
                TCF     +3                  # NEWJOB STILL PNZ
                RELINT                      # NEW JOB HAS CHANGED TO +0.  WAKE UP JOB
                TCF     ADVAN   +2          # VIA NUDIRECT.  (VERY RARE CASE.)

                CAF     TWO
                EXTEND
                WOR     DSALMOUT            # TURN ON ACTIVITY LIGHT
                DXCH    LOC                 # AND SAVE ADDRESS INFO FOR BENEFIT OF
                TCF     CHANJOB +4          #       POSSIBLE SLEEPINT JOB.
```

```
# TO WAKE UP A JOB, EACH CORE SET IS FOUND TO LOCATE ALL JOBS WHICH ARE ASLEEP.  IF THE FCADR IN THE
# LOC REGISTER OF ANY SUCH JOB MATCHES THAT SUPPLIED BY THE CALLER, THAT JOB IS AWAKENED.  IF NO JOB IS FOUND,
# LOCCTR IS SET TO -1 AND NO FURTHER ACTION TAKES PLACE.

JOBWAKE2        TS      EXECTEM1
                CAF     ZERO            # BEGIN CORE SET SCAN
                TS      LOCCTR
                CAF     NO.CORES
JOBWAKE4        TS      EXECTEM2
                INDEX   LOCCTR
                CCS     PRIORITY
                TCF     JOBWAKE3        # ACTIVE JOB -- CHECK NEXT CORE SET.
COREINC         DEC     12              # 12 REGISTERS PER CORE SET.
                TCF     WAKETEST        # SLEEPING JOB -- SEE IF CADR MATCHES.

JOBWAKE3        CAF     COREINC
                ADS     LOCCTR
                CCS     EXECTEM2
                TCF     JOBWAKE4
                CS      ONE             # EXIT IF SLEEPIN JOB NOT FOUND.
                TS      LOCCTR
                TCF     ENDFIND

WAKETEST        CS      NEWLOC
                INDEX   LOCCTR
                AD      LOC
                EXTEND
                BZF     +2              # IF MATCH.
                TCF     JOBWAKE3        # EXAMINE NEXT CORE SET IF NO MATCH.

                INDEX   LOCCTR          # RE-COMPLEMENT PRIORITY TO SHOW JOB AWAKE
                CS      PRIORITY
                TS      NEWPRIO
                INDEX   LOCCTR
                TS      PRIORITY

                CS      FBANKMSK        # MAKE UP THE 2CADR OF THE WAKE ADDRESS
                MASK    NEWLOC          # USING THE CADR IN NEWLOC AND THE EBANK
                AD      2K              # HALF OF BBANK SAVED IN BANKSET.
                XCH     NEWLOC
                MASK    FBANKMSK
                INDEX   LOCCTR
                AD      BANKSET
                TS      NEWLOC +1

                CCS     LOCCTR          # SPECIAL TREATMENT IF THIS JOB WAS
                TCF     SETLOC          # ALREADY IN THE RUN (0) POSITION.
                TCF     SPECTEST
```

```
# PRIORITY CHANGE.  CHANGE THE CONTENTS OF PRIORITY AND SCAN FOR THE JOB OF HIGHEST PRIORITY.

PRIOCH2         TS      LOC
                CAF     ZERO            # SET FLAG TO TELL ENDJOB SCANNER IF THIS
                TS      BUF             # JOB IS STILL HIGHEST PRIORITY.
                CAF     LOW9
                MASK    PRIORITY
                AD      NEWPRIO
                TS      PRIORITY
                COM
                TCF     JOBSLP2         # AND TO EJSCAN.
```

```
# RELEASE THIS CORE SET AND VAC AREA AND SCAN FOR THE JOB OF HIGHEST ACTIVE PRIORITY.

ENDJOB1         INHINT
                CS      ZERO
                TS      BUF     +1
                XCH     PRIORITY
                MASK    LOW9
                TS      L

                CS      FAKEPRET
                AD      L

                EXTEND
                BZMF    EJSCAN          # NOVAC ENDOFJOB

                CCS     L
                INDEX   A
                TS      0

EJSCAN          CCS     PRIORITY +12D
                TC      EJ1
                TC      CCSHOLE
                TCF     +1

                CCS     PRIORITY +24D   # EXAMINE EACH PRIORITY REGISTER TO FIND
                TC      EJ1             # THE JOB OF HIGHEST ACTIVE PRIORITY.
                TC      CCSHOLE
                TCF     +1

                CCS     PRIORITY +36D
                TC      EJ1
-CCSPR          -CCS    PRIORITY
                TCF     +1

                CCS     PRIORITY +48D
                TC      EJ1
                TC      CCSHOLE
                TCF     +1

                CCS     PRIORITY +60D
                TC      EJ1
                TC      CCSHOLE
                TCF     +1

                CCS     PRIORITY +72D
                TC      EJ1
                TC      CCSHOLE
                TCF     +1

                CCS     PRIORITY +84D
```

```
          TC      EJ1
          TC      CCSHOLE
          TCF     +1
```

```
# EVALUATE THE RESULTS OF THE SCAN.

                CCS     BUF     +1      # SEE IF THERE ARE ANY ACTIVE JOBS WAITING
                TC      CCSHOLE
                TC      CCSHOLE

                TCF     +2
                TCF     DUMMYJOB
                CCS     BUF             # BUF IS ZERO IS THIS IS A PRIOCHNG AND
                TCF     +2              # CHANGED PRIORITY IS STILL HIGHEST.
                TCF     ENDPRCHG -1

                INDEX   A               # OTHERWISE, SET NEWJOB TO THE RELATIVE
                CAF     0       -1      # ADDRESS OF THE NEW JOB'S CORE SET.
                AD      -CCSPR
                TS      NEWJOB
                TCF     CHANJOB -2

EJ1             TS      BUF     +2
                AD      BUF     +1      # - OLD HIGH PRIORITY.
                CCS     A
                CS      BUF     +2
                TCF     EJ2             # NEW HIGH PRIORITY.
                NOOP
                INDEX   Q
                TC      2               # PROCEED WITH SEARCH.

EJ2             TS      BUF     +1
                EXTEND
                QXCH    BUF             # FOR LOCATING CCS PRIORITY + X INSTR.
                INDEX   BUF
                TC      2
```

```
# IDLING AND COMPUTER ACTIVITY (GREEN) LIGHT MAINTENANCE. THE IDLING ROUTINE IS NOT A JOB IN ITSELF,
# BUT RATHER A SUBROUTINE OF THE EXECUTIVE.

                EBANK=  SELFRET         # SELF-CHECK STORAGE IN EBANK.

DUMMYJOB        CS      ZERO            # SET NEWJOB TO -0 FOR IDLING.
                TS      NEWJOB
                RELINT
                CS      TWO             # TURN OFF THE ACTIVITY LIGHT.
                EXTEND
                WAND    DSALMOUT
ADVAN           CCS     NEWJOB          # IS THE NEWJOB ACTIVE?
                TCF     NUCHANG2        # YES... ONE REQUIRING A CHANGE JOB.
                CAF     TWO             # NEW JOB ALREADY IN POSITION FOR
                TCF     NUDIRECT        # EXECUTION

                CA      SELFRET
                TS      L               # PUT RETURN ADDRESS IN L.
                CAF     SELFBANK
                TCF     SUPDXCHZ +1     # AND DISPATCH JOB.

                EBANK=  SELFRET
SELFBANK        BBCON   SELFCHK

NUDIRECT        EXTEND                  # TURN THE GREEN LIGHT BACK ON.
                WOR     DSALMOUT
                DXCH    LOC             # JOBS STARTED IN THIS FASHION MUST BE
                TCF     SUPDXCHZ

                BLOCK   2               # IN FIXED-FIXED SO OTHERS MAY USE.

                COUNT*  $$/EXEC

# SUPDXCHZ -- ROUTINE TO TRANSFER TO SUPEBANK.
# CALLING SEQUENCE:
#               TCF     SUPDXCHZ        # WITH 2CADR OF DESIRED LOCATION IN A + L.

SUPDXCHZ        XCH     L               # BASIC.
 +1             EXTEND
                WRITE   SUPERBNK
                TS      BBANK
                TC      L

NEG100          OCT     77677
```

```
# PROGRAM DESCRIPTION                                                       DATE -- 10 OCTOBER 1966
# MOD NO -- 2                                                               LOG SECTION -- WAITLIST
# MOD BY -- MILLER       (DTMAX INCREASED TO 162.5 SEC)                     ASSEMBLY -- SUNBURST REV 5
# MOD 3 BY KERNAN        (INHINT INSERTED AT WAITLIST) 2/28/68 SKIPPER REV 4
# MOD 4 BY KERNAN        (TWIDDLE IN 54) 3/28/68 SKIPPER REV 13.
#
# FUNCTIONAL DESCRIPTION --
#       PART OF A SECTION OF PROGRAMS -- WAITLIST, TASKOVER, T3RUPT, USED TO CALL A PROGRAM (CALLED A TASK),
#       WHICH IS TO BEGIN IN C(A) CENTISECONDS.  WAITLIST UPDATES TIME3, LST1, AND LST2.  THE MEANING OF THESE LISTS
#       FOLLOW.
#
#               C(TIME3) = 16384 -(T1-T) CENTISECONDS, (T=PRESENT TIME, T1-TIME FOR TASK1)
#
#                       C(LST1)         =       -(T2-T1)+1
#                       C(LST1 +1)      =       -(T3-T2)+1
#                       C(LST1 +2)      =       -(T4-T3)+1
#                                      ...
#                       C(LST1 +6)      =       -(T8-T7)+1
#                       C(LST1 +7)      =       -(T9-T8)+1
#
#                       C(LST2)         =       2CADR OF TASK1
#                       C(LST2 +2)      =       2CADR OF TASK2
#                                      ...
#                       C(LST2 +14)     =       2CADR OF TASK8
#                       C(LST2 +16)     =       2CADR OF TASK9
#
# WARNINGS --
#       1)      1 <= C(A) <= 16250D (1 CENTISECOND TO 162.5 SEC)
#       2)      9 TASKS MAXIMUM
#       3)      TASKS CALLED UNDER INTERRUPT INHIBITED
#       4)      TASKS END BY TC TASKOVER
#
# CALLING SEQUENCE --
#       L-1     CA      DELTAT  (TIME IN CENTISECONDS TO TASK START)
#       L       TC      WAITLIST
#       L+1     2CADR   DESIRED TASK.
#       L+2     (MINOR OF 2CADR)
#       L+3     RELINT          (RETURNS HERE)
#
# TWIDDLE --
#       TWIDDLE IS FOR USE WHEN THE TASK BEING SET UP IS IN THE SAME EBANK AND FBANK AS THE USER.  IN
#       SUCH CASES, IT IMPROVES UPON WAITLIST BY ELIMINATING THE NEED FOR THE BBCON HALF OF THE 2CADR,
```

```
#       SAVING A WORD.  TWIDDLE IS LIKE WAITLIST IN EVERY RESPECT EXCEPT CALLING SEQUENCE, TO WIT,
#               L-1     CA      DELTAT
#               L       TC      TWIDDLE
#               L+1     ADRES   DESIRED TASK
#               L+2     RELINT          (RETURNS HERE)
#
# NORMAL EXIT MODES --
#       AT L+3 OF CALLING SEQUENCE.
#
# ALARM OR ABORT EXIT MODES --
#       TC      ABORT
#       OCT     1203    (WAITLIST OVERFLOW -- TOO MANY TASKS)
#
# ERASABLE INITIALIZATION REQUIRED --
#       ACCOMPLISHED BY FRESH START --  LST2, ..., LST2 +16 = ENDTASK
#                                       LST1, ..., LST1 +7  = NEG1/2
#
# OUTPUT --
#       LST1 AND LST2 UPDATED WTIH NEW TASK AND ASSOCIATED TIME.
#
# DEBRIS --
#       CENTRALS -- A,Q,L
#       OTHER    -- WAITEXIT, WAITADR, WAITTEMP, WAITBANK
#
# DETAILED ANALYSIS OF TIMING --
#       CONTROL WILL NOT BE RETURNED TO THE SPECIFIED ADDRESS (2CADR) IN EXACTLY DELTA T CENTISECONDS.
#       THE APPROXIMATE TIME MAY BE CALCULATED AS FOLLOWS:
#               LET TO = THE TIME OF THE TC WAITLIST
#               LET TS = TO +147U + COUNTER INCREMENTS (SET UP TIME)
#               LET X  = TS -(100TS)/100  (VARIANCE FROM COUNTERS)
#               LET Y  = LENGTH OF TIME OF INHIBIT INTERRUPT AFTER T3RUPT
#               LET Z  = LENGTH OF TIME TO PROCESS TASKS WHICH ARE DUE THIS T3RUPT BUT DISPATCHED EARLIER.
#                       (Z=0, USUALLY).
#               LET DELTD  = THE ACTUAL TIME TAKEN TO GIVE CONTROL TO 2CADR
#               THEN DELTD = TS+DELTA T -X +Y +Z +1.05MS* +COUNTERS*
#               *THE TIME TAKEN BY WAITLIST ITSELF AND THE COUNTER TICKING DURING THIS WAITLIST TIME.
#       IN SHORT, THE ACTUAL TIME TO RETURN CONTROL TO A 2CADR IS AUGMENTED BY THE TIME TO SET UP THE TASK'S
#       INTERRUPT, ALL COUNTERS TICKING, THE T3RUPT PROCESSING TIME, THE WAITLIST PROCESSING TIME AND THE POSSIBILITY
#       OF OTHER TASKS INHIBITING THE INTERRUPT.

                BLOCK   02
```

```
                EBANK=  LST1            # TASK LISTS IN SWITCHED E BANK.

                COUNT*  $$/WAIT
TWIDDLE         INHINT
                TS      L               # SAVE DELAY TIME IN L
                CA      POSMAX
                ADS     Q               # CREATING OVERFLOW AND Q-1 IN Q
                CA      BBANK
                EXTEND
                ROR     SUPERBNK
                XCH     L

WAITLIST        INHINT
                XCH     Q               # SAVE DELTA T IN Q AND RETURN IN
                TS      WAITEXIT        # WAITEXIT.
                EXTEND
                INDEX   WAITEXIT        # IF TWIDDLING, THE TS SKIPS TO HERE
                DCA     0               # PICK UP 2CADR OF TASK.
 -1             TS      WAITADR         # BBCON WILL REMAIN IN L
DLY2            CAF     WAITBB          # ENTRY FROM FIXDELAY AND VARDELAY.
                XCH     BBANK
                TCF     WAIT2

# RETURN TO CALLER AFTER TASK INSERTION:

LVWTLIST        DXCH    WAITEXIT
                AD      TWO
                DTCB

                EBANK=  LST1
WAITBB          BBCON   WAIT2

# RETURN TO CALLER +2 AFTER WAITING DT SPECIFIED AT CALLER +1.

FIXDELAY        INDEX   Q               # BOTH ROUTINES MUST BE CALLED UNDER
                CAF     0               # WAITLIST CONTROL AND TERMINATE THE TASK
                INCR    Q               # IN WHICH THEY WERE CALLED.

# RETURN TO CALLER +1 AFTER WAITING THE DT AS ARRIVING IN A.

VARDELAY        XCH     Q               # DT TO Q.  TASK ADRES TO WAITADR.
                TS      WAITADR
                CA      BBANK           # BBANK IS SAVED DURING DELAY.
                EXTEND
                ROR     SUPERBNK        # ADD SBANK TO BBCON.
                TS      L
                CAF     DELAYEX
                TS      WAITEXIT        # GO TO TASKOVER AFTER TASK ENTRY.
                TCF     DLY2
```

```
DELAYEX         TCF     TASKOVER -2     # RETURNS TO TASKOVER.
```

```
# ENDTASK MUST ENTERED IN FIXED-FIXED SO IT IS DISTINGUISHABLE BY ITS ADRES ALONE.

                EBANK=  LST1
ENDTASK         -2CADR  SVCT3

SVCT3           CCS     FLAGWRD2        # DRIFT FLAG
                TCF     TASKOVER
                TCF     TASKOVER
                TCF     +1

CKIMUSE         CCS     IMUCADR         # DON'T DO NBDONLY IF SOMEONE ELSE IS IN
                TCF     SVCT3X          # IMUSTALL.
                TCF     +3
                TCF     SVCT3X
                TCF     SVCT3X

 +3             CAF     PRIO35          # COMPENSATE FOR NBD COEFFICIENTS ONLY.
                TC      NOVAC           #       ENABLE EVERY 81.93 SECONDS
                EBANK=  NBDX
                2CADR   NBDONLY

                TCF     TASKOVER

SVCT3X          TC      FIXDELAY        # DELAY MAX OF 2 TIMES FOR IMUZERO.
                DEC     500
                TC      SVCT3           # CHECK DRIFT FLAG AGAIN.
```

```
# BEGIN TASK INSERTION.

                                BANK    01
                                COUNT*  $$/WAIT
WAIT2                           TS      WAITBANK        # BBANK OF CALLING PROGRAM.
                                CA      Q
                                EXTEND
                                BZMF    WAITPOOH

                                CS      TIME3
                                AD      BIT8            # BIT 8 = OCT 200
                                CCS     A               # TEST 200 - C(TIME3).  IF POSITIVE,
                                                        # IT MEANS THAT TIME3 OVERFLOW HAS OCCURRED PRIOR TO CS TIME3 AND THAT
                                                        # C(TIME3) = T - T1, INSTEAD OF 1.0 - (T1 - T).  THE FOLLOWING FOUR
                                                        # ORDERS SET C(A) = TD - T1 + 1 IN EITHER CASE.

                                AD      OCT40001        # OVERFLOW HAS OCCURRED.  SET C(A) =
                                CS      A               # T - T1 + 1.0 - 201

# NORMAL CASE (C(A) NNZ) YIELDS SAME C(A):  -( -(1.0-(T1-T)) + 200) - 1

                                AD      OCT40201
                                AD      Q               # RESULT = TD - T1 + 1.

                                CCS     A               # TEST TD - T1 + 1.

                                AD      LST1            # IF TD - T1 POS, GO TO WTLST5 WITH
                                TCF     WTLST5          # C(A) = (TD - T1) + C(LST1) = TD-T2+1

                                NOOP
                                CS      Q

# NOTE THAT THIS PROGRAM SECTION IS NEVER ENTERED WHEN T-T1 G/E -1,
# SINCE TD-T1+1 = (TD-T) + (T-T1+1), AND DELTA T = TD-T G/E +1.  (G/E
# SYMBOL MEANS GREATER THAN OR EQUAL TO).  THUS THERE NEED BE NO CON-
# CERN OVER A PREVIOUS OR IMMINENT OVEFLOW OF TIME3 HERE.

                                AD      POS1/2          # WHEN TD IS NEXT, FORM QUANTITY
                                AD      POS1/2          #       1.0 - DELTA T = 1.0 - (TD - T)
                                XCH     TIME3
                                AD      NEGMAX
                                AD      Q               # 1.0 - DELTAT T NOW COMPLETE.
                                EXTEND                  # ZERO INDEX Q.
                                QXCH    7               # (ZQ)
```

```
WTLST4          XCH     LST1
                XCH     LST1    +1
                XCH     LST1    +2
                XCH     LST1    +3
                XCH     LST1    +4
                XCH     LST1    +5
                XCH     LST1    +6
                XCH     LST1    +7

                CA      WAITADR         # (MINOR PART OF TASK CADR HAS BEEN IN L.)
                INDEX   Q
                TCF     +1

                DXCH    LST2
                DXCH    LST2    +2
                DXCH    LST2    +4
                DXCH    LST2    +6
                DXCH    LST2    +8D
                DXCH    LST2    +10D    # AT END, CHECK THAT C(LST2 +10) IS STD
                DXCH    LST2    +12D
                DXCH    LST2    +14D
                DXCH    LST2    +16D
                AD      ENDTASK         # END ITEM, AS CHECK FOR EXCEEDING
                                        # THE LENGTH OF THE LIST.
                EXTEND                  # DUMMY TASK ADRES SHOULD BE IN FIXED-
                BZF     LVWTLIST        # FIXED SO ITS ADRES ALONE DISTINGUISHES
                TCF     WTABORT         # IT.
```

```
WTLST5          CCS     A               # TEST TD - T2 + 1
                AD      LST1    +1
                TCF     +4
                AD      ONE
                TC      WTLST2
                OCT     1

 +4             CCS     A               # TEST TD - T3 + 1
                AD      LST1    +2
                TCF     +4
                AD      ONE
                TC      WTLST2
                OCT     2

 +4             CCS     A               # TEST TD - T4 + 1
                AD      LST1    +3
                TCF     +4
                AD      ONE
                TC      WTLST2
                OCT     3

 +4             CCS     A               # TEST TD - T5 + 1
                AD      LST1    +4
                TCF     +4
                AD      ONE
                TC      WTLST2
                OCT     4

 +4             CCS     A               # TEST TD - T6 + 1
                AD      LST1    +5
                TCF     +4
                AD      ONE
                TC      WTLST2
                OCT     5

 +4             CCS     A               # TEST TD - T7 + 1
                AD      LST1    +6
                TCF     +4
                AD      ONE
                TC      WTLST2
                OCT     6
```

```
 +4             CCS     A               # TEST TD - T2 + 1
                AD      LST1    +7
                TCF     +4
                AD      ONE
                TC      WTLST2
                OCT     7

 +4             CCS     A
WTABORT         TC      FILLED
                NOOP                    # CAN'T GET HERE
                AD      ONE
                TC      WTLST2
                OCT     10

OCT40201        OCT     40201
```

```
FILLED          DXCH    WAITEXIT
                TC      BAILOUT1        # NO ROOM IN THE INN
                OCT     01203
```

```
# THE ENTRY TC WTLST2 JUST PRECEDING OCT N IS FOR T  LE TD LE T    -1.
#                                                N            N+1
# (LE MEANS LESS THAN OR EQUAL TO).  AT ENTRY, C(A) = -(TD - T    + 1)
#                                                          N+1
# THE LST1 ENTRY -(T    -T +1) IS TO BE REPLACED BY -(TD - T  + 1), AND
#                   N+1  N                                 N
# THE ENTRY -(T    - TD + 1) IS TO BE INSERTED IMMEDIATELY FOLLOWING.
#              N+1

WTLST2          TS      WAITTEMP        # C(A) = -(TD - T + 1)
                INDEX   Q
                CAF     0
                TS      Q               # INDEX VALUE INTO Q.

                CAF     ONE
                AD      WAITTEMP
                INDEX   Q               # C(A) = -(TD - T ) + 1.
                ADS     LST1    -1      #                N

                CS      WAITTEMP
                INDEX   Q
                TCF     WTLST4

#       C(TIME3)        =       1.0 - (T1 - T)
#
#       C(LST1)         =       - (T2 - T1) + 1
#       C(LST1+1)       =       - (T3 - T2) + 1
#       C(LST1+2)       =       - (T4 - T3) + 1
#       C(LST1+3)       =       - (T5 - T4) + 1
#       C(LST1+4)       =       - (T6 - T5) + 1
#
#       C(LST2)         =       2CADR TASK1
#       C(LST2+2)       =       2CADR TASK2
#       C(LST2+4)       =       2CADR TASK3
#       C(LST2+6)       =       2CADR TASK4
#       C(LST2+8)       =       2CADR TASK5
#       C(LST2+10)      =       2CADR TASK6
```

```
# ENTERS HERE ON T3 RUPT TO DISPATCH WAITLISTED TASK.

T3RUPT          EXTEND
                ROR     SUPERBNK        # READ CURRENT SUPERBANK VALUE AND
                TS      BANKRUPT        # SAVE WITH E AND F BANK VALUES.
                EXTEND
                QXCH    QRUPT

T3RUPT2         CAF     NEG1/2          # DISPATCH WAITLIST TASK.
                XCH     LST1    +7
                XCH     LST1    +6
                XCH     LST1    +5
                XCH     LST1    +4      # 1. MOVE UP LST1 CONTENTS, ENTERING
                XCH     LST1    +3      #    A VALUE OF 1/2 +1 AT THE BOTTOM
                XCH     LST1    +2      #    FOR T6-T5, CORRESPONDING TO THE
                XCH     LST1    +1      #    INTERVAL 81.91 SEC FOR ENDTASK.
                XCH     LST1
                AD      POSMAX          # 2. SET T3 = 1.0 - T2 - T USING LIST 1.
                ADS     TIME3           #    SO T3 WON'T TICK DURING UPDATE.
                TS      RUPTAGN
                CS      ZERO
                TS      RUPTAGN         # SETS RUPTAGN TO +1 ON OVERFLOW.

                EXTEND                  # DISPATCH TASK.
                DCS     ENDTASK
                DXCH    LST2    +16D
                DXCH    LST2    +14D
                DXCH    LST2    +12D
                DXCH    LST2    +10D
                DXCH    LST2    +8D
                DXCH    LST2    +6
                DXCH    LST2    +4
                DXCH    LST2    +2
                DXCH    LST2

                XCH     L
                EXTEND
                WRITE   SUPERBNK        # SET SUPERBANK FROM BBCON OF 2CADR
                XCH     L               # RESTORE TO L FOR DXCH Z.
                DTCB
```

```
# RETURN, AFTER EXECUTION OF T3 OVERFLOW TASK:

                BLOCK   02
                COUNT*  $$/WAIT
TASKOVER        CCS     RUPTAGN         # IF +1 RETURN TO T3RUPT, IF -0 RESUME.
                CAF     WAITBB
                TS      BBANK
                TCF     T3RUPT2         # DISPATCH NEXT TASK IF IT WAS DUE.

                CA      BANKRUPT
                EXTEND
                WRITE   SUPERBNK        # RESTORE SUPERBANK BEFORE RESUME IS DONE

RESUME          EXTEND
                QXCH    QRUPT
NOQRSM          CA      BANKRUPT
                XCH     BBANK
NOQBRSM         DXCH    ARUPT
                RELINT
                RESUME
```

```
# LONGCALL
# PROGRAM DESCRIPTION                                  DATE -- 17 MARCH 1967
# PROGRAM WRITTEN BY W.H.VANDEVER                      LOG SECTION WAITLIST
# MOD BY -- R. MELANSON TO ADD DOCUMENTATION           ASSEMBLY SUNDISK REV. 100
#
# FUNCTIONAL DESCRIPTION --
#       LONGCALL IS CALLED WITH THE DELTA TIME ARRIVING IN A,L SCALED AS TIME2,TIME1 WITH THE 2CADR OF THE TASK
#       IMMEDIATELY FOLLOWING THE TC LONGCALL.  FOR EXAMPLE, IT MIGHT BE DONE AS FOLLOWS WHERE TIMELOC IS THE NAME OF
#       A DP REGISTER CONTAINING A DELTA TIME AND WHERE TASKTODO IS THE NAME OF THE LOCATION AT WHICH LONGCALL IS TO
#       START.
# CALLING SEQUENCE --
#               EXTEND
#               DCA     TIMELOC
#               TC      LONGCALL
#               2CADR   TASKTODO
# NORMAL EXIT MODE --
#       1)      TC      WAITLIST
#       2)      DTCB    (TC L+3 OF CALLING ROUTINE 1ST PASS THRU LONGCYCL)
#       3)      DTCB    (TO TASKOVER ON SUBSEQUENT PASSES THRU LONGCYCL)
# ALARM OR ABORT EXIT MODE --
#       NONE
# OUTPUT --
#       LONGTIME AND LONGTIME+1 = DELTA TIME
#       LONGEXIT AND LONGEXIT+1 = RETURN 2CADR
#       LONGCADR AND LONGCADR+1 = TASK 2CADR
#       A = SINGLE PRECISION TIME FOR WAITLIST
# ERASABLE INITIALIZATION --
#       A = MOST SIGNIFICANT PART OF DELTA TIME
#       L = LEAST SIGNIFICANT PART OF DELTA TIME
#       Q = ADDRESS OF 2CADR TASK VALUE
# DEBRIS --
#       A,Q,L
#       LONGCADR AND LONGCADR+1
#       LONGEXIT AND LONGEXIT+1
#       LONGTIME AND LONGTIME+1
# *** THE FOLLOWING IS TO BE IN FIXED-FIXED AND UNSWITCHED ERRASIBLE **

                BLOCK   02
                EBANK=  LST1
LONGCALL        DXCH    LONGTIME        # OBTAIN THE DELTA TIME

                EXTEND                  # OBTAIN THE 2CADR
```

```
                NDX     Q
                DCA     0
                DXCH    LONGCADR

                EXTEND                  # NOW GO TO THE APPROPRIATE SWITCHED BANK
                DCA     LGCL2CDR        # FOR THE REST OF LONGCALL
                DTCB

                EBANK=  LST1
LGCL2CDR        2CADR   LNGCALL2

# *** THE FOLLOWING MAY BE IN A SWITCHED BANK, INCLUDING ITS ERASABLE ***

                BANK    01
                COUNT*  $$/WAIT
LNGCALL2        LXCH    LONGEXIT +1     # SAVE THE CORRECT BB FOR RETURN
                CA      TWO             # OBTAIN THE RETURN ADDRESS
                ADS     Q
                TS      LONGEXIT

                CA      LONGTIME        # CHECK FOR LEGITIMATE DELTA-TIME
                CCS     A
                TCF     LONGCYCL        # HI-ORDER OK --> ALL IS OK.
                TCF     +2              # HI-ORDER ZERO --> CHECK LO-ORDER.
                TCF     LONGPOOH        # HI-ORDER NEG. --> NEG. DT
 +2             CA      LONGTIME +1     # CHECK LO-ORDER FOR ZERO OR NEGATIVE.
                EXTEND
                BZMF    LONGPOOH        # BAD DELTA-TIME.  ABORT

# *** WAITLIST TASK LONGCYCL ***

LONGCYCL        EXTEND                  # CAN WE SUCCESFULLY TAKE ABOUT 1.25
                DCS     DPBIT14         # MINUTES OFF OF LONGTIME
                DAS     LONGTIME

                CCS     LONGTIME +1     # THE REASONING BEHIND THIS PART IS
                TCF     MUCHTIME        # INVOLVED, TAKING INTO ACCOUNT THAT THE
                                        # WORDS MAY NOT BE SIGNED CORRECTED (DP
                                        # BASIC INSTRUCTIONS
                                        # DO NOT SIGN CORRECT) AND THAT WE SUBTRAC-
                                        # TED BIT14 (1 OVER HALF THE POS. VALUE
                                        # REPRESENTABLE IN SINGLE WORD)
                NOOP                    # CAN'T GET HERE *************
                TCF     +1
                CCS     LONGTIME
                TCF     MUCHTIME
DPBIT14         OCT     00000
                OCT     20000

                                        # LONGCALL
```

```
LASTTIME        CA      BIT14           # GET BACK THE CORRECT DELTA T FOR WAITLIST
                ADS     LONGTIME +1
                TC      WAITLIST
                EBANK=  LST1
                2CADR   GETCADR         # THE ENTRY TO OUR LONGCADR

LONGRTRN        CA      TSKOVCDR        # SET IT UP SO THAT ONLY THE FIRST EXIT IS
                DXCH    LONGEXIT        # TO THE CALLER OF LONGCALL
                DTCB                    # THE REST ARE TO TASKOVER

MUCHTIME        CA      BIT14           # WE HAVE OVER OUR ABOUT 1.25 MINUTES
                TC      WAITLIST        # SO SET UP FOR ANOTHER CYCLE THROUGH HERE
                EBANK=  LST1
                2CADR   LONGCYCL

                TCF     LONGRTRN        # NOW EXIT PROPERLY

# *** WAITLIST TASK GETCADR ***

GETCADR         DXCH    LONGCADR        # GET THE LONGCALL THAT WE WISHED TO START
                DTCB                    # AND TRANSFER CONTROL TO IT

TSKOVCDR        GENADR  TASKOVER
LONGPOOH        DXCH    LONGEXIT
                TCF     +2
WAITPOOH        DXCH    WAITEXIT
 +2             TC      POODOO1
                OCT     01204
```

```
# SUBROUTINE TO CONVERT RAD VECTOR AT GIVEN TIME TO LAT,LONG AND ALT
#
# CALLING SEQUENCE
#
#   L-1     CALL
#   L              LAT-LONG
# SUBROUTINES USED
#
#  R-TO-RP,ARCTAN,SETGAMMA,SETRE
# ERASABLE INIT. REQ.
#
#   AXO,-AYO,AZO,TEPHEM (SET AT LAUNCH TIME)
# ALPHAV = POSITION VECTOR METERS B-29
#   MPAC-- TIME  (CSECS B-28)
#  ERADFLAG =1, TO COMPUTE EARTH RADIUS, =0 FOR FIXED EARTH RADIUS
#  LUNAFLAG=0 FOR EARTH,1 FOR MOON
# OUTPUT
#
#   LATITUDE IN LAT   (REVS. B-0)
#   LONGITUDE IN LONG   (REVS. B-0)
# ALTITUDE IN ALT METERS B-29
                BANK    30
                SETLOC  LATLONG
                BANK

                COUNT*  $$/LT-LG
                EBANK=  ALPHAV
LAT-LONG        STQ     SETPD
                        INCORPEX
                        OD
                STOVL   6D              # SAVE TIME IN 6-7D FOR R-TO-RP
                        ALPHAV
                PUSH    ABVAL           # O-5D= R FOR R-TO-RP
                STODL   ALPHAM          # ABS. VALUE OF R FOR ALT FORMULA BELOW
                        ZEROVEC         # SET MPAC=0 FOR EARTH,NON-ZERO FOR MOON
                BOFF    COS             # USE COS(0) TO GET NON-ZERO IN MPAC
                        LUNAFLAG        # 0=EARTH,1=MOON
                        CALLRTRP
CALLRTRP        CALL
                        R-TO-RP         # RP VECTOR CONVERTED FROM R B-29
                UNIT                    # UNIT RP B-1
                STCALL  ALPHAV          # U2= 1/2 SINL FOR SETRE SUBR BELOW
                        SETGAMMA        #  SET GAMMA=B2/A2 FOR EARTH,=1 FOR MOON
                CALL                    #  SCALED B-1
                        SETRE           # CALC RE METERS B-29
                DLOAD   DSQ
                        ALPHAV
                PDDL    DSQ
                        ALPHAV +2
                DAD     SQRT
```

```
                DMP     SL1R
                        GAMRP
                STODL   COSTH           # COS(LAT) B-1
                        ALPHAV +4
                STCALL  SINTH           # SIN(LAT) B-1
                        ARCTAN
                STODL   LAT             # LAT BO
                        ALPHAV
                STODL   COSTH           # COS(LONG) B-1
                        ALPHAV +2
                STCALL  SINTH           # SIN(LONG) B-1
                        ARCTAN
                STODL   LONG            # LONG. REVS B-0 IN RANGE -1/2 TO 1/2
                        ALPHAM
                DSU                     # ALT= R-RE METERS B-29
                        ERADM
                STCALL  ALT             # EXIT WITH ALT METERS B-29
                        INCORPEX
```

```
# SUBROUTINE TO CONVERT LAT,LONG,ALT AT GIVEN TIME TO RADIUS VECTOR
# CALLING SEQUENCE
#
#   L-1    CALL
#   L              LALOTORV
# SUBROUTINES USED
#
#  SETGAMMA,SETRE,RP-TO-R
# ERASABLE INIT. REQ.
#
#   AXO,AYO,AZO,TEPHEM  SET AT LAUNCH TIME
#   LAT-- LATITUDE  (REVS B0)
#   LONG-- LONGITUDE  (REVS B0)
# ALT--ALTITUDE (METERS) B-29
#   MPAC-- TIME  (CSECS B-28)
#   ERADFLAG =1 TO COMPUTE EARTH RADIUS, =0 FOR FIXED EARTH RADIUS
#  LUNAFLAG=0 FOR EARTH,1 FOR MOON
# OUTPUT
#
# R-VECTOR IN ALPHAV (METERS B-29)
LALOTORV        STQ     SETPD           # LAT,LONG,ALT TO R VECTOR
                        INCORPEX
                        0D
                STCALL  6D              # 6-7D= TIME FOR RP-TO-R
                        SETGAMMA        # GAMMA=B2/A2 FOR EARTH,1 FOR MOON B-1
                DLOAD   SIN             #              COS(LONG)COS(LAT) IN MPAC
                        LAT             #     UNIT RP=  SIN(LONG)COS(LAT)    2-3D
                DMPR    PDDL            # PD 2          GAMMA*SIN(LAT)       0-1D
                        GAMRP
                        LAT             #         0-1D= GAMMA*SIN(LAT) B-2
                COS     PDDL            # PD4      2-3D=COS(LAT) B-1 TEMPORARILY
                        LONG
                SIN     DMPR            # PD 2
                PDDL    COS             # PD 4    2-3D=SIN(LONG)COS(LAT) B-2
                        LAT
                PDDL    COS             # PD 6     4-5D=COS(LAT) B-1 TEMPORARILY
                        LONG
                DMPR    VDEF            # PD 4   MPAC=  COS(LONG)COS(LAT) B-2
                UNIT    PUSH            # 0-5D= UNIT RP FOR RP-TO-R SUBR.
                STCALL  ALPHAV          # ALPHAV +4= SINL FOR SETRE SUBR.
                        SETRE           # RE METERS B-29
                DLOAD   BOFF            # SET MPAC=0 FOR EARTH,NON-ZERO FOR MOON
                        ZEROVEC
                        LUNAFLAG
                        CALLRPRT
                COS                     # USE COS(0) TO GET NON-ZERO IN MPAC
CALLRPRT        CALL
                        RP-TO-R         # EXIT WITH UNIT R VECTOR IN MPAC
                STODL   ALPHAV
                        ERADM
```

```
                DAD     VXSC            # (RE + ALT)(UNIT R) METERS B-30
                        ALT
                        ALPHAV
                VSL1                    # R METERS B-29
                STCALL  ALPHAV          # EXIT WITH R IN METERS B-29
                        INCORPEX
# SUBROUTINE TO COMPUTE EARTH RADIUS
#
# INPUT
#
#   1/2 SIN LAT IN ALPHAV +4
#
# OUTPUT
#
#   EARTH RADIUS IN ERADM AND MPAC (METERS B-29)

GETERAD         DLOAD   DSQ
                        ALPHAV +4       # SIN**2(L)
                SL1     BDSU
                        DP1/2           # COS**2(L)
                DMPR    BDSU
                        EE
                        DP1/2
                BDDV    SQRT
                        B2XSC
                SR4R
                STORE   ERADM
                RVQ

# THE FOLLOWING CONSTANTS WERE COMPUTED WITH A=6378166,B=6356784 METERS
# B2XSC= B**2 SCALED B-51
# B2/A2= B**2/A**2 SCALED B-1
# EE=(1-B**2/A**2) SCALED B-0
B2XSC           2DEC    .0179450689     # B**2 SCALED B-51
DP1/2           =       XUNIT
B2/A2           2DEC    .9933064884 B-1 # GAMMA= B**2/A**2 B-1
EE              2DEC    6.6935116 E-3   # (1-B**2/A**2) B-0
```

```
# ARCTAN SUBROUTINE
#
# CALLING SEQUENCE
#
#    SIN THETA IN SINTH B-1
#    COS THETA IN COSTH B-1
#    CALL ARCTAN
#
# OUTPUT
#    ARCTAN THETA IN MPAC AND THETA B-0 IN RANGE -1/2 TO +1/2

ARCTAN          BOV
                        CLROVFLW
CLROVFLW        DLOAD   DSQ
                        SINTH
                PDDL    DSQ
                        COSTH
                DAD
                BZE     SQRT
                        ARCTANXX        # ATAN=0/0  SET THETA=0
                BDDV    BOV
                        SINTH
                        ATAN=90
                SR1     ASIN
                STORE   THETA
                PDDL    BMN
                        COSTH
                        NEGCOS
                DLOAD   RVQ
NEGCOS          DLOAD   DCOMP
                BPL     DAD
                        NEGOUT
                        DP1/2
ARCTANXX        STORE   THETA
                RVQ

NEGOUT          DSU     GOTO
                        DP1/2
                        ARCTANXX
ATAN=90         DLOAD   SIGN
                        LODP1/4
                        SINTH
                STORE   THETA
                RVQ

2DZERO          =       DPZERO
```

```
# ..... SETGAMMA SUBROUTINE .....
# SUBROUTINE TO SET GAMMA FOR THE LAT-LONG AND LALOTORV SUBROUTINES
#
# GAMMA = B**2/A**2 FOR EARTH (B-1)
# GAMMA = 1 FOR MOON (B-1)
#
# CALLING SEQUENCE
#  L        CALL
#  L+1            SETGAMMA
#
# INPUT
#  LUNAFLAG=0 FOR EARTH,=1 FOR MOON
#
# OUTPUT
#  GAMMA IN GAMRP  (B-1)

SETGAMMA        DLOAD   BOFF            # BRANCH FOR EARTH
                        B2/A2           # EARTH GAMMA
                        LUNAFLAG
                        SETGMEX
                SLOAD
                        1B1             # MOON GAMMA
SETGMEX         STORE   GAMRP
                RVQ
GAMRP           =       8D
```

```
# .....SETRE SUBROUTINE .....
# SUBROUTINE TO SET RE (EARTH OR MOON RADIUS)
#
# RE= RM FOR MOON
#  RE= RREF FOR FIXED EARTH RADIUS OR COMPUTED RF FOR FISCHER ELLIPSOID
#
# CALLING SEQUENCE
#  L        CALL
#  L+1             SETRE
#
# SUBROUTINES USED
#  GETERAD
#
# INPUT
#  ERADFLAG=0 FOR FIXED RE, 1 FOR COMPUTED RE
#  ALPHAV +4= 1/2 SINL IF GETERAD IS CALLED
#  LUNAFLAG=0 FOR EARTH,=1 FOR MOON
#
# OUTPUT
#  ERADM= 504RM FOR MOON (METERS B-29)
#  ERADM= ERAD OR COMPUTED RE FOR EARTH (METERS B-29)

SETRE           STQ     DLOAD
                        SETREX
                        504RM
                BON     DLOAD           # BRANCH FOR MOON
                        LUNAFLAG
                        TSTRLSRM
                        ERAD
                BOFF    CALL            # ERADFLAG=0 FOR FIXED RE,1 FOR COMPUTED
                        ERADFLAG
                        SETRXX
                        GETERAD
SETRXX          STCALL  ERADM           # EXIT WITH RE OR RM METERS B-29
                        SETREX
TSTRLSRM        BON     VLOAD           # ERADFLAG=0,SET RO=RLS
                        ERADFLAG        #          =1     RO=RM
                        SETRXX
                        RLS
                ABVAL   SR2R            # SCALE FROM B-27 TO B-29
                GOTO
                        SETRXX
SETREX          =       S2
```

```
# ..... RP-TO-R SUBROUTINE .....
# SUBROUTINE TO CONVERT RP (VECTOR IN PLANETARY COORDINATE SYSTEM,EITHER
#  EARTH-FIXED OR MOON-FIXED) TO R (SAME VECTOR IN THE BASIC REF. SYSTEM)
#
#  R=MT(T)*(RP+LPXRP)    MT= M MATRIX TRANSPOSE
#
# CALLING SEQUENCE
#  L        CALL
#  L+1             RP-TO-R
#
# SUBROUTINES USED
#  EARTHMX,MOONMX,EARTHL
#
#    ITEMS AVAILABLE FROM LAUNCH DATA
#     504LM= THE LIBRATION VECTOR L OF THE MOON AT TIME TIMSUBL,EXPRESSED
#     IN THE MOON-FIXED COORD. SYSTEM   RADIANS  B0
#       ITEMS NECESSARY FOR SUBR. USED (SEE DESCRIPTION OF SUBR.)
#
# INPUT
#  MPAC= 0 FOR EARTH,NON-ZERO FOR MOON
#  0-5D= RP VECTOR
#  6-7D= TIME
#
# OUTPUT
#  MPAC= R VECTOR METERS B-29 FOR EARTH, B-27 FOR MOON

                SETLOC  PLANTIN1
                BANK

                COUNT*  $$/LUROT

RP-TO-R         STQ     BHIZ
                        RPREXIT
                        RPTORA
                CALL                    # COMPUTE M MATRIX FOR MOON
                        MOONMX          # LP=LM FOR MOON  RADIANS B0
                VLOAD
                        504LM
RPTORB          VXV     VAD
                        504RPR
                        504RPR
                VXM     GOTO
                        MMATRIX         # MPAC=R=MT(T)*(RP+LPXRP)
                        RPRPXXXX        # RESET PUSHLOC TO 0 BEFORE EXITING
RPTORA          CALL                    # EARTH COMPUTATIONS
                        EARTHMX         # M MATRIX B-1
                CALL
                        EARTHL          # L VECTOR RADIANS B0
                MXV     VSL1            # LP=M(T)*L  RAD B-0
                        MMATRIX
```

```
                GOTO
                        RPTORB
                SETLOC  PLANTIN
                BANK
                COUNT*  $$/LUROT
```

```
# ..... R-TO-RP SUBROUTINE .....
# SUBROUTINE TO CONVERT R (VECTOR IN REFERENCE COORD. SYSTEM) TO RP
#  (VECTOR IN PLANETARY COORD SYSTEM) EITHER EARTH-FIXED OR MOON-FIXED
#
#  RP=M(T)*(R-LXR)
#
# CALLING SEQUENCE
#  L        CALL
#  L+1             R-TO-RP
#
# SUBROUTINES USED
#  EARTHMX,MOONMX,EARTHL
#
# INPUT
#  MPAC= 0 FOR EARTH, NON-ZERO FOR MOON
#  0-5D= R VECTOR
#  6-7D= TIME
#
#    ITEMS AVAILABLE FROM LAUNCH DATA
#     504LM= THE LIBRATION VECTOR L OF THE MOON AT TIME TIMSUBL,EXPRESSED
#     IN THE MOON-FIXED COORD. SYSTEM   RADIANS B0
#       ITEMS NECESSARY FOR SUBROUTINES USED (SEE DESCRIPTION OF SUBR.)
#
# OUTPUT
#  MPAC=RP VECTOR METERS B-29 FOR EARTH, B-27 FOR MOON

R-TO-RP         STQ     BHIZ
                        RPREXIT
                        RTORPA
                CALL
                        MOONMX
                VLOAD   VXM
                        504LM           # LP=LM
                        MMATRIX
                VSL1                    #  L=MT(T)*LP  RADIANS B0
RTORPB          VXV     BVSU
                        504RPR
                        504RPR
                MXV                     # M(T)*(R-LXR) B-2
                        MMATRIX
RPRPXXXX        VSL1    SETPD
                        0D
                GOTO
                        RPREXIT
RTORPA          CALL                    # EARTH COMPUTATIONS
                        EARTHMX
                CALL
                        EARTHL
                GOTO                    # MPAC=L=(-AX,-AY,0) RAD B-0
                        RTORPB
```

```
# ..... MOONMX SUBROUTINE .....
# SUBROUTINE TO COMPUTE THE TRANSFORMATION MATRIX M FOR THE MOON
#
# CALLING SEQUENCE
#  L        CALL
#  L+1            MOONMX
#
# SUBROUTINES USED
#  NEWANGLE
#
# INPUT
#  6-7D= TIME
#    ITEMS AVAILABLE FROM LAUNCH DATA
#     BSUBO,BDOT
#     TIMSUBO,NODIO,NODDOT,FSUBO,FDOT
#     COSI= COS(I) B-1
#     SINI= SIN(I) B-1
#       I  IS THE ANGLE BETWEEN THE MEAN LUNAR EQUATORIAL PLANE AND THE
#       PLANE OF THE ECLIPTIC  (1 DEGREE  32.1 MINUTES)
#
# OUTPUT
#  MMATRIX= 3X3 M MATRIX B-1   (STORED IN VAC AREA)

MOONMX          STQ     SETPD
                        EARTHMXX
                        8D
                AXT,1                   # B REQUIRES SL 0, SL 5 IN NEWANGLE
                        5
                DLOAD   PDDL            # PD 10D            8-9D=BSUBO
                        BSUBO           #                   10-11D= BDOT
                        BDOT
                PUSH    CALL            # PD 12D
                        NEWANGLE        # EXIT WITH PD 8D AND MPAC= B  REVS BO
                PUSH    COS             # PD 10D
                STODL   COB             # PD 8D    COS(B) B-1
                SIN                     #          SIN(B) B-1
                STODL   SOB             #           SETUP INPUT FOR NEWANGLE
                        FSUBO           #                     8-9D=FSUBO
                PDDL    PUSH            # PD 10D THEN 12D   10-11D=FDOT
                        FDOT
                AXT,1   CALL            # F REQUIRES SL 1, SL 6 IN NEWANGLE
                        4
                        NEWANGLE        # EXIT WITH PD 8D AND MPAC= F REVS BO
                STODL   AVECTR +2       # SAVE F TEMP
                        NODIO           #                       8-9D=NODIO
                PDDL    PUSH            # PD 10D THEN 12D     10-11D=NODDOT
                        NODDOT          #                       MPAC=T
                AXT,1   CALL            # NODE REQUIRES SL 0, SL 5 IN NEWANGLE
                        5
                        NEWANGLE        # EXIT WITH PD 8D AND MPAC= NODI REVS BO
```

```
                PUSH    COS             # PD 10D   8-9D= NODI  REVS  B0
                PUSH                    # PD 12D 10-11D= COS(NODI)  B-1
                STORE   AVECTR
                DMP     SL1R
                        COB             #                          COS(NODI)    B-1
                STODL   BVECTR +2       # PD 10D   20-25D=AVECTR= COB*SIN(NODI)
                DMP     SL1R            #                          SOB*SIN(NODI)
                        SOB
                STODL   BVECTR +4       # PD 8D
                SIN     PUSH            # PD 10D                  -SIN(NODI)   B-1
                DCOMP                   #          26-31D=BVECTR= COB*COS(NODI)
                STODL   BVECTR          # PD 8D                    SOB*COS(NODI)
                        AVECTR +2       # MOVE F FROM TEMP LOC. TO 504F
                STODL   504F
                DMP     SL1R
                        COB
                GOTO
                        MOONMXA

                BANK    25
                SETLOC  PLANTIN3
                BANK
                COUNT*  $$/LUROT

MOONMXA         STODL   AVECTR +2
                        SINNODI         # 8-9D=SIN(NODI)  B-1
                DMP     SL1R
                        SOB
                STODL   AVECTR +4       #                           0
                        HI6ZEROS        #         8-13D= CVECTR= -SOB  B-1
                PDDL    DCOMP           # PD 10D                   COB
                        SOB
                PDDL    PDVL            # PD 12D THEN PD 14D
                        COB
                        BVECTR
                VXSC    PDVL            # PD 20D         BVECTR*SINI  B-2
                        SINI
                        CVECTR
                VXSC    VAD             # PD 14D         CVECTR*COSI  B-2
                        COSI
                VSL1
                STOVL   MMATRIX +12D    # PD 8D  M2=BVECTR*SINI+CVECTR*COSI  B-1
                VXSC    PDVL            # PD 14D
                        SINI            #                CVECTR*SINI  B-2
                        BVECTR
                VXSC    VSU             # PD 8D          BVECTR*COSI  B-2
                        COSI
                VSL1    PDDL            # PD 14D
                        504F            # 8-13D=DVECTR=BVECTR*COSI-CVECTR*SINI B-1
                COS     VXSC
```

```
                        DVECTR
                PDDL    SIN             # PD 20D  14-19D= DVECTR*COSF  B-2
                        504F
                VXSC    VSU             # PD 14D          AVECTR*SINF  B-2
                        AVECTR
                VSL1
                STODL   MMATRIX +6      # M1= AVECTR*SINF-DVECTR*COSF  B-1
                        504F
                SIN     VXSC            # PD 8D
                PDDL    COS             # PD 14D  8-13D=DVECTR*SINF B-2
                        504F
                VXSC    VAD             # PD 8D         AVECTR*COSF B-2
                        AVECTR
                VSL1    VCOMP
                STCALL  MMATRIX         # M0= -(AVECTR*COSF+DVECTR*SINF)  B-1
                        EARTHMXX
# COMPUTE X=X0+(XDOT)(T+T0)
# 8-9D= X0 (REVS B-0),PUSHLOC SET AT 12D
# 10-11D=XDOT (REVS/CSEC) SCALED B+23 FOR WEARTH,B+28 FOR NODDOT AND BDOT
#                        AND B+27 FOR FDOT
#  X1=DIFFERENCE IN 23 AND SCALING OF XDOT,=0 FOR WEARTH,5 FOR NODDOT AND
#                                         BDOT AND 4 FOR FDOT
# 6-7D=T (CSEC B-28), TIMSUBO= (CSEC B-42 TRIPLE PREC.)

NEWANGLE        DLOAD   SR              # ENTER PD 12D
                        6D
                        14D
                TAD     TLOAD           # CHANGE MODE TO TP
                        TIMSUBO
                        MPAC
                STODL   TIMSUBM         # T+T0 CSEC B-42
                        TIMSUBM +1
                DMP                     # PD 10D  MULT BY XDOT IN 10-11D
                SL*     DAD             # PD 8D   ADD X0 IN 8-9D AFTER SHIFTING
                        5,1             #         SUCH THAT SCALING IS B-0
                PUSH    SLOAD           # PD 10D SAVE PARTIAL (X0+XDOT*T) IN 8-9D
                        TIMSUBM
                SL      DMP
                        9D
                        10D             # XDOT
                SL*     DAD             # PD 8D   SHIFT SUCH THAT THIS PART OF X
                        10D,1           #         IS SCALED REVS/CSEC B-0
                BOV                     # TURN OFF OVERFLOW IF SET BY SHIFT
                        +1              # INSTRUCTION BEFORE EXITING
                RVQ                     # MPAC=X= X0+(XDOT)(T+T0)  REVS B0
```

```
# ..... EARTHMX SUBROUTINE .....
# SUBROUTINE TO COMPUTE THE TRANSFORMATION MATRIX M FOR THE EARTH
#
# CALLING SEQUENCE
#  L        CALL
#  L+1            EARTHMX
#
# SUBROUTINE USED
#  NEWANGLE
#
# INPUT
#    INPUT AVAILABLE FROM LAUNCH DATA     AZO  REVS B-0
#                                         TEPHEM  CSEC B-42
#  6-7D= TIME CSEC B-28
#
# OUTPUT
#  MMATRIX= 3X3 M MATRIX B-1   (STORED IN VAC AREA)

                BANK    26
                SETLOC  PLANTIN1
                BANK
                COUNT*  $$/LUROT

EARTHMX         STQ     SETPD           # SET   8-9D=AZO
                        EARTHMXX
                        8D              # 10-11D=WEARTH
                AXT,1                   # FOR SL 5, AND SL 10  IN NEWANGLE
                        0
                DLOAD   PDDL            #   LEAVING PD SET AT 12D FOR NEWANGLE
                        AZO
                        WEARTH
                PUSH    CALL
                        NEWANGLE
                SETPD   PUSH            # 18-19D=504AZ
                        18D             #                     COS(AZ) SIN(AZ) 0
                COS     PDDL            # 20-37D=  MMATRIX= -SIN(AZ) COS(AZ) 0 B-1
                        504AZ           #                      0       0      1
                SIN     PDDL
                        HI6ZEROS
                PDDL    SIN
                        504AZ
                DCOMP   PDDL
                        504AZ
                COS     PDVL
                        HI6ZEROS
                PDDL    PUSH
                        HIDPHALF
                GOTO
                        EARTHMXX
```

```
# ..... EARTHL SUBROUTINE .....
# SUBROUTINE TO COMPUTE L VECTOR FOR EARTH
#
# CALLING SEQUENCE
#  L        CALL
#  L+1            EARTHL
#
# INPUT
#  AXO,AYO SET AT LAUNCH TIME WITH AYO IMMEDIATELY FOLLOWING AXO IN CORE
#
# OUTPUT
#           -AX
#   MPAC=   -AY    RADIANS B-0
#             0

                BANK    06
                SETLOC  EARTHLOC
                BANK
                COUNT*  $$/LUROT

EARTHL          DLOAD   DCOMP
                        AXO
                STODL   504LPL
                        -AYO
                STODL   504LPL +2
                        LO6ZEROS
                STOVL   504LPL +4
                        504LPL
                RVQ
```

```
# CONSTANTS AND ERASABLE ASSIGNMENTS
1B1             =       DP1/2           # 1  SCALED B-1
RPREXIT         =       S1              # R-TO-RP AND RP-TO-R SUBR EXIT
EARTHMXX        =       S2              # EARTHMX,MOONMX SUBR. EXITS
504RPR          =       0D              # 6 REGS  R OR RP VECTOR
SINNODI         =       8D              # 2       SIN(NODI)
DVECTR          =       8D              # 6       D VECTOR MOON
CVECTR          =       8D              # 6       C VECTR MOON
504AZ           =       18D             # 2      AZ
TIMSUBM         =       14D             # 3       TIME SUB M (MOON) T+TO IN GETAZ
504LPL          =       14D             # 6       L OR LP VECTOR
AVECTR          =       20D             # 6       A VECTOR (MOON)
BVECTR          =       26D             # 6       B VECTOR (MOON)
MMATRIX         =       20D             # 18      M MATRIX
COB             =       32D             # 2       COS(B) B-1
SOB             =       34D             # 2       SIN(B) B-1
504F            =       6D              # 2       F (MOON)
```

```
# INCORP1 -- PERFORMS THE SIX DIMENSIONAL STATE VECTOR DEVIATION FOR POSITION
# AND VELOCITY OR THE NINE-DIMENSIONAL DEVIATION OF POSITION, VELOCITY, AND
# RADAR OR LANDMARK BIAS. THE OUTPUT OF THE BVECTOR ROUTINE ALONG WITH THE
# ERROR TRANSITION MATRIX (W) ARE USED AS INPU TO THE ROUTINE. THE DEVIATION
# IS OBTAINED BY COMPUTING AN ESTIMATED TRACKING MEASUREMENT FROM THE
# CURRENT STATE VECTOR AND COMPARING IT WITH AN ACTUAL TRACKING MEASUREMENT
# AND APPLYING A STATISTICAL WEIGHTING VECTOR.
#
# INPUT
#        DMENFLG = 0 (6-DIMENSIONAL BVECTOR), =1 (9-DIMENSIONAL)
#              W = ERROR TRANSITION MATRIX 6X6 OR 9X9
#       VARIANCE = VARIANCE (SCALAR)
#         DELTAQ = MEASURED DEVIATION (SCALAR)
#        BVECTOR = 6 OR 9 DIMENSIONAL BVECTOR
#
# OUTPUT
#         DELTAX = STATE VECTOR DEVIATIONS 6 OR 9 DIMENSIONAL
#             ZI = VECTOR USED FOR THE INCORPORATION 6 OR 9 DIMENSIONAL
#          GAMMA = SCALAR
#          OMEGA = OMEGA WEIGHTING VECTOR 6 OR 9 DIMENSIONAL
#
# CALLING SEQUENCE
#       L       CALL    INCORP1
#
# NORMAL EXIT
#       L+1 OF CALLING SEQUENCE

                BANK    37
                SETLOC  MEASINC
                BANK

                COUNT*  $$/INCOR

                EBANK=  W

INCORP1         STQ
                        EGRESS
                AXT,1   SSP
                        54D
                        S1
                        18D             # IX1 = 54      S1= 18
                AXT,2   SSP
                        18D
                        S2
                        6               # IX2 = 18      S2=6
Z123            VLOAD   MXV*
                        BVECTOR         # BVECTOR (0)
                        W +54D,1
                STORE   ZI +18D,2
                VLOAD
                        BVECTOR +6      # BVECTOR (1)
```

```
                MXV*     VAD*
                         W +108D,1
                         ZI +18D,2
                STORE    ZI +18D,2
                VLOAD
                         BVECTOR +12D     # BVECTOR (2)
                MXV*     VAD*
                         W +162D,1
                         ZI +18D,2        # B(0)*W+B(1)*(W+54)+B(2)*(W+108) FIRST PASS
                STORE    ZI +18D,2        # ZI THEN Z2 THEN Z3
                TIX,1
                         INCOR1
INCOR1          TIX,2    BON
                         Z123             # LOOP FOR Z1,Z2,Z3
                         DMENFLG
                         INCOR1A
                VLOAD
                         ZEROVECS
                STORE    ZI +12D
INCOR1A         SETPD    VLOAD
                         0
                         ZI
                VSQ      RTB
                         TPMODE
                PDVL     VSQ
                         ZI +6
                RTB      TAD
                         TPMODE
                PDVL     VSQ
                         ZI +12D
                RTB      TAD
                         TPMODE
                TAD      AXT,2
                         VARIANCE
                         0
                STORE    TRIPA            # ZI*2 + Z2*2 + Z3*2 + VARIANCE
                TLOAD    BOV
                         VARIANCE         # CLEAR OVFIND
                         +1
                STORE    TEMPVAR          # TEMP STORAGE FOR VARIANCE
                BZE
                         INCOR1C
INCOR1B         SL2      BOV
                         INCOR1C
                STORE    TEMPVAR
                INCR,2   GOTO
                DEC      1
                         INCOR1B
INCOR1C         TLOAD    ROUND
                         TRIPA
```

```
                                DMP     SQRT
                                        TEMPVAR
                                SL*     TAD
                                        0,2
                                        TRIPA
                                NORM    INCR,2
                                        X2
                                DEC     -2
                                SXA,2   AXT,2
                                        NORMGAM         # NORMALIZATION COUNT -2 FOR GAMMA
                                        162D
                                BDDV    SETPD
                                        DP1/4TH
                                        0
                                STORE   GAMMA
                                TLOAD   NORM
                                        TRIPA
                                        X1
                                DLOAD   PDDL            # PD 0-1 = NORM (A)
                                        MPAC
                                        DELTAQ
                                NORM
                                        S1
                                XSU,1   SR1
                                        S1
                                DDV     PUSH            # PD 0-1 = DELTAQ/A
                                GOTO
                                        NEWZCOMP
 -3                             SSP
                                        S2
                                        54D
INCOR2                          VLOAD   VXM*            # COMPUT OMEGA1,2,3
                                        ZI
                                        W +162D,2
                                PUSH    VLOAD
                                        ZI +6
                                VXM*    VAD
                                        W +180D,2
                                PUSH    VLOAD
                                        ZI +12D
                                VXM*    VAD
                                        W +198D,2
                                PUSH    TIX,2           # PD 2-7=OMEGA1, 8-13=OMEGA2, 14-19=OMEGA3
                                        INCOR2
                                VLOAD   STADR
                                STORE   OMEGA +12D
                                VLOAD   STADR
                                STORE   OMEGA +6
                                VLOAD   STADR
                                STORE   OMEGA
```

```
                BON      VLOAD
                         DMENFLG
                         INCOR2AB
                         ZEROVECS
                STORE    OMEGA +12D
INCOR2AB        AXT,2    SSP
                         18D
                         S2
                         6
INCOR3          VLOAD*
                         OMEGA +18D,2
                VXSC     VSL*
                         0                # DELTAQ/A
                         0,1
                STORE    DELTAX +18D,2
                TIX,2    VLOAD
                         INCOR3
                         DELTAX +6
                VSL3
                STORE    DELTAX +6
                GOTO
                         EGRESS
```

```
# INCORP2 - INCORPORATES THE COMPUTED STATE VECTOR DEVIATIONS INTO THE
# ESTIMATED STATE VECTOR. THE STATE VECTOR UPDATED MAY BE FOR EITHER THE
# LEM OR THE CSM. DETERMINED BY FLAG VEHUPFLG. (ZERO = LEM) (1 = CSM)
#
# INPUT
#       PERMANENT STATE VECTOR FOR EITHER THE LEM OR CSM
#       VEHUPFLG = UPDATE VEHICLE 0=LEM  1=CSM
#       W = ERROR TRANSITION MATRIX
#       DELTAX = COMPUTED STATE VECTOR DEVIATIONS
#       DMENFLG = SIZE OF W MATRIX (ZERO=6X6) (1=9X9)
#       GAMMA = SCALAR FOR INCORPORATION
#       ZI = VECTOR USED IN INCORPORATION
#       OMEGA = WEIGHTING VECTOR
#
# OUTPUT
#       UPDATED PERMANENT STATE VECTOR
#
# CALLING SEQUENCE
#       L       CALL    INCORP2
#
# NORMAL EXIT
#       L+1 OF CALLING SEQUENCE
#

                SETLOC  MEASINC1
                BANK

                COUNT*  $$/INCOR

INCORP2         STQ     CALL
                        EGRESS
                        INTSTALL
                VLOAD   VXSC            # CALC. GAMMA * OMEGA1,2,3
                        OMEGA
                        GAMMA
                STOVL   OMEGAM1
                        OMEGA +6
                VXSC
                        GAMMA
                STOVL   OMEGAM2
                        OMEGA +12D
                VXSC
                        GAMMA
                STORE   OMEGAM3
                EXIT
                CAF     54DD            # INITIAL IX 1 SETTING FOR W MATRIX
                TS      WIXA
                TS      WIXB
                CAF     ZERO
                TS      ZIXA            # INITIAL IX 2 SETTING FOR Z COMPONENT
                TS      ZIXB
FAZA            TC      PHASCHNG
```

```
                OCT     04022
                TC      UPFLAG
                ADRES   REINTFLG
FAZA1           CA      WIXB            # START FIRST PHASE OF INCORP2
                TS      WIXA            # TO UPDATE 6 OR 9 DIM. W MATRIX IN TEMP
                CA      ZIXB
                TS      ZIXA
                TC      INTPRET
                LXA,1   LXA,2
                        WIXA
                        ZIXA
                SSP     DLOAD*
                        S1
                        6
                        ZI,2
                DCOMP   NORM            # CALC UPPER 3X9 PARTITION OF W MATRIX
                        S2
                VXSC    XCHX,2
                        OMEGAM1
                        S2
                LXC,2   XAD,2
                        X2
                        NORMGAM
                VSL*    XCHX,2
                        0,2
                        S2
                VAD*
                        W +54D,1
                STORE   HOLDW
                DLOAD*  DCOMP           # CALC MIDDLE 3X9 PARTITION OF W MATRIX
                        ZI,2
                NORM    VXSC
                        S2
                        OMEGAM2
                XCHX,2  LXC,2
                        S2
                        X2
                XAD,2   VSL*
                        NORMGAM
                        0,2
                XCHX,2  VAD*
                        S2
                        W +108D,1
                STORE   HOLDW +6
                BOFF
                        DMENFLG         # BRANCH IF 6 DIMENSIONAL
                        FAZB
                DLOAD*  DCOMP           # CALC LOWER 3X9 PARTITION OF W MATRIX
                        ZI,2
                NORM    VXSC
```

```
                                S2
                                OMEGAM3
                XCHX,2          LXC,2
                                S2
                                X2
                XAD,2           VSL*
                                NORMGAM
                                0,2
                XCHX,2          VAD*
                                S2
                                W +162D,1
                STORE           HOLDW +12D
FAZB            CALL
                                GRP2PC
                EXIT
FAZB1           CA              WIXA            # START 2ND PHASE OF INCORP2 TO TRANSFER
                AD              6DD             #       TEMP REG TO PERM W MATRIX
                TS              WIXB
                CA              ZIXA
                AD              MINUS2
                TS              ZIXB
                TC              INTPRET
                LXA,1           SSP
                                WIXA
                                S1
                                6
                VLOAD
                                HOLDW
                STORE           W +54D,1
                VLOAD
                                HOLDW +6
                STORE           W +108D,1
                BOFF            VLOAD
                                DMENFLG
                                FAZB5
                                HOLDW +12D
                STORE           W +162D,1
FAZB2           TIX,1           GOTO
                                +2
                                FAZC            # DONE WITH W MATRIX.  UPDATE STATE VECTOR
                RTB
                                FAZA
FAZB5           SLOAD           DAD
                                ZIXB
                                12DD
                BHIZ            GOTO
                                FAZC
                                FAZB2
FAZC            CALL
                                GRP2PC
```

```
                VLOAD   VAD             # START 3RD PHASE OF INCORP2
                        X789            # 7TH, 8TH, 9TH COMPONENT OF STATE VECTOR
                        DELTAX +12D     # INCORPORATION FOR X789
                STORE   TX789
                BON     RTB
                        VEHUPFLG
                        DOCSM
                        MOVEPLEM
FAZAB           BOVB    AXT,2
                        TCDANZIG
                        0
                BOFF    AXT,2
                        MOONTHIS
                        +2
                        2
                VLOAD   VSR*
                        DELTAX          # B27 IF MOON ORBIT, B29 IF EARTH
                        0 -7,2
                VAD     BOV
                        TDELTAV
                        FAZAB1
                STOVL   TDELTAV
                        DELTAX +6       # B5 IF MOON ORBIT, B7 IF EARTH
                VSR*    VAD
                        0 -4,2
                        TNUV
                BOV
                        FAZAB2
                STCALL  TNUV
                        FAZAB3
FAZAB1          VLOAD   VAD
                        RCV
                        DELTAX
                STORE   RCV
FAZAB2          VLOAD   VAD
                        VCV
                        DELTAX +6
                STORE   VCV
                SXA,2   CALL
                        PBODY
                        RECTIFY
FAZAB3          CALL
                        GRP2PC
                BON     RTB
                        VEHUPFLG
                        DOCSM1
                        MOVEALEM
                CALL
                        SVDWN2          # STORE DOWNLINK STATE VECTOR
FAZAB4          CALL
```

```
                                GRP2PC          # PHASE CHANGE
                        BOFF    VLOAD
                                DMENFLG
                                FAZAB5          # 6 DIMENSIONAL
                                TX789           # 9 DIMENSIONAL
                        STORE   X789
        FAZAB5          LXA,1   SXA,1
                                EGRESS
                                QPRET
                        EXIT
                        TC      POSTJUMP        # EXIT
                        CADR    INTWAKE
        DOCSM           RTB     GOTO
                                MOVEPCSM
                                FAZAB
        DOCSM1          RTB     CALL
                                MOVEACSM
                                SVDWN1          # STORE DOWNLINK STATE VECTOR
                        GOTO
                                FAZAB4
        ZEROD           =       ZEROVECS
        54DD            DEC     54
        6DD             DEC     -6
        12DD            DEC     12

                        SETLOC  RENDEZ
                        BANK
                        COUNT*  $$/INCOR

        NEWZCOMP        VLOAD   ABVAL
                                ZI
                        STOVL   NORMZI
                                ZI +6
                        ABVAL   PUSH
                        DSU     BMN
                                NORMZI
                                +3
                        DLOAD   STADR
                        STORE   NORMZI
                        VLOAD   ABVAL
                                ZI +12D
                        PUSH    DSU
                                NORMZI
                        BMN     DLOAD
                                +3
                        STADR
                        STORE   NORMZI          # LARGEST ABVAL
                        DLOAD   SXA,1
                                NORMZI
                                NORMZI          # SAVE X1
                        NORM    INCR,1
```

```
                        X1
                DEC     2
                VLOAD   VSL*
                        ZI
                        0,1
                STOVL   ZI
                        ZI +6
                VSL*
                        0,1
                STOVL   ZI +6
                        ZI +12D
                VSL*    SXA,1
                        0,1
                        NORMZI +1       # SAVE SHIFT
                STORE   ZI +12D
                LXA,1   XSU,1
                        NORMGAM
                        NORMZI +1
                XSU,1
                        NORMZI +1
                SXA,1   LXC,1
                        NORMGAM
                        NORMZI +1
                XAD,1   SETPD
                        NORMZI
                        2D
                GOTO
                        INCOR2 -3
NORMZI          =       36D
```

```
# PROGRAM DESCRIPTION -- ENTIRE CONIC SUBROUTINE LOG SECTION    DATE - 1 SEPTEMBER 1967
# MOD NO. - 0                                                   LOG SECTION - CONIC SUBROUTINES
# MOD BY KRAUSE                                                 ASSEMBLY - COLOSSUS REVISION 88
#
# FUNCTIONAL DESCRIPTION -
#       THE FOLLOWING SET OF SUBROUTINES SOLVE VARIOUS PROBLEMS INVOLVING THE TRAJECTORY PRODUCED BY A CENTRAL
# INVERSE-SQUARE FORCE ACTING ON A POINT MASS, AS OUTLINED IN THE CMC AND LGC LUNAR LANDING MISSION GSOP, SECTION
# 5.5.1.2.  A GENERAL USAGE POINT-OF-VIEW WAS TAKEN IN FORMULATING, MECHANIZING, AND SCALING THE SUBROUTINES,
# RATHER THAN OPTIMIZING EACH FOR A PARTICULAR USE.  THEREFORE, MULTIPLE USAGE CAN BE MADE OF THE SUBROUTINES
# INVOLVING ANY REALISTIC SET OF CONSTRAINTS.  IT SHOULD BE NOTED THAT ONLY ONE SET OF CODING IS USED, WHETHER THE
# EARTH, MOON, OR ANY OTHER CELESTIAL BODY IS SPECIFIED AS THE CENTRAL BODY OF THE PROBLEM, PROVIDED ONE OBSERVES
# THE INHERENT SCALE CHANGE REQUIRED IN POSITION, VELOCITY, MU, AND TIME, AS OUTLINES IN MISSION PROGRAMMING
# DEFINITION MEMO NO. 10.  THIS CAN BE ACCOMPLISHED BY SIMPLY ADDING TO THE MUTABLE AND INITIALIZING THE SUBROUTINES
# APPROPRIATELY.
#
#       DUE TO THE UNIFORMITY OF THE EQUATIONS INVOLVED, CODING WAS MINIMIZED BY TREATING INDIVIDUAL EQUATIONS AND
# BLOCKS OF EQUATIONS AS SUBROUTINES OF LOWER RANK WHENEVER POSSIBLE.  AS A RESULT, THREE BY-PRODUCTS SUBROUTINES,
# DIRECTLY USABLE AS INDEPENDENT SUBROUTINES, WERE GENERATED.
#
# RESTRICTIONS -
#       THE ONLY LIMITATION IN THE SCOPE OF THE PROBLEM WHICH CAN BE SOLVED BY A PARTICULAR SUBROUTINE IS THE SCALING
# LIMIT OF EACH PARAMETER AS SPECIFIED IN THE GSOP.  THESE SCALING LIMITS WERE CHOSEN SO THAT ALL FEASIBLE TRAJECTORIES
# COULD BE HANDLED.
#
#       SINCE THE SUBROUTINES (EXCEPT KEPLER) USE COMMON SUBROUTINES OF LOWER RANK WHICH USE ERASABLE OTHER THAN
# THE PUSHLIST (DUE TO ITS LIMITED SIZE) AND COMMON INTERPRETIVE SWITCHES, THE CONIC SUBROUTINES CANNOT BE ALLOWED
# TO INTERRUPT EACH OTHER.  IT IS UP TO THE USER TO GUARANTEE THIS CONDITION.
```

```
# PROGRAM DESCRIPTION - KEPLER SUBROUTINE                                DATE - 11 OCTOBER 1967
# MOD NO. -1                                                             LOG SECTION - CONIC SUBROUTINES
# MOD BY KRAUSE                                                          ASSEMBLY - COLOSSUS 103 AND SUNDANCE 222
# MOD NO. - 2 (AUGUST 1968) BY ROBERTSON: TO PERMIT BACKDATING BY MORE THAN ONE ORBITAL PERIOD.
# MOD NO. - 3 (DEC 1968) BY ROBERTSON:  SUPPRESSION OF X-MODULO-ING
# MOD NO. - 4 (JAN 1969) BY ROBERTSON:  CLEAR OVFIND AT KEPLER ENTRY
#
# FUNCTIONAL DESCRIPTION -
#       THIS SUBROUTINE, GIVEN AN INITIAL STATE VECTOR AND THE DESIRED TRANSFER TIME THROUGH WHICH THE STATE IS TO
# BE UPDATED ALONG A CONIC TRAJECTORY, COMPUTES THE NEW, UPDATED STATE VECTOR.  THE TRAJECTORY MAY BE ANY CONIC
# SECTION - CIRCULAR, ELLIPTIC, PARABOLIC, HYPERPOLIC, OR RECTILINEAR WITH RESPECT TO THE EARTH OR THE MOON.  THE
# USE OF THE SUBROUTINE CAN BE EXTENDED USING OTHER PRIMARY BODIES BY SIMPLE ADDITIONS TO THE MUTABLE WITHOUT
# INTRODUCING ANY CODING CHANGES, ACCEPTING THE INHERENT SCALE FACTOR CHANGES IN POSITION AND VELOCITY.  AN ITERATION
# TECHNIQUE IS UTILIZED IN THE COMPUTATION.
#
#       IF A NEGATIVE TIME-OF-FLIGHT IS INPUT, THE PROGRAM WILL SOLVE FOR THE STATE WHICH WOULD BE PRODUCED BY
# EXTRAPOLATING THE POSITION BACKWARD IN TIME.
#
#       IF THE ABSOLUTE VALUE OF THE DESIRED TRANSFER TIME EXCEEDS THE ORBITAL PERIOD, THE SUBROUTINE, THROUGH A
#  MODULAR TECHNIQUE, WILL COMPUTE THE STATE CORRESPONDING TO THE DESIRED TIME (WHETHER POSITIVE OR NEGATIVE).
#
# THE RESTRICTIONS ARE -
#       1. (PREVIOUS RESTRICTION ON THE NEGATIVE DESIRED TRANSFER TIME IS NOW DELETED.)
#       2. THE PARAMETERS IN THE PROBLEM CANNOT EXCEED THEIR SCALING LIMITS AS SPECIFIED IN THE GSOP.  IF
#          ANY OF THESE LIMITS ARE EXCEEDED, THE RESULTING SOLUTION WILL BE MEANINGLESS.
#
#       THE NUMBER OF ITERATIONS AND, THEREFORE, THE COMPUTATION SPEED IS DEPENDENT ON THE ACCURACY OF THE
# GUESS, XKFPNEW.  THE AGC COMPUTATION TIME IS APPROXIMATELY .061 SECONDS FOR INITIALIZATION, .065 SECONDS FOR THE
# FINAL COMPUTATIONS, PLUS .083 SECONDS FOR EACH ITERATION.
#
# REFERENCES -
#       R-479, MISSION PROGRAMMING DEFINITION MEMO NO. 10, LUNAR LANDING MISSION GSOP, SECTION 5.5, SGA
#       MEMO 67-4.
#
# INPUT - ERASABLE INITIALIZATION REQUIRED
#                       * SCALE FACTOR  *
#       VARIABLE        *IN POWERS OF 2 *               DESCRIPTION AND REMARKS
#       --------        *-------------- *               -----------------------
#       RRECT           * +29 FOR EARTH *       DP INITIAL POSITION VECTOR IN METERS
#                       * +27 FOR MOON  *
```

```
#       VRECT           * +7 FOR EARTH  *       DP INITIAL VELOCITY VECTOR IN METERS/CENTISECOND
#                       * +5 FOR MOON   *
#       X1 (38D)        * NONE          *       INDEX REGISTER SET TO -2D OR -10D ACCORDING TO WHETHER THE EARTH OR MOON,
#                       *               *               RESPECTIVELY, IS THE CENTRAL BODY
#       TAU             * +28           *       DESIRED TRANSFER TIME IN CENTISECONDS (DP)
#                       *               *               MAY BE POS OR NEG AND ABSOLUTE VALUE MAY BE GREATER OR LESS THAN ONE ORBITAL
#       XKEPNEW         * +17 FOR EARTH *       DP GUESS OF ROOT X OF KEPLERS EQN IN SQRT(METERS). SIGN SHOULD AGREE WITH THAT OF TA
#                       * +16 FOR MOON  *               AND ABS VALUE SHOULD BE LESS THAN THAT CORRESPONDING TO A PERIOD, VIZ, 2PI S
#                       *               *               MAJOR AXIS), FOR SPEED OF CONVERGENCE, BUT IF EITHER CONDITION FAILS, XKEPNE
#                       *               *               BY KEPLER TO A POOR BUT VALID GUESS.
#       TC              * +28           *       DP PREV. VALUE OF TIME IN CENTISECS.  MUST BE LESS THAN ONE ORBITAL PERIOD.
#       XPREV           * +17 FOR EARTH *       DP PREV. VALUE OF X IN SQRT(METERS).  MUST BE LESS THAN AN X CORRESPONDING TO ONE
#                       * +16 FOR MOON  *               ORBITAL PERIOD, VIZ, 2PI SQRT(SEMI-MAJOR AXIS)
#
# SUBROUTINES CALLED -
#       DELTIME
#
# CALLING SEQUENCE AND NORMAL EXIT MODES -
#       KEPRTN-2        GOTO                    # MUST BE IN INTERPRETIVE MODE BUT OVFIND ARBITRARY.
#       KEPRTN-1                KEPLER          # RETURNS WITH XPREV IN MPAC. PL IS AT 0.
#       KEPRTN          ...                     # CONTINUE
#
# KEPLER MUST NOT BE CALLED DIRECTLY SINCE AN INTERRUPTION OF IT WOULD DESTROY THE ERASABLES IT NEEDS TO COMPLETE
# THE INTERRUPTED JOB.  THEREFORE THE USER MUST CALL CSMCONIC OR LEMCONIC WHICH GUARANTEES NO INTERRUPTS AND WHICH
# ALSO CALLS KEPPREP TO COMPUTE A GUESS OF XKEPNEW.
#
# ABORT EXIT MODES -
#       NONE
#
# OUTPUT -
#                       * SCALE FACTOR  *
#       VARIABLE        *IN POWERS OF 2 *       DESCRIPTION AND REMARKS
#       --------        *-------------- *       -----------------------
#       RCV             * +29 FOR EARTH *       DP TERMINAL POSITION VECTOR IN METERS
#                       * +27 FOR MOON  *
#       VCV             * +7 FOR EARTH  *       DP TERMINAL VELOCITY VECTOR IN METERS/CENTISEC
#                       * +5 FOR MOON   *
#       TC              * +28           *       DP TRANSFER TIME IN CENTISECS TO WHICH KEPLER CONVERGED. ALWAYS LESS THAN ONE PERIOD
#       XPREV           * +17 FOR EARTH *       DP VALUE OF X IN SQRT(METERS) TO WHICH KEPLER CONVERGED. ALWAYS LESS THAN THE X
#                       * +16 FOR MOON  *               CORRESPONDING TO ONE PERIOD.
```

```
# FOR OTHER OUTPUT WHICH MAY BE OF USE, SEE DEBRIS.
#
# DEBRIS -
#       PARAMETERS WHICH MAY BE OF USE -
#                       * SCALE FACTOR  *
#       VARIABLE        *IN POWERS OF 2 *       DESCRIPTION AND REMARKS
#       --------        *-------------- *       -----------------------
#       URRECT          * +1            *       DP UNIT VECTOR OF INITIAL POSITION
#       R1              * +29 FOR EARTH *       DP MAGNITUDE OF INITIAL POSITION IN METERS
#                       * +27 FOR MOON  *
#       ALPHA           * -22 FOR EARTH *       DP INVERSE OF SEMI-MAJOR AXIS IN 1/METERS
#                       * -20 FOR MOON  *
#       TMODULO         * +28           *       DP INTEGRAL NUMBER OF PERIODS IN CENTISECS, WHICH WAS SUBTRACTED FROM TAU, TO PRODUC
#                       *               *       TAU. OF LESS THAN ONE PERIOD.
#
# PARAMETERS OF NO USE -
#       DP PARAMETERS - EPSILONT, DELX, DELT, RCNORM, XMODULO, PLUS PUSHLIST REGISTERS 0 THROUGH 39D.
```

```
# PROGRAM DESCRIPTION - LAMBERT SUBROUTINE                                DATE - 1 SEPTEMBER 1967
# MOD NO. - 0                                                             LOG SECTION - CONIC SUBROUTINES
# MOD BY KRAUSE                                                           ASSEMBLY - COLOSSUS REVISION 88
#
# FUNCTIONAL DESCRIPTION -
#       THIS SUBROUTINE CALCULATES THE INITIAL VELOCITY REQUIRED TO TRANSFER A POINT-MASS ALONG A CONIC TRAJECTORY
# FROM AN INITIAL POSITION TO A TERMINAL POSITION IN A PRESCRIBED TIME INTERVAL.  THE RESULTING TRAJECTORY MAY BE
# A SECTION OF A CIRCLE, ELLIPSE, PARABOLA, OR HYPERBOLA WITH RESPECT TO TEH EARTH OR THE MOON.  THE USE OF THE
# SUBROUTINE CAN BE EXTEDED USING OTHER PRIMARY BODIES BY SIMPLE ADDITIONS TO THE MUTABLE WITHOUT INTRODUCING ANY
# CODING CHANGES, ACCEPTING THE INHERENT SCALE FACTOR CHANGES IN POSITION AND VELOCITY.  AN ITERATION TECHNIQUE IS
# UTILIZED IN THE COMPUTATION.
#
# THE RESTRICTIONS ARE: -
#       1. RECTILINEAR TRAJECTORIES CANNOT BE COMPUTED.
#       2. AN ACCURACY DEGRADATION OCCURS AS THE COSINE OF THE TRUE ANOMALY DIFFERENCE APPROACHES +1.0.
#       3. THE ANGLE BETWEEN ANY POSITION VECTOR AND ITS VELOCITY VECTOR MUST BE GREATER THAN 1 DEGREE 47.5 MINUTES
#          AND LESS THAN 178 DEGREES 12.5 MINUTES.
#       4. NEGATIVE TRANSFER TIME IS AMBIGUOUS AND WILL RESULT IN NO SOLUTION.
#       5. THE PARAMETERS IN THE PROBLEM MUST NOT EXCEED THEIR SCALING LIMITS SPECIFIED IN THE GSOP.  IF THE
#          LIMITS ARE EXCEEDED, THE RESULTING SOLUTION WILL BE MEANINGLESS.
#
#       THE NUMBER OF ITERATIONS AND, THEREFORE, THE COMPUTATIONS SPEED IS DEPENDENT ON THE ACCURACY OF THE FIRST
# GUESS OF THE INDEPENDENT VARIABLE, COGA.  THE AGC COMPUTATION TIME IS APPROXIMATELY
# .105 SECONDS FOR INITIALIZATION, .069 SECONDS FOR FINAL COMPUTATIONS, PLUS .205 SECONDS FOR EACH ITERATION.
#
# REFERENCES -
#       R-479, MISSION PROGRAMMING DEFINITION MEMO NO. 10, LUNAR LANDING MISSION GSOP - SECTION 5.5, SGA MEMO 67-8,
#       SGA MEMO 67-4.
#
# INPUT - ERASABLE INITIALIZATION REQUIRED
#
#                               * SCALE FACTOR  *
#       VARIABLE                *IN POWERS OF 2 *       DESCRIPTION AND REMARKS
#       --------                *-------------- *       -----------------------
#       R1VEC                   * +29 FOR EARTH *       DP INITIAL POSITION VECTOR IN METERS
#                               * +27 FOR MOON  *
#       R2VEC                   * +29 FOR EARTH *       DP TARGET OR TERMINAL POSITION VECTOR IN METERS
#                               * +27 FOR MOON  *
#       TDESIRED                * +28           *       DP DESIRED TRANSFER TIME IN CENTISECONDS
#       X1 (38D)                * NONE          *       INDEX REGISTER SET TO -2D OR -10D ACCORDING TO WHETHER THE EARTH OR MOON,
#                               *               *               RESPECTIVELY, IS THE CENTRAL BODY
#       GEOMSGN                 * NONE          *       SP +.5 IF DESIRED TRANSFER ANGLE IS LESS THAN 180 DEGREES, -.5 IF GREATER THAN 180 D
#       GUESSW                  * NONE          *       AN INTERPRETER SWITCH TO BE SET IF NO GUESS OF COGA IS AVAILABLE, CLEAR IF A GUESS O
```

```
#                       *               *               COGA IS TO BE USED BY LAMBERT
#       COGA            * +5            *       DP GUESS OF COTANGENT OF FLIGHT PATH ANGLE (MEASURED FROM VERTICAL).  THIS WILL BE
#                       *               *       IGNORED IF GUESSW IS SET.
#       NORMSW          * NONE          *       AN INTERPRETER SWITCH TO BE SET IF UN IS TO BE AN INPUT TO THE SUBROUTINE, CLEAR IF
#                       *               *               LAMBERT IS TO COMPUTE ITS OWN NORMAL (UN).
#       UN              * +1            *       DP UNIT NORMAL TO THE DESIRED ORBIT PLANE IN THE DIRECTION OF THE RESULTING ANGULAR
#                       *               *               MOMENTUM VECTOR.  THIS WILL BE IGNORED IF NORMSW IS CLEAR.
#       VTARGTAG        * NONE          *       A S.P. TAG TO BE SET TO ZERO IF LAMBERT IS TO COMPUTE THE VELOCITY AT R2VEC AS WELL
#                       *               *               AT R1VEC.
#       ITERCTR         * NONE          *       A S.P. COUNTER WHICH SPECIFIES THE MAXIMUM NUMBER OF ITERATIONS ALLOWABLE.
#                       *               *               (AN ITERATION MEANS A PASS THRU KEPLER EQN (DELTIME). AT LEAST ONE OF THESE
#                       *               *               ALWAYS OCCUR, EVEN IF COGA CORRESPONDING TO SOLUTION WERE INPUT AS A GUESS.)
#                       *               *               TWENTY ITERATIONS ARE SUFFICIENT TO SOLVE ALL PROBLEMS INCLUDING THOSE WITHO
#
# SUBROUTINES CALLED -
#       GEOM, GETX, DELTIME, ITERATOR, LAMENTER (PART OF NEWSTATE)
#
# CALLING SEQUENCE AND NORMAL EXIT MODES -
#       L       CALL                    # MUST BE IN INTERPRETIVE MODE BUT OVFIND ARBITRARY.
#       L+1             LAMBERT         # RETURNS WITH PL AT 0 AND WITH VVEC IN MPAC IF VTARGTAG WAS WAS NON-ZERO OR VTARGET
#                                       # IN MPAC IF VTARGTAG WAS ZERO
#       L+2     BON                     # CONTINUE IF SOLNSW CLEAR SINCE SOLUTION IS ACCEPTABLE
#       L+3             SOLNSW
#       L+4             LAMABORT
#
#       IF A LAMBERT RESULT IS TO BE A FIRST GUESS FOR THE NEXT LAMBERT CALCULATION, COGA MUST BE PRESERVED AND
# GUESSW MUST BE CLEAR FOR EACH SUCCEEDING LAMBERT CALL.
#
# ABORT EXIT MODES -
#       IF SOLNSW WAS SET UPON EXITING, EITHER LAMBERT WAS ASKED TO COMPUTE A TRANSFER TOO NEAR 0 OR 360 DEG, OR T
# WAS TOO SMALL TO PRODUCE A REALISTIC TRANSFER BETWEEN R1VEC AND R2VEC.  IN EITHER CASE THE FIX MUST BE MADE
# ACCORDING TO THE NEEDS OF THE PARTICULAR USER.  THE ABORT EXIT MODE MAY BE CODED AS ...
# LAMBABORT     DLOAD   ABS             # A MEASURE OF THE PROXIMITY TO 0 OR
#                       1-CSTH          # 360 DEGREES.
#               DSU     BMN
#                       ONEBIT
#                       CHANGER2        # CHANGE R2VEC DIRECTION SLIGHTLY.
#               DLOAD   DAD
#                       TDESIRED
#                       SOMETIME
#               STCALL  TDESIRED        # INCRESE TDESIRED
#                       LAMBERT
#
```

```
# OUTPUT -
#                       * SCALE FACTOR  *
#       VARIABLE        *IN POWERS OF 2 *       DESCRIPTION AND REMARKS
#       --------        *-------------- *       -----------------------
#       VVEC            * +7 FOR EARTH  *       DP INITIAL VELOCITY VECTOR IN METERS/CENTISECOND REQUIRED TO SATISFY THE BOUNDARY VA
#                       * +5 FOR MOON   *               PROBLEM.
#       VTARGET         * +7 FOR EARTH  *       DP RESULTANT VELOCITY VECTOR AT R2VEC IN METERS/CENTISECOND.
#                       * +5 FOR MOON   *
#       SOLNSW          * NONE          *       INTERPRETER SWITCH WHICH IS SET IF THE SUBROUTINE CANNOT SOLVE THE PROBLEM, CLEAR IF
#                       *               *               SOLUTION EXISTS.
#
#       FOR OTHER OUTPUT WHICH MAY BE OF USE, SEE DEBRIS.
#
# DEBRIS -
#       PARAMETERS WHICH MAY BE OF USE -
#
#                       * SCALE FACTOR  *
#       VARIABLE        *IN POWERS OF 2 *       DESCRIPTION AND REMARKS
#       --------        *-------------- *       -----------------------
#       SNTH            * +1            *       DP SIN OF ANGLE BETWEEN R1VEC AND R2VEC
#       CSTH            * +1            *       DP COSINE OF ANGLE
#       1-CSTH          * +2            *       DP 1-CSTH
#       COGA            * +5            *       DP COTAN OF INITIAL REQUIRED FLIGHT PATH ANGLE MEASURED FROM VERTICAL
#       P               * +4            *       DP RATIO OF SEMILATUS RECTUM TO INITIAL RADIUS
#       R1A             * +6            *       DP RATIO OF INITIAL RADIUS TO SEMI-MAJOR AXIS
#       R1 (32D)        * +29 FOR EARTH *       DP INITIAL RADIUS IN METERS
#                       * +27 FOR MOON  *
#       UR1             * +1            *       DP UNIT VECTOR OF R1VEC
#       U2              * +1            *       DP UNIT VECTOR OF R2VEC
#
#       PARAMETERS OF NO USE -
#               DP PARAMETERS - EPSILONL, CSTH-RHO, TPREV, TERRLAMB, R2, RTNLAMB (SP), PLUS PUSHLIST REGISTER 0 THROUGH 41D
#               ADDITIONAL INTERPRETIVE SWITCHES USED - INFINFLG, 360SW, SLOPESW, ORDERSW
```

```
# PROGRAM DESCRIPTION - TIME-THETA SUBROUTINE                              DATE - 1 SEPTEMBER 1967
# MOD NO. - 0                                                              LOG SECTION - CONIC SUBROUTINES
# MOD BY KRAUSE                                                            ASSEMBLY - COLOSSUS REVISION 88
#
# FUNCTIONAL DESCRIPTION -
#       THIS SUBROUTINE, GIVEN AN INITIAL STATE VECTOR AND A DESIRED TRUE-ANOMALY-DIFFERENCE THROUGH WHICH THE
# STATE IS TO BE UPDATED ALONG A CONIC TRAJECTORY, CALCULATES THE CORRESPONDING TIME-OF-FLIGHT AND, IN ADDITION,
# PROVIDES THE OPTION OF COMPUTING THE NEW UPDATED STATE VECTOR.  THE RESULTING TRAJECTORY MAY BE A SECTION OF A
# CIRCLE, ELLIPSE, PARABOLA, OR HYPERBOLA WITH RESPECT TO THE EARTH OR THE MOON.  THE USE OF TEH SUBROUTINE CAN BE
# EXTENDED USING OTHER PRIMARY BODIES BY SIMPLE ADDITIONS TO THE MUTABLE WTIHOUT INTRODUCING ANY CODING CHANGES,
# ACCEPTING THE INHERENT SCALE FACTOR CHANGES IN POSITION AND VELOCITY.
#
# THE RESTRICTIONS ARE -
#       1. THE ANGLE BETWEEN ANY POSITION VECTOR AND ITS VELOCITY VECTOR MUST BE GREATER THAN 1 DEGREE 47.5 MINUTES
#          AND LESS THAN 178 DEGREES 12.5 MINUTES.
#       2. THE PARAMETERS IN THE PROBLEM MUST NOT EXCEED THEIR SCALING LIMITS SPECIFIED IN THE GSOP.  IF THE LIMITS
#          ARE EXCEEDED, THE RESULTING SOLUTION WILL BE MEANINGLY.
#
#       THE AGC COMPUTATION TIME IS APPROXIMATELY .292 SECONDS.
#
# REFERENCES -
#       R-479, MISSION PROGRAMMING DEFINITION MEMO NO. 10, LUNAR LANDING MISSION GSOP-SECTION 5.5, SGA MEMO 67-8.
#
# INPUT - ERASABLE INITIALIZATION REQUIRED
#                     * SCALE FACTOR  *
#       VARIABLE      *IN POWERS OF 2 *       DESCRIPTION AND REMARKS
#       --------      *-------------- *       -----------------------
#       RVEC          * +29 FOR EARTH *       DP INITIAL POSITION VECTOR IN METERS
#                     * +27 FOR MOON  *
#       VVEC          * +7 FOR EARTH  *       DP INITIAL VELOCITY VECTOR IN METERS/CENTISECOND
#                     * +5 FOR MOON   *
#       SNTH          * +1            *       DP SINE OF TRUE-ANOMALY-DIFFERENCE THROUGH WHICH THE STATE IS TO BE UPDATED
#       CSTH          * +1            *       DP COSINE OF THE ANGLE
#       RVSW          * NONE          *       AN INTERPRETIVE SWITCH TO BE SET IF ONLY TIME IS TO BE AN OUTPUT, CLEAR IF THE NEW S
#                     *               *               IS TO BE COMPUTED ALSO.
#       X1 (38D)      * NONE          *       INDEX REGISTER TO BE SET TO -2D OR -10D ACCORDING TO WHETHER THE EARTH OR MOON,
#                     *               *               RESPECTIVELY, IS THE CENTRAL BODY.
#
# SUBROUTINES CALLED -
```

```
#       PARAM, GEOM, GETX, DELTIME, NEWSTATE
#
# CALLING SEQUENCE AND NORMAL EXIT MODES -
#       IF ONLY TIME IS DESIRED AS OUTPUT -
#       L       SET     CALL            # MUST BE IN INTERPRETIVE MODE BUT OVFIND ARBITRARY.
#       L+1             RVSW
#       L+2             TIMETHET        # RETURN WITH PL AT 0 AND T IN MPAC
#       L+3     ...                     # CONTINUE
#
#       IF THE UPDATE STATE VECTOR IS DESIRED AS WELL -
#       L       CLEAR   CALL            # MUST BE IN INTERPRETIVE MODE BUT OVFIND ARBITRARY.
#       L+1             RVSW
#       L+2             TIMETHET        # RETURNS WITH PL AT 6.  THE INITIAL POSITION VECTOR IS IN OD OF THE PUSHLIST AND
#                                       # THE INITIAL VELOCITY VECTOR IN MPAC.
#       L+3     STOVL   NEWVVEC
#       L+4     STADR
#       L+5     STORE   NEWRVEC         # NEWVVEC AND NEWRVEC ARE SYMBOLIC REPRESENTATIONS OF THE USERS LOCATIONS.
#       L+6     ...                     # CONTINUE.
#
# ABORT EXIT MODES -
#       IF COGAFLAG AND/OR INFINFLG IS SET AT THE EXIT TO TIME-THETA, TIME-THETA WILL TRANSFER TO POODOO WITH
#       AN ALARM CODE (ORIGINALLY 00607), AND NOT RETURN TO THE CALLING PROGRAM.  (PCR 692 AND 721).
#
# OUTPUT -
#                       * SCALE FACTOR  *
#       VARIABLE        *IN POWERS OF 2 *       DESCRIPTION AND REMARKS
#       --------        *-------------- *       -----------------------
#       T (30D)         * +28           *       DP TRANSFER TIME IN CENTISECONDS
#       INFINFLG        * NONE          *       AN INTERPRETIVE SWITCH WHICH IS SET IF THE TRANSFER ANGLE REQUIRES CLOSURE THROUGH
#                       *               *               INFINITY (NO SOLUTION), CLEAR IF A PHYSICAL SOLUTION IS POSSIBLE.
#       COGAFLAG        * NONE          *       AN INTERPRETIVE SWITCH WHICH IS SET IF RESTRICTION 1 HAS BEEN VIOLATED (NO SOLUTION)
#                       *               *               CLEAR IF NOT.
#
#       IN ADDITION, IF RVSW IS CLEAR, THE FOLLOWING ARE OUTPUT -
#       MPAC -          * +7 FOR EARTH  *       DP TERMINAL VELOCITY VECTOR IN METERS/CENTISEC.
#        MPAC +5        * +5 FOR MOON   *
#       OD - 5D         * +29 FOR EARTH *       DP TERMINAL POSITION VECTOR IN METERS (PL AT 6D)
#                       * +27 FOR MOON  *
#
#       FOR OTHER OUTPUT WHICH MAY BE OF USE, SEE DEBRIS.
#
```

```
# DEBRIS -
#       PARAMETERS WHICH MAY BE OF USE -
#
#                       * SCALE FACTOR  *
#       VARIABLE        *IN POWERS OF 2 *       DESCRIPTION AND REMARKS
#       --------        *-------------- *       -----------------------
#       R1 (32D)        * +29 FOR EARTH *       DP MAGNITUDE OF INITIAL POSITION VECTOR, RVEC, IN METERS
#                       * +27 FOR MOON  *
#       R1A             * +6            *       DP RATIO OF R1 TO SEMIMAJOR AXIS (NEG. FOR HYPERBOLIC TRAJECTORIES)
#       P               * +4            *       DP RATIO OF SEMILATUS RECTUM TO R1
#       COGA            * +5            *       DP COTAN OF ANGLE BETWEEN RVEC AND VVEC
#       UR1             * +1            *       DP UNIT VECTOR OF RVEC
#       U2              * +1            *       DP UNIT VECTOR OF VVEC
#       UN              * +1            *       DP UNIT VECTOR OF UR1*U2
#
#       PARAMETERS OF NO USE -
#       SP PARAMETERS -- RTNTT, GEOMSGN, RTNPRM, MAGVEC2=R2 (DP), PLUS PUSHLIST LOCATIONS 0-11D, 14D-21D, 24D-39D, 41D
#       ADDITIONAL INTERPRETIVE SWITCHES USED -- NORMSW, 360SW
```

```
# PROGRAM DESCRIPTION - TIME-RADIUS SUBROUTINE          DATE - 11 OCTOBER 1967
# MOD NO. -1                                            LOG SECTION - CONIC SUBROUTINES
# MOD BY KRAUSE                                         ASSEMBLY - COLOSSUS REVISION 88
#
# FUNCTIONAL DESCRIPTION -
#
#       THIS SUBROUTINE, GIVEN AN INITIAL STATE VECTOR AND A DESIRED RADIUS TO WHICH THE
#       STATE IS TO BE UPDATED ALONG A CONIC TRAJECTORY, CALCULATES THE CORRESPONDING TIME-OF-FLIGHT AND, IN ADDITION,
#       PROVIDES THE OPTION OF COMPUTING THE NEW UPDATED STATE VECTOR.  THE RESULTING TRAJECTORY MAY BE A SECTION OF A
#       CIRCLE, ELLIPSE, PARABOLA, OR HYPERBOLA WITH RESPECT TO THE EARTH OR THE MOON.  THE USE OF THE SUBROUTINE CAN BE
#       EXTENDED USING OTHER PRIMARY BODIES BY SIMMPE ADDITIONS TO THE MUTABLE WITHOUT INTRODUCING ANY CODING CHANGES,
#       ACCEPTING THE INHERENT SCALE FACTOR CHANGES IN POSITION AND VELOCITY.
#
#       IF THE DESIRED RADIUS IS BEYOND THE RADIUS OF APOCENTER OF THE CONIC OR BELOW THE RADIUS OF PERICENTER,
#       APSESW WILL BE SET AND THE SUBROUTINE WILL RETURN THE APOCENTER OR PERICENTER SOLUTION, RESPECTIVELY.
#
# THE RESTRICTIONS ARE -
#       1. THE ANGLE BETWEEN ANY POSITION VECTOR AND ITS VELOCITY VECTOR MUST BE GREATER THAN 1 DEGREE 47.5 MINUTES
#          AND LESS THAN 178 DEGREES 12.5 MINUTES.
#       2. THE PARAMETERS IN THE PROBLEM MUST NOT EXCEED THEIR SCALING LIMITS SPECIFIED IN THE GSOP.  IF THE LIMITS
#          ARE EXCEEDED, THE RESULTING SOLUTION WILL BE MEANINGLESS.
#       3. AN ACCURACY DEGRADATION OCCURS AS THE SENSITIVITIES OF TIME AND UPDATED STATE VECTOR TO CHANGES IN
#          RDESIRED INCREASE.  THIS WILL OCCUR NEAR EITHER APSIS OF THE CONIC AND WHEN THE CONIC IS NEARLY CIRCULAR.  IN
#          PARTICULAR, IF THE CONIC IS AN EXACT CIRCLE, THE PROBLEM IS UNDEFINED AND THE SUBROUTINE WILL ABORT.
#
#       THE AGC COMPUTATION TIME IS APPROXIMATELY .363 SECONDS.
#
# REFERENCES -
#       R-479, MISSION PROGRAMMING DEFINITION MEMO NO. 10, LUNAR LANDING MISSION GSOP-SECTION 5.5, SGA MEMO 67-8.
#
# INPUT - ERASABLE INITIALIZATION REQUIRED.
#                      * SCALE FACTOR  *
#       VARIABLE       *IN POWERS OF 2 *       DESCRIPTION AND REMARKS
#       --------       *-------------- *       -----------------------
#       RVEC           * +29 FOR EARTH *       DP INITIAL POSITION VECTOR IN METERS
#                      * +27 FOR MOON  *
#       VVEC           * +7 FOR EARTH  *       DP INITIAL VELOCITY VECTOR IN METERS/CENTISECOND
#                      * +5 FOR MOON   *
#       RDESIRED       * +29 FOR EARTH *       DP TERMINAL RADIAL DISTANCE ON CONIC TRAJECTORY FOR WHICH TRANSFER TIME IS TO BE
#                      * +27 FOR MOON  *               COMPUTED
#       SGNRDOT        * NONE          *       SP TAG SET TO +.5 OR -.5 ACCORDING TO WHETHER THE RADIAL VELOCITY AT RDESIRED IS TO
#                      *               *               POSITIVE OR NEGATIVE, RESPECTIVELY.   THIS TAG REDUCES THE DOUBLE-VALUED PRO
```

```
#                       *               *               SINGLE-VALUED PROBLEM.
#       X1 (38D)        * NONE          *       INDEX REGISTER TO BE SET TO -2D OR -10D ACCORDING TO WHETHER THE EARTH OR MOON,
#                       *               *               RESPECTIVELY, IS THE CENTRAL BODY.
#       RVSW            * NONE          *       AN INTERPRETIVE SWITCH TO BE SET IF ONLY TIME IS TO BE AN OUTPUT, CLEAR IF THE NEW S
#                       *               *               IS TO BE COMPUTED ALSO.
#
# SUBROUTINES CALLED -
#       PARAM, GEOM, GETX, DELTIME, NEWSTATE
#
# CALLING SEQUENCE AND NORMAL EXIT MODES -
#
#       IF ONLY TIME IS DESIRED AS OUTPUT -
#       L       SET     CALL            # MUST BE IN INTERPRETIVE MODE BUT OVFIND ARBITRARY.
#       L+1             RVSW
#       L+2             TIMERAD         # RETURN WITH PL AT 0 AND T IN MPAC
#       L+3     ...                     # CONTINUE
#
#       IF THE UPDATE STATE VECTOR IS DESIRED AS WELL -
#       L       CLEAR   CALL            # MUST BE IN INTERPRETIVE MODE BUT OVFIND ARBITRARY.
#       L+1             RVSW
#       L+2             TIMERAD         # RETURNS WITH PL AT 6.  THE INITIAL POSITION VECTOR IS IN OD OF THE PUSHLIST AND
#                                       # THE INITIAL VELOCITY VECTOR IN MPAC.
#       L+3     STOVL   NEWVVEC
#       L+4     STADR
#       L+5     STORE   NEWRVEC         # NEWVVEC AND NEWRVEC ARE SYMBOLIC REPRESENTATIONS OF THE USERS LOCATIONS.
#       ...                             # CONTINUE
#
# ABORT EXIT MODES -
#       IF SOLNSW AND/OR COGAFLAG AND/OR INFINFLG IS SET AT THE EXIT TO TIME-RADIUS, TIME-RADIUS WILL TRANSFER
#       TO POODOO WITH AN ALARM CODE (ORIGINALLY 00607), AND NOT RETURN TO THE CALLING PROGRAM. (PCR 692 & 721)
#
# OUTPUT -
#                       * SCALE FACTOR  *
#       VARIABLE        *IN POWERS OF 2 *       DESCRIPTION AND REMARKS
#       --------        *-------------- *       -----------------------
#       T (30D)         * +28           *       DP TRANSFER TIME IN CENTISECONDS.
#       INFINFLG        * NONE          *       AN INTERPRETIVE SWITCH WHICH IS SET IF RDESIRED AND SGNRDOT REQUIRE CLOSURE THROUGH
#                       *               *               INFINITY (NO SOLUTION), CLEAR IF A PHYSICAL SOLUTION IS POSSIBLE.
#       COGAFLAG        * NONE          *       AN INTERPRETIVE SWITCH WHICH IS SET IF RESTRICTION 1 HAS BEEN VIOLATED (NO SOLUTION)
#                       *               *               CLEAR IF NOT.
#       APSESW          * NONE          *       AN INTERPRETIVE SWITCH WHICH IS SET IF RDESIRED WAS GREATER THAN RADIUS OF APOCENTER
```

```
#                       *               *               LESS THAN RADIUS OF PERICENTER.  THE APOCENTER OR PERICENTER SOLUTION, RESPE
#                       *               *               WILL THEN BE RETURNED.  THE SWITCH IS CLEAR IF RDESIRED WAS BETWEEN PERICENT
#                       *               *               APOCENTER.
#       SOLNSW          * NONE          *       AN INTERPRETIVE SWITCH WHICH IS SET IF THE CONIC IS SO CLOSE TO A CIRCLE THAT THE TE
#                       *               *               POINT IS AMBIGUOUS, VIOLATING RESTRICTION 3.  IF ECCENTRICITY IS GREATER THA
#                       *               *               MINUS-18, THE SWITCH IS CLEAR.
#
#       IN ADDITION, IF RVSW IS CLEAR, THE FOLLOWING ARE OUTPUT -
#
#       MPAC -          * +7 FOR EARTH  *       DP TERMINAL VELOCITY VECTOR IN METERS/CENTISEC.
#        MPAC +5        * +5 FOR MOON   *
#       0D - 5D         * +29 FOR EARTH *       DP TERMINAL POSITION VECTOR IN METERS (PL AT 6D)
#                       * +27 FOR MOON  *
#
#       FOR OTHER OUTPUT WHICH MAY BE OF USE, SEE DEBRIS.
#
# DEBRIS -
#       PARAMETERS WHICH MAY BE OF USE -
#
#                       * SCALE FACTOR  *
#       VARIABLE        *IN POWERS OF 2 *       DESCRIPTION AND REMARKS
#       --------        *-------------- *       -----------------------
#       R1 (32D)        * +29 FOR EARTH *       DP MAGNITUDE OF INITIAL POSITION VECTOR, RVEC, IN METERS
#                       * +27 FOR MOON  *
#       R1A             * +6            *       DP RATIO OF R1 TO SEMIMAJOR AXIS (NEG. FOR HYPERBOLIC TRAJECTORIES)
#       P               * +4            *       DP RATIO OF SEMILATUS RECTUM TO R1
#       COGA            * +5            *       DP COTAN OF ANGLE BETWEEN RVEC AND VVEC
#       UR1             * +1            *       DP UNIT VECTOR OF RVEC
#       U2              * +1            *       DP UNIT VECTOR OF VVEC
#       UN              * +1            *       DP UNIT VECTOR OF UR1*U2
#       CSTH            * +1            *       DP COSINE OF TRUE ANOMALY DIFFERENCE BETWEEN RVEC AND RDESIRED.
#       SNTH            * +1            *       DP SINE OF TRUE ANOMALY DIFFERENCE.
#
#       PARAMETERS OF NO USE -
#               SP PARAMETERS -- RTNTT, GEOMSGN, RTNPRM, MAGVEC2=R2 (DP), PLUS PUSHLIST LOCATIONS 0-11D, 14D-21D, 24D-39D, 41D
#               ADDITIONAL INTERPRETIVE SWITCHES USED -- NORMSW, 360SW
#
```

```
# PROGRAM DESCRIPTION - APSIDES SUBROUTINE              DATE - 1 SEPTEMBER 1967
# MOD NO. - 0                                           LOG SECTION - CONIC SUBROUTINES
# MOD BY KRAUSE                                         ASSEMBLY - COLOSSUS REVISION 88
#
# FUNCTIONAL DESCRIPTION -
#
#       THIS SUBROUTINE, GIVEN AN INITIAL STATE VECTOR CALCULATES THE RADIUS OF PERICENTER AND OF APOCENTER AND THE
#       ECCENTRICITY OF THE RESULTING CONIC TRAJECTORY, WHICH MAY BE A STRAIGHT LINE,
#       CIRCLE, ELLIPSE, PARABOLA, OR HYPERBOLA WITH RESPECT TO THE EARTH OR THE MOON.  THE USE OF THE SUBROUTINE CAN
#       BE EXTENDED USING OTHER PRIMARY BODIES BY SIMPLE ADDITIONS TO THE MUTABLE WITHOUT INTRODUCING ANY CODING CHANGES,
#       ACCEPTING THE INHERENT SCALE FACTOR CHANGES IN POSITION AND VELOCITY.
#
# THE RESTRICTIONS ARE -
#       1. IF APOCENTER IS BEYOND THE SCALING OF POSITION, THE SCALE FACTOR LIMIT (536,870,910 METERS WITH RESPECT
#          TO THE EARTH OR 134,217,727.5 METERS WITH RESPECT TO THE MOON) WILL BE RETURNED.
#       2. THE PARAMETERS IN THE PROBLEM MUST NOT EXCEED THEIR SCALING LIMITS SPECIFIED IN THE GSOP.  IF THE LIMITS
#          ARE EXCEEDED, THE RESULTING SOLUTION WILL BE MEANINGLESS.
#
#       THE AGC COMPUTATION TIME IS APPROXIMATELY .103 SECONDS.
#
# REFERENCES -
#       MISSION PROGRAMMING DEFINITION MEMO NO. 10, LUNAR LANDING MISSION GSOP-SECTION 5.5
#
# INPUT - ERASABLE INITIALIZATION REQUIRED
#                       * SCALE FACTOR  *
#       VARIABLE        *IN POWERS OF 2 *       DESCRIPTION AND REMARKS
#       --------        *-------------- *       -----------------------
#       RVEC            * +29 FOR EARTH *       DP INITIAL POSITION VECTOR IN METERS
#                       * +27 FOR MOON  *
#       VVEC            * +7 FOR EARTH  *       DP INITIAL VELOCITY VECTOR IN METERS/CENTISECOND
#                       * +5 FOR MOON   *
#       X1 (38D)        * NONE          *       INDEX REGISTER TO BE SET TO -2D OR -10D ACCORDING TO WHETHER THE EARTH OR MOON,
#                       *               *               RESPECTIVELY, IS THE CENTRAL BODY.
#
# SUBROUTINES CALLED -
#       PARAM, GEOM
#
# CALLING SEQUENCE AND NORMAL EXIT MODES -
```

```
#       IF ONLY TIME IS DESIRED AS OUTPUT -
#       L       CALL                    # MUST BE IN INTERPRETIVE MODE BUT OVFIND ARBITRARY.
#       L+1             APSIDES         # RETURNS WITH PL AT 0, RADIUS OF APOCENTER IN MPAC AND RADIUS OF PERICENTER IN OD
#       L+2     STODL   APOAPSE
#       L+3             OD
#       L+4     STORE   PERIAPSE        # APOAPSE AND PERIAPSE ARE SYMBOLIC REPRESENTATIONS OF THE USERS LOCATIONS
#       L+5     ...                     # CONTINUE
#
# OUTPUT -
#                       * SCALE FACTOR  *
#       VARIABLE        *IN POWERS OF 2 *       DESCRIPTION AND REMARKS
#       --------        *-------------- *       -----------------------
#       MPAC            * +29 FOR EARTH *       DP RADIUS OF APOCENTER IN METERS
#                       * +27 FOR MOON  *
#       OD-1D           * +29 FOR EARTH *       DP RADIUS OF PERICENTER IN METERS
#                       * +27 FOR MOON  *
#       ECC             * +3            *       DP ECCENTRICITY OF CONIC TRAJECTORY.
#
#       FOR OTHER OUTPUT WHICH MAY BE OF USE, SEE DEBRIS.
#
# DEBRIS -
#
#       PARAMETERS WHICH MAY BE OF USE -
#
#                       * SCALE FACTOR  *
#       VARIABLE        *IN POWERS OF 2 *       DESCRIPTION AND REMARKS
#       --------        *-------------- *       -----------------------
#       R1 (32D)        * +29 FOR EARTH *       DP MAGNITUDE OF INITIAL POSITION VECTOR, RVEC, IN METERS
#                       * +27 FOR MOON  *
#       R1A             * +6            *       DP RATIO OF R1 TO SEMI-MAJOR AXIS (NEG. FOR HYPERBOLIC TRAJECTORIES)
#       P               * +4            *       DP RATIO OF SEMILATUS RECTUM TO R1
#       COGA            * +5            *       DP COTAN OF ANGLE BETWEEN RVEC AND VVEC
#       UR1             * +1            *       DP UNIT VECTOR OF RVEC
#       U2              * +1            *       DP UNIT VECTOR OF VVEC
#       UN              * +1            *       DP UNIT VECTOR OF UR1*U2
#       MAGVEC2         * +7 FOR EARTH  *       DP MAGNITUDE OF VVEC
#                       * +5 FOR MOON   *
#
#       PARAMETERS OF NO USE -
#               SP PARAMETERS - RTNAPSE, GEOMSGN, RTNPRM, PLUS PUSHLIST LOCATIONS 0-5, 10D-11D, 14D-21D, 31D-38D.
#               ADDITIONAL INTERPRETIVE SWITCHES USED - NORMSW

                SETLOC  CONICS
```

```
                BANK

                COUNT*  $$/CONIC
                EBANK=  UR1
KEPLERN         SETPD   BOV
                        0
                        +1
                VLOAD*
                        MUTABLE,1
                STOVL   14D
                        RRECT
                UNIT    SSP
                        ITERCTR
                        20D
                STODL   URRECT
                        36D
                STOVL   R1
                        RRECT
                DOT     SL1R
                        VRECT
                DMP     SL1R
                        1/ROOTMU        # 1/ROOTMU (-17 OR -14)
                STOVL   KEPC1           # C1=R.V/ROOTMU (+17 OR +16)
                        VRECT
                VSQ     DMPR
                        1/MU            # 1/MU (-34 OR -28)
                DMP     SL3
                        R1
                DSU     ROUND
                        D1/64
                STORE   KEPC2           # C2=RV.V/MU -1 (+6)

                BDSU    SR1R
                        D1/64
                DDV
                        R1
                STORE   ALPHA           # ALPHA=(1-C2)/R1 (-22 OR -20)

                BPL     DLOAD           # MAXIMUM X DEPENDS ON TYPE OF CONIC
                        1REV
                        -50SC           # -50SC (+12)
                DDV     BOV
                        ALPHA
                        STOREMAX
                SQRT    GOTO
                        STOREMAX

1REV            SQRT    BDDV
```

```
                        2PISC           # 2PISC (+6)
                BOV
                        STOREMAX
STOREMAX        STORE   XMAX
                DMP     PDDL
                        1/ROOTMU
                        ALPHA
                NORM    PDDL
                        X1
                SL*     DDV
                        0 -6,1
                BOV     BMN
                        MODDONE
                        MODDONE         # MPAC=PERIOD
PERIODCH        PDDL    ABS             # 0D=PERIOD
                        TAU.
                DSU     BMN
                        0D
                        MODDONE
                SIGN
                        TAU.
                STODL   TAU.
                GOTO
                        PERIODCH
MODDONE         SETPD   DLOAD
                        0
                        XKEPNEW
                STORE   X
                SIGN    BZE
                        TAU.
                        BADX
                BMN     ABS
                        BADX
                DSU     BPL
                        XMAX
                        BADX
STORBNDS        DLOAD   BPL
                        TAU.
                        STOREMIN
                DLOAD   DCOMP
                        XMAX
                STODL   XMIN
                        KEPZERO
                STORE   XMAX
                GOTO
                        DXCOMP
STOREMIN        DLOAD
                        KEPZERO
                STORE   XMIN
DXCOMP          DLOAD   DMPR
```

```
                                TAU.
                                BEE22
                ABS
                STODL           EPSILONT
                                XPREV
XDIFF           BDSU
                                X
                STORE           DELX

KEPLOOP         DLOAD           DSQ
                                X               # X=XKEP
                NORM            PUSH            # OD=XSQ (+34 OR +32 -N1)        PL AT 2
                                X1
                DMP             SRR*
                                ALPHA
                                0 -6,1
                STCALL          XI              # XI=ALPHA XSQ (+6)
                                DELTIME
                BOV             BDSU
                                TIMEOVFL        # UNLIKELY
                                TAU.
                STORE           DELT            # DELT=DELINDEP
                ABS             BDSU
                                EPSILONT
                BPL             DLOAD
                                KEPCONVG
                                T
                DSU             NORM
                                TC
                                X1
                PDDL            NORM
                                DELX
                                X2
                XSU,1           DMP
                                X2
                                DELT
                SLR*            DDV
                                1,1
                SR1             PUSH            # OD=TRIAL DELX          PL AT 2
                BPL             DLOAD
                                POSDELX
                                X
                STORE           XMAX            # MOVE MAX BOUND IN
                BDSU            DSU             #                        PL AT 0
                                XMIN
                BOV             BPL
                                NDXCHNGE
                                NDXCHNGE
                DLOAD           GOTO
```

```
                                OD
                                NEWDELX

        NDXCHNGE        DLOAD   DSU
                                XMIN
                                X
                        DMPR    GOTO            # TO FORCE MPAC +2 TO ZERO
                                DP9/10
                                NEWDELX

        POSDELX         DLOAD
                                X
                        STORE   XMIN            # MOVE MIN BOUND IN
                        BDSU    DSU             #                       PL AT 0
                                XMAX
                        BOV     BMN
                                PDXCHNGE
                                PDXCHNGE
                        DLOAD

                                OD
        NEWDELX         STORE   DELX
                        BZE     DAD
                                KEPCONVG
                                X
                        STODL   X
                                T
                        STORE   TC
        BRNCHCTR        RTB     BHIZ
                                CHECKCTR
                                KEPCONVG
                        GOTO

                                KEPLOOP         # ITERATE

        PDXCHNGE        DLOAD   DSU
                                XMAX
                                X
                        DMPR    GOTO            # TO FORCE MPAC +2 TO ZERO
                                DP9/10
                                NEWDELX

        BADX            DLOAD   SR1
                                XMAX
                        SIGN

                                TAU.
                        STORE   X
                        GOTO
```

```
                                STORBNDS
TIMEOVFL        DLOAD           BMN             # X WAS TOO BIG
                                X
                                NEGTOVFL
                STORE           XMAX
CMNTOVFL        DLOAD           SR1
                                DELX
                STORE           DELX
                BZE             BDSU
                                KEPRTN
                                X
                STODL           X
                                TC
                STORE           T
                GOTO
                                BRNCHCTR
NEGTOVFL        STORE           XMIN
                GOTO
                                CMNTOVFL
KEPCONVG        DLOAD           SR4R
                                R1
                DSU             VXSC
                                XSQC(XI)
                                URRECT
                VSL1            PDDL            # OD=(R1-XSQC(XI))URRECT (+33 OR +31)
                                X
                DSQ             NORM
                                X1
                DMPR            DMPR
                                1/ROOTMU
                                X
                DMP             SRR*
                                S(XI)
                                0 -7,1
                BDSU
                                T
                SL1             VXSC
                                VRECT
                VSL1            VAD             #                               PL AT 0
                VSL4
                STORE           RCV             # RCV (+29 OR +27)

                ABVAL           NORM
                                X2
                STODL           RCNORM
                                XI
                DMPR            DSU
                                S(XI)
                                D1/128
```

```
                DMP     SL1R
                        ROOTMU
                DMP     SLR*
                        X
                        0 -3,2
                DDV     VXSC
                        RCNORM
                        URRECT
                VSL1    PDDL            # OD=URRECT(XI S(XI)-1)X ROOTMU/RCV (+15
                        XSQC(XI)        # OR +13)                        PL AT 6
                SLR*    DDV
                        0 -4,2
                        RCNORM
                BDSU    VXSC
                        D1/256
                        VRECT
                VAD     VSL8
                STADR                   #                                PL AT 0
                STODL   VCV             # VCV (+7 OR +5)
                        T
                STODL   TC
                        X
                STORE   XPREV
                GOTO
                        KEPRTN
```

```
DELTIME         EXIT                        # MPAC=XI (+6), OD=XSQ (+34 OR +32 -N1)
                TC      POLY
                DEC     8
                2DEC    .083333334
                2DEC    -.266666684
                2DEC    .406349155
                2DEC    -.361198675
                2DEC    .210153242
                2DEC    -.086221951
                2DEC    .026268812
                2DEC    -.006163316
                2DEC    .001177342
                2DEC    -.000199055

                TC      INTPRET
                STODL   S(XI)
                        XI
                EXIT
                TC      POLY
                DEC     8
                2DEC    .031250001
                2DEC    -.166666719
                2DEC    .355555413
                2DEC    -.406347410
                2DEC    .288962094
                2DEC    -.140117894
                2DEC    .049247387
                2DEC    -.013081923
                2DEC    .002806389
                2DEC    -.000529414
                TC      INTPRET
```

```
                DMP     SRR*            #                               PL AT 0
                        OD
                        0 -5,1
                STORE   XSQC(XI)        # XSQC(XI) (+33 OR +31)
                DMP     SL1
                        KEPC1
                RTB     PDDL            # XCH WITH PL. OD=C1 XSQ C(XI) (+49 OR +46)
                        TPMODE          #                               PL AT 0,3
                DMP     SRR*
                        S(XI)
                        0 -5,1
                DMP     SL1
                        KEPC2
                RTB     PDDL            # 3D=C2 XSQ S(XI) (+35 OR +33)  PL AT 6
                        TPMODE
                        R1
                SR      TAD             #                               PL AT 3
                        6
                NORM    DMP             # TO PRESERVE SIGNIF.
                        X1
                        X
                SR*     TAD             # X(C2 XSQ S(XI) +R1) (+49 OR +46)  PL AT 0
                        0 -3,1
                SL4R    DMPR
                        1/ROOTMU
                STORE   T
                RVQ
```

```
ITERATOR        BONCLR  DLOAD
                        SLOPESW
                        FIRSTIME
                        DEP
                DSU     NORM
                        DEPREV
                        X1
                PDDL    NORM
                        DELINDEP
                        X2
                XSU,1   DMP
                        X2
                        DELDEP
                SLR*    DDV             #                               PL UP 2
                        1,1
                SR1     BOFF
                        ORDERSW
                        SGNCHECK
                ABS     SIGN            # IN CASE 2ND DERIV. CHANGED SIGN, MUST
                        DELDEP          # DISREGARD IT TO FIND MIN.

SGNCHECK        PUSH    BPL             # TRIAL DELINDEP                PL DOWN 2
                        POSDEL
                DLOAD   BON
                        INDEP
                        ORDERSW
                        MINCHECK
                STORE   MAX             # IF NOT 2ND ORDER, CAN MOVE MAX BOUND IN.

MINCHECK        BDSU    DSU
                        MIN
                BOV     BPL
                        MODNGDEL
                        MODNGDEL
                GOTO
                        DELOK

MODNGDEL        DLOAD   DSU             # TRIAL DELINDEP WOULD EXCEED MIN BOUND
                        MIN
                        INDEP
                DMP     GOTO
                        DP9/10
                        NEWDEL

FIRSTIME        DLOAD   DMP
                        MIN
                        TWEEKIT         # DLOAD TWEEKIT(40D) SENSITIVE TO CHANGE.
                PDDL    DMP             # S2(41D) SHOULDNT CONTAIN HI ORDER ONES
```

```

                                        MAX
                                        TWEEKIT
                                DSU
                                SIGN    GOTO
                                        DELDEP
                                        SGNCHECK

            POSDEL              DLOAD   BON
                                        INDEP
                                        ORDERSW
                                        MAXCHECK
                                STORE   MIN                # IF NOT 2ND ORDER, CAN MOVE MIN BOUND IN.

            MAXCHECK            BDSU    DSU
                                        MAX
                                BOV     BMN
                                        MODPSDEL
                                        MODPSDEL
            DELOK               DLOAD
                                        OD
            NEWDEL              STORE   DELINDEP
                                RVQ

            MODPSDEL            DLOAD   DSU
                                        MAX
                                        INDEP
                                DMP     GOTO
                                        DP9/10
                                        NEWDEL

            CHECKCTR            CS      ONE
                                INDEX   FIXLOC
                                AD      ITERCTR
                                INDEX   FIXLOC
                                TS      ITERCTR
                                TS      MPAC
                                TC      DANZIG
```

```
NEWSTATE        DLOAD   SR4R
                        R1
                DSU     VXSC
                        XSQC(XI)
                        UR1
                VSL1    PDDL            # OD=(R1-XSQC(XI))UR1 (+33 OR 31) PL AT 6
                        X
                DSQ     NORM
                        X1
                DMPR    DMPR
                        1/ROOTMU
                        X
                DMP     SRR*
                        S(XI)
                        0 -7,1
                BDSU
                        T
                SL1     VXSC
                        VVEC
                VSL1    VAD             #                               PL AT 0
                VSL4    PUSH
                ABVAL
LAMENTER        NORM
                        X1
                STODL   R2
                        XI
                DMP     DSU
                        S(XI)
                        D1/128
                DMP     SL1R
                        ROOTMU
                DMP     SLR*
                        X
                        0 -3,1
                DDV     VXSC
                        R2
                        UR1
                VSL1    PDDL            # 6D=V2VEC PART (+15 OR 13)     PL AT 12
                        XSQC(XI)
                SLR*    DDV
                        0 -4,1
                        R2
                BDSU
                        D1/256
                VXSC    VAD             #                               PL AT 6
                        VVEC
                VSL8    RVQ
```

```
                SETLOC  CONICS1
                BANK

                COUNT*  $$/CONIC
# DO NOT DISTURB THE ORDER OF THESE CDS, OVERLAYS HAVE BEEN MADE.
BEE17           DEC     0               # KEEP WITH D1/8 2DEC 1.0B-17 (0000004000)
D1/8            2DEC    1.0 B-3
D1/128          2DEC    1.0 B-7
D1/64           2DEC    1.0 B-6
D1/4            2DEC    1.0 B-2
D1/16           2DEC    1.0 B-4
D1/32           2DEC    1.0 B-5
D1/1024         2DEC    1.0 B-10
D1/256          2DEC    1.0 B-8
DP9/10          2DEC    .9
KEPZERO         EQUALS  LO6ZEROS
-50SC           2DEC    -50.0 B-12
2PISC           2DEC    6.28318530 B-6
BEE19           EQUALS  D1/32 -1        # 2DEC 1.0 B-19 (00000 01000)
BEE22           EQUALS  D1/256 -1       # 2DEC 1.0 B-22 (00000 00100)
ONEBIT          2DEC    1.0 B-28
COGUPLIM        2DEC    .999511597
COGLOLIM        2DEC    -.999511597
```

```
                SETLOC  CONICS
                BANK

                COUNT*  $$/CONIC
TIMETHET        STQ     SETPD           #                               PL AT 0
                        RTNTT
                        0
                BOV
                        +1
                VLOAD   PDVL            # SETUP FOR PARAM CALL          PL AT 6
                        RVEC
                        VVEC
                CALL
                        PARAM
                BOV     CALL            #                               PL AT 0
                        COGAOVFL
                        GETX
COMMNOUT        DLOAD   BON
                        XI
                        INFINFLG
                        ABTCONIC
                CLEAR   CALL
                        COGAFLAG
                        DELTIME
                BON     CALL
                        RVSW
                        RTNTT
                        NEWSTATE
                GOTO
                        RTNTT

COGAOVFL        SETGO
                        COGAFLAG
                        ABTCONIC

                BANK    4
                SETLOC  CONICS1
                BANK
                COUNT*  $$/CONIC
PARAM           STQ     CLEAR           # MPAC=V1VEC, OD=R1VEC          PL AT 6
                        RTNPRM
                        NORMSW
                CLEAR
                        COGAFLAG
                SSP     CALL
                        GEOMSGN
                        37777           # GAMMA ALWAYS LESS THAN 180DEG
                        GEOM            # MPAC=SNGA (+1), OD=CSGA (+1)  PL AT 2
                STODL   36D             # 36D=SIN GAMMA (+1)            PL AT 0
                SR      DDV
```

```
                                5
                                36D
                STOVL*          COGA
                                MUTABLE,1
                STODL           1/MU
                                MAGVEC2
                DSQ             NORM
                                X1
                DMPR            DMP
                                1/MU
                                R1
                SRR*
                                0 -3,1
                PUSH            BDSU            # OD=R1 V1SQ/MU (+6)            PL AT 2
                                D1/32
                STODL           R1A             # R1A (+6)                      PL AT 0

                DMP             NORM
                                36D
                                X1
                DMP             SR*
                                36D
                                0 -4,1
                STORE           P               # P (+4)
                GOTO
                                RTNPRM
```

```
GEOM            UNIT                    # MPAC=V2VEC, OD=R1VEC          PL AT 6
                STODL   U2              # U2 (+1)
                        36D
                STOVL   MAGVEC2         #                               PL AT 0
                UNIT
                STORE   UR1             # UR1 (+1)
                DOT     SL1
                        U2
                PDDL                    # OD=CSTH (+1)                  PL AT 2
                        36D
                STOVL   R1              # R1 (+29 OR +27)
                        UR1
                VXV     VSL1
                        U2
                BON     SIGN
                        NORMSW
                        HAVENORM
                        GEOMSGN
                UNIT    BOV
                        COLINEAR
UNITNORM        STODL   UN              # UN (+1)
                        36D
                SIGN    RVQ             # MPAC=SNTH (+1), 34D=SNTH.SNTH (+2)
                        GEOMSGN

COLINEAR        VSR1    GOTO
                        UNITNORM

HAVENORM        ABVAL   SIGN
                        GEOMSGN
                RVQ                     # MPAC=SNTH (+1), 34D=SNTH.SNTH (+2)
```

```
                BANK    12
                SETLOC  CONICS
                BANK

                COUNT*  $$/CONIC
GETX            AXT,2   SSP             # ASSUMES P (+4) IN MPAC
                        3
                        S2
                        1
                CLEAR
                        360SW
                SQRT    PDDL            # OD=SQRT(P)                    PL AT 2
                        CSTH
                SR1     BDSU
                        D1/4
                PDDL    SRR             #                               PL AT 4D
                        SNTH
                        6
                DDV                     #                               PL AT 2
                BOV
                        360CHECK
                DSU     DMP
                        COGA            #                               PL AT 0
                SL2R    BOV
                        360CHECK
WLOOP           PUSH    DSQ             # OD=W (+5)                     PL AT 2
                TLOAD   PDDL            # 2D=WSQ (+10)                  PL AT 5
                        MPAC
                        R1A
                SR4     TAD             #                               PL AT 2
                BMN     SQRT
                        INFINITY
                ROUND   DAD             #                               PL AT OD
                BOV     TIX,2
                        RESETX2
                        WLOOP

                BDDV    BOV
                        D1/128
                        INFINITY
POLYCOEF        BMN     PUSH            # OD=1/W (+2) OR 16/W (+6)      PL AT 2
                        INFINITY
                DSQ
                NORM    DMP
                        X1
                        R1A
                SRR*    EXIT
                        0 -10D,1
                TC      POLY
```

```
                DEC     5
                2DEC    .5
                2DEC    -.166666770
                2DEC    .100000392
                2DEC    -.071401086
                2DEC    .055503292
                2DEC    -.047264098
                2DEC    .040694204

                TC      INTPRET
                DMP     SL1R            #                               PL AT OD
                PUSH    BON
                        360SW
                        TRUE360X
XCOMMON         DSQ     NORM
                        X1
                DMP     SRR*
                        R1A
                        0 -12D,1
                STODL   XI              # XI (+6)
                        R1
                SR1     SQRT
                ROUND   DMP
                SL4R                    #                               PL AT 0
                STORE   X               # X (+17 OR +16)

                DSQ     NORM
                        X1
                PDDL    DMP             # OD=XSQ (+34 OR +32 -N1)       PL AT 2
                        P
                        R1
                SL3     SQRT
                DMP     SL3R
                        COGA
                STODL   KEPC1
                        R1A
                BDSU    CLEAR
                        D1/64
                        INFINFLG
                STORE   KEPC2
                RVQ
```

```
RESETX2         AXT,2
                        3

360CHECK        SETPD   BPL
                        OD
                        INVRSEQN
                SET
                        360SW

INVRSEQN        DLOAD   SQRT
                        P
                PDDL    DMP             # OD=SQRT(P) (+2)               PL AT 2
                        SNTH
                        COGA
                SL1     PDDL            # 2D=SNTH COGA (+5)             PL AT 4
                        CSTH
                SR4     DAD
                        D1/32
                DSU     DMP             #                               PL AT 2,0
                NORM    BDDV
                        X1
                        SNTH
                SLR*    ABS             # NOTE: NEAR 360 CASE TREATED DIFFERENTLY
                        0 -5,1
                PUSH    DSQ             # OD=1/W (-1)                   PL AT 2
                STODL   34D
                        D1/16
1/WLOOP         PUSH    DSQ             # 2D=G (+4)                     PL AT 4
                RTB     PDDL            #                               PL AT 7
                        TPMODE
                        R1A
                DMP     SR4
                        34D
                TAD                     #                               PL AT 4
                BMN     SQRT
                        INFINITY
                DAD                     #                               PL AT 2
                TIX,2   NORM
                        1/WLOOP
                        X1
                BDDV
                SLR*    GOTO            #                               PL AT 0
                        0 -7,1
                        POLYCOEF

TRUE360X        DLOAD   BMN
                        R1A
```

```
                                INFINITY
                        SQRT    NORM
                                X1
                        BDDV    SL*
                                2PISC
                                0 -3,1
                        DSU     PUSH            # OD=2PI/SQRT(R1A) -X           PL AT 0,2
                        GOTO
                                XCOMMON
        INFINITY        SETPD   BOV             # NO SOLUTION EXISTS SINCE CLOSURE THROUGH
                                0               # INFINITY IS REQUIRED
                                OVFLCLR
        OVFLCLR         SET     RVQ
                                INFINFLG
```

```
LAMBERT         STQ     SETPD
                        RTNLAMB
                        OD
                BOV
                        +1
                CLEAR   VLOAD*
                        SOLNSW
                        MUTABLE,1
                STODL   1/MU
                        TDESIRED
                DMPR
                        BEE19
                STORE   EPSILONL
                SET     VLOAD
                        SLOPESW
                        R1VEC
                PDVL    CALL            # OD=R1VEC (+29 OR +27)          PL AT 6
                        R2VEC           # MPAC=R2VEC (+29 OR +27)
                        GEOM
                STODL   SNTH            # OD=CSTH (+1)                   PL AT 2
                        MAGVEC2
                NORM    PDDL
                        X1
                        R1
                SR1     DDV             #                                PL AT 2
                SL*     PDDL            # DXCH WITH OD, OD=R1/R2 (+7)    PL AT 0,2
                        0 -6,1
                STADR
                STORE   CSTH            # CSTH (+1)
                SR1     BDSU
                        D1/4
                STORE   1-CSTH          # 1-CSTH (+2)

                ROUND   BZE
                        360LAMB
                NORM    PDDL            #                                PL AT 4
                        X1
                        OD
                SR1     DDV             #                                PL AT 2
                SL*     SQRT
                        0 -3,1
                PDDL    SR              # 2D=SQRT(2R1/R2(1-CSTH)) (+5)   PL AT 4
                        SNTH
                        6
                DDV     DAD             #                                PL AT 2
                        1-CSTH
                STADR
                STORE   COGAMAX
                BOV     BMN             # IF OVFL, COGAMAX=COGUPLIM
                        UPLIM           # IF NEG, USE EVEN IF LT COGLOLIM, SINCE
```

```
                        MAXCOGA         #       THIS WOULD BE RESET IN LAMBLOOP
                DSU     BMN             # IF COGAMAX GT COGUPLIM, COGAMAX=COGUPLIM
                        COGUPLIM
                        MAXCOGA         # OTHERWISE OK, SO GO TO MAXCOGA
UPLIM           DLOAD
                        COGUPLIM        # COGUPLIM=.999511597 = MAX VALUE OF COGA
                STORE   COGAMAX         #       NOT CAUSING OVFL IN R1A CALCULATION
MAXCOGA         DLOAD
                        CSTH
                SR      DSU             #                               PL AT 0
                        6
                STADR
                STODL   CSTH-RHO
                        GEOMSGN
                BMN     DLOAD
                        LOLIM
                        CSTH-RHO
                SL1     DDV
                        SNTH
                BOV
                        LOLIM
MINCOGA         STORE   COGAMIN         # COGAMIN (+5)
                BON     SSP
                        GUESSW
                        NOGUESS
                        TWEEKIT
                        00001
                DLOAD
                        COGA

LAMBLOOP        DMP
                        SNTH
                SR1     DSU
                        CSTH-RHO
                NORM    PDDL            # OD=SNTH COGA-(CSTH-RHO) (+7+C(X1)) PL=2
                        X1
                        1-CSTH
                SL*     DDV             # 1-CSTH (+2)                   PL AT 0
                        0 -9D,1
                BMN     BZE
                        NEGP
                        NEGP
                STODL   P               # P=(1-CSTH)/(SNTH COGA-(CSTH-RHO)) (+4)
                        COGA
                DSQ     DAD
                        D1/1024
                NORM    DMP
                        X1
                        P
```

```
                SR*     BDSU
                        0 -8D,1
                        D1/32
                STODL   R1A             # R1A=2-P(1+COGA COGA) (+6)
                        P
                BOV     CALL
                        HIENERGY
                        GETX
                DLOAD
                        T
                STODL   TPREV
                        XI
                BON     CALL
                        INFINFLG
                        NEGP            # HAVE EXCEEDED THEORETICAL BOUNDS
                        DELTIME
                BOV     BDSU
                        BIGTIME
                        TDESIRED
                STORE   TERRLAMB
                ABS     BDSU
                        EPSILONL
                BPL     RTB
                        INITV
                        CHECKCTR
                BHIZ    CALL
                        SUFFCHEK
                        ITERATOR
                DLOAD   BZE
                        MPAC
                        SUFFCHEK
                DAD
                        COGA
                STORE   COGA
                GOTO
                        LAMBLOOP

NEGP            DLOAD   BPL             # IMPOSSIBLE TRAJECTORY DUE TO INACCURATE
                        DCOGA           # BOUND CALCULATION. TRY NEW COGA.
                        LOENERGY

HIENERGY        SETPD   DLOAD           # HIGH ENERGY TRAJECTORY RESULTED
                        0
                        COGA            # IN OVFL OF P OR R1A, OR XI EXCEEDING 50.
                STORE   COGAMIN         # THIS IS THE NEW BOUND.
COMMONLM        DLOAD   SR1
                        DCOGA
```

```
                STORE   DCOGA           # USE DCOGA/2 AS DECREMENT
                BZE     BDSU
                        SUFFCHEK
                        COGA
                STORE   COGA
                GOTO                    # RESTART THIS LOOP
                        LAMBLOOP

BIGTIME         DLOAD
                        TPREV
                STORE   T

LOENERGY        SETPD   DLOAD           # LOW ENERGY TRAJECTORY RESULTED
                        0
                        COGA            # IN OVERFLOW OF TIME.
                STORE   COGAMAX         # THIS IS THE NEW BOUND.
                GOTO
                        COMMONLM

SUFFCHEK        DLOAD   ABS
                        TERRLAMB
                PDDL    DMP             #                               PL AT 2D
                        TDESIRED
                        BEE17
                DAD     DSU             #                               PL AT 0D
                        ONEBIT
                BPL     SETGO
                        INITV
                        SOLNSW
                        INITV
360LAMB         SETPD   SETGO           # LAMBERT CANNOT HANDLE CSTH=1
                        0
                        SOLNSW
                        RTNLAMB

NOGUESS         SSP     DLOAD
                        TWEEKIT
                        20000
                        COGAMIN
                SR1     PDDL            #                               PL AT 2
                        COGAMAX
                SR1     DAD
                STADR                   #                               PL AT 0
                STORE   COGA
                STORE   DCOGA
                GOTO
```

```
                        LAMBLOOP
LOLIM           DLOAD   GOTO
                        COGLOLIM        # COGLOLIM=-.999511597
                        MINCOGA
INITV           DLOAD   NORM
                        R1
                        X1
                PDDL    SR1             #                               PL AT 2
                        P
                DDV                     #                               PL AT 0
                SL*     SQRT
                        0 -4,1
                DMP     SL1
                        ROOTMU
                PUSH    DMP             # 0D=VTAN (+7)                  PL AT 2
                        COGA
                SL      VXSC
                        5
                        UR1
                PDDL                    # XCH WITH 0D                   PL AT 0,6
                VXSC    VSL1
                        UN
                VXV     VAD             #                               PL AT 0
                        UR1
                VSL1
                STORE   VVEC
                SLOAD   BZE
                        VTARGTAG
                        TARGETV
                GOTO
                        RTNLAMB
TARGETV         DLOAD   CALL
                        MAGVEC2
                        LAMENTER
                STORE   VTARGET
                GOTO
                        RTNLAMB
```

```
TIMERAD         STQ     SETPD           #                               PL AT 0
                        RTNTR
                        0
                BOV
                        +1
                VLOAD   PDVL            #                               PL AT 6
                        RVEC
                        VVEC
                CALL
                        PARAM
                BOV     DLOAD           #                               PL AT 0
                        COGAOVFL
                        D1/32
                DSU     DMP
                        R1A
                        P
                SQRT    DMP
                        COGA
                SL4     VXSC
                        U2
                PDDL    DSU             #                               PL AT 6
                        D1/64
                        R1A
                VXSC    VSU             #                               PL AT 0
                        UR1
                VSL4    UNIT
                BOV
                        CIRCULAR
                PDDL    NORM            # 0D=UNIT(ECC) (+3)             PL AT 6
                        RDESIRED        # 35D=ECC (+3)
                        X1
                PDDL    DMP             #                               PL AT 8
                        R1
                        P
                SL*     DDV             #                               PL AT 6
                        0,1
                DSU     DDV
                        D1/16
                        36D             # 36D=ECC (+3)
                STORE   COSF
                BOV     DSQ
                        BADR2
                BDSU    BMN
                        D1/4
                        BADR2
                SQRT    SIGN
                        SGNRDOT
                CLEAR
                        APSESW
```

```
TERMNVEC        VXSC    VSL1
                        UN
                VXV     PDVL            # VXCH WITH OD                  PL AT 0,6
                        OD
                VXSC    VAD             #                               PL AT 0
                        COSF
                VSL1    PUSH            # OD=U2                         PL AT 6
                DOT     DDV             # LIMITS RESULT TO POSMAX OR NEGMAX
                        UR1
                        DP1/4
                SR1     BOV             # SCALE BACK DOWN TO NORMAL
                        +1              # CLEAR OVFIND IF SET
                STOVL   CSTH            # CSTH (+1)
                        UR1
                VXV     VSL1
                DOT     SL1
                        UN
                STODL   SNTH            # SNTH (+1)
                        P
                CALL
                        GETX
                CLRGO
                        SOLNSW
                        COMMNOUT
CIRCULAR        SETPD   SETGO
                        0
                        SOLNSW
                        ABTCONIC

BADR2           DLOAD   SIGN
                        LODPHALF
                        COSF
                STODL   COSF
                        KEPZERO
                SETGO
                        APSESW
                        TERMNVEC
```

```
APSIDES         STQ     SETPD           #                               PL AT 0
                        RTNAPSE
                        OD
                BOV
                        +1
                VLOAD   PDVL            #                               PL AT 6
                        RVEC
                        VVEC
                CALL
                        PARAM
                BOV                     #                               PL AT 0
                        GETECC
GETECC          DMP     SL4
                        R1A
                BDSU    SQRT
                        D1/64
                STORE   ECC
                DAD     PDDL            #                               PL AT 2
                        D1/8
                        R1
                DMP     SL1
                        P
                DDV                     #                               PL AT 0
                PDDL    NORM            # OD=RP (+29 OR +27)            PL AT 2
                        R1A
                        X1
                PDDL    SL*             #                               PL AT 4
                        R1
                        0 -5,1
                DDV     DSU             #                               PL AT 2,0
                BOV     BMN
                        INFINAPO
                        INFINAPO
                GOTO
                        RTNAPSE
INFINAPO        DLOAD   GOTO            # RETURNS WITH APOAPSIS IN MPAC, PERIAPSIS
                        LDPOSMAX
                        RTNAPSE         # THAT PL IS AT 0.
```

```
ABTCONIC        EXIT
                TC      POODOO
                OCT     00607
```

```
LDPOSMAX        EQUALS  LODPMAX         # DPPOSMAX IN LOW MEMORY.

# ERASABLE ASSIGNEMENTS

# KEPLER SUBROUTINE

#       INPUT -
# RRECT         ERASE   +5
# VRECT         ERASE   +5
# TAU.          ERASE   +1
# XKEP          ERASE   +1
# TC            ERASE   +1
# XPREV         ERASE   +1
1/MU            EQUALS  14D
ROOTMU          EQUALS  16D
1/ROOTMU        EQUALS  18D

#       OUTPUT -
# RCV           ERASE   +5
# VCV           ERASE   +5
# RC            ERASE   +1
# XPREV         ERASE   +1

#       DEBRIS -
ALPHA           EQUALS  8D
XMAX            EQUALS  10D
XMIN            EQUALS  12D
X               EQUALS  20D
XI              EQUALS  24D
S(XI)           EQUALS  26D
XSQC(XI)        EQUALS  28D
T               EQUALS  30D
R1              EQUALS  32D
KEPC1           EQUALS  34D
KEPC2           EQUALS  36D
# DELX          ERASE   +1
# DELT          ERASE   +1
# URRECT        ERASE   +5
# RCNORM        ERASE   +1
# XPREV         EQUALS  XKEP


# LAMBERT SUBROUTINE
#
#       INPUT -
# R1VEC         ERASE   +5
# R2VEC         ERASE   +5
# TDESIRED      ERASE   +1
# GEOMSGN       ERASE   +0
# GUESSW                        # 0 IF COGA GUESS AVIABLE, 1 IF NOT
```

```
# COGA          ERASE   +1      # INPUT ONLY IF GUESS IS ZERO.
# NORMSW                        # 0 IF UN TO BE COMPUTED, 1 IF UN INPUT
# UN            ERASE   +5      # ONLY USED IF NORMSW IS 1
# VTARGTAG      ERASE   +0
# TWEEKIT       EQUALS  40D     # ONLY USED IF GUESSW IS 0

#       OUTPUT -
# VTARGET       ERASE   +5      # AVAILABLE ONLY IF VTARGTAG IS ZERO.
# V1VEC         EQUALS  MPAC

#       DEBRIS -
# RTNLAMB       ERASE   +0
# U2            ERASE   +5
# MAGVEC2       ERASE   +1
# UR1           ERASE   +5
# R1            EQUALS  31D
# UN            ERASE   +5
# SNTH          ERASE   +1
# CSTH          ERASE   +1
# 1-CSTH        ERASE   +1
# CSTH-RHO      ERASE   +1
COGAMAX         EQUALS  14D     # CLOBBERS 1/MU
COGAMIN         EQUALS  8D
DCOGA           EQUALS  12D
# TWEEKIT       EQUALS  40D
# P             ERASE   +1
# COGA          ERASE   +1
# R1A           ERASE   +1
# X             EQUALS  20D
# XSQ           EQUALS  22D
# XI            EQUALS  24D
# S(XI)         EQUALS  26D
# XSQC(XI)      EQUALS  28D
# T             EQUALS  30D
# KEPC1         EQUALS  34D
# KEPC2         EQUALS  36D
# SLOPESW
# SOLNSW

#       OTHERS -
# RVEC          EQUALS  R1VEC
# VVEC          ERASE   +5
# COGAFLAG
# RVSW
# INFINFLG
# APSESW
# 360SW
# RTNTT         EQUALS  RTNLAMB
# ECC           ERASE   +1
# RTNTR         EQUALS  RTNLAMB
```

```
# RTNAPSE         EQUALS  RTNLAMB
# R2              EQUALS  MAGVEC2
COSF              EQUALS  24D
# RTNPRM          ERASE   +0
# SCNRDOT         ERASE   +0
# RDESIRED        ERASE   +1


# ITERATOR SUBROUTINE
# ORDERSW
MAX               EQUALS  14D               # CLOBBERS 1/MU
MIN               EQUALS  8D
# INDEP           ERASE   +1
DELINDEP          EQUALS  12D
ITERCTR           EQUALS  22D
DEP               EQUALS  30D
# DELDEP          ERASE   +1
# DEPREV          ERASE   +1
TWEEKIT           EQUALS  40D


# MORE KEPLER
# EPSILONT        ERASE   +1


# MORE LAMBERT
# TERRLAMB        EQUALS  DELDEP
# TPREV           EQUALS  DEPREV

# EPSILONL        EQUALS  EPSILONT +2      # DOUBLE PRECISION WORD
```

```
# 1.0 INTRODUCTION
# ----------------
#
# FROM A USER'S POINT OF VIEW, ORBITAL INTEGRATION IS ESSENTIALLY THE SAME AS THE 278 INTEGRATION
# PROGRAM.  THE SAME ENTRANCES TO THE PROGRAM WILL BE MAINTAINED, THE SAME STALLING ROUTINE WILL BE USED AND
# OUTPUT WILL STILL BE VIA THE PUSHLIST.  THE PRIMARY DIFFERENCES TO A USER INVOLVE THE ADDED CAPABILITY OF
# TERMINATING INTEGRATION AT A SPECIFIC FINAL RADIUS AND THE DIFFERENCE IN STATE VECTOR SCALING INSIDE AND OUTSIDE
# THE LUNAR SPHERE OF INFLUENCE.
#
# IN ORDER TO MAKE THE CSM(LEM)PREC AND CSM(LEM)CONIC ENTRANCES SIMILAR TO FLIGHT 278, THE INTEGRATION PROGRAM
# WILL ITSELF SET THE FINAL RADIUS (RFINAL) TO 0 SO THAT REACHING THE DESIRED TIME ONLY WILL TERMINATE
# INTEGRATION.  THE DP REGISTER RFINAL MUST BE SET BY USERS OF INTEGRVS AND INTEGRV, AND MUST BE DONE AFTER THE
# CALL TC INTSTALL.
#
# WHEN THE LM IS ON THE LUNAR SURFACE (INDICATED BY LUNAR SURFACE FLAG SET) CALLS TO LEMCONIC, LEMPREC, AND
# INTEGRV WITH VINFLAG = 0 WILL RESULT IN THE USE OF THE PLANETARY INERTIAL ORIENTATION SUBROUTINES TO PROVIDE
# BOTH THE LM'S POSITION AND VELOCITY IN THE REFERENCE COORDINATE SYSTEM.
# THE PROGRAM WILL PROVIDE OUTPUT AS IF INTEGRATION WAS USED.  THAT IS, THE PUSHLIST WILL BE SET AS NOTED BELOW AND
# THE PERMANENT STATE VECTOR UPDATED WHEN SPECIFIED BY AN INTEGRV CALL.
#
# USERS OF INTEGRVS DESIRING INTEGRATION (INTYPFLG = 0) SHOULD NOTE THAT THE OBLATENESS PERTURBATION COMPUTATION
# IN LUNAR ORBIT IS TIME DEPENDENT.  THEREFORE, THE USER SHOULD SUPPLY AN INITIAL STATE VECTOR VALID AT SOME REAL
# TIME AND THE DESIRED TIME (TDEC1) ALSO AT SOME REAL TIME.  FOR CONIC "INTEGRATION" THE USER MAY STILL USE ZERO
# AS THE INITIAL TIME AND DELTA TIME AS THE DESIRED TIME.
#
# 2.0 GENERAL DESCRIPTION
# -----------------------
#
# THE INTEGRATION PROGRAM OPERATES AS A CLOSED INTERPRETIVE SUBROUTINE AND PERFORMS THESE FUNCTIONS --
#       1) INTEGRATES (PRECISION OR CONIC) EITHER CSM OR LM STATE VECTOR
#       2) INTEGRATES THE W-MATRIX
#       3) PERMANENT OR TEMPORARY UPDATE OF THE STATE VECTOR
#
# THERE ARE SIX ENTRANCES TO THE INTEGRATION PROGRAM.  FOUR OF THESE (CSMPREC, LEMPREC, CSMCONIC, LEMCONIC) SET
# ALL THE FLAGS REQUIRED IN THE INTEGRATION PROGRAM ITSELF TO CAUSE THE PRECISION OR CONIC INTEGRATION (KEPLER) OF
# THE LM OR CSM STATE VECTOR, AS THE NAMES SUGGEST.  ONE ENTRANCE (INTEGRVS) PERMITS THE CALLING PROGRAM TO
# PROVIDE A STATE VECTOR TO BE INTEGRATED.  THE CALLING PROGRAM MUST SET THE FLAGS INDICATING (1) PRECISION OR
# CONIC INTEGRATION, (2) IN OR OUT OF LUNAR SPHERE, (3) MIDCOURSE OR NOT, AND THE INTEGRATION PROGRAM COMPLETES
# THE FLAG SETTING TO BYPASS W-MATRIX INTEGRATION.  THE LAST ENTRANCE (INTEGRV, USED IN GENERAL BY THE
# NAVIGATION PROGRAMS) PERMITS THE CALLER TO SET FIVE FLAGS (NOT MOONFLAG OR MIDFLAG) BUT NOT TO INPUT A STATE
# VECTOR.  ANY PROGRAM WHICH CALLS INTEGRVS OR INTEGRV MUST CALL INTSTALL BEFORE IT SETS THE INTEGRATION FLAGS
# AND/OR STATE VECTOR.
#
# THREE SETS OF 42 REGISTERS AND 2 FLAGS ARE USED FOR THE STATE VECTORS.  TWO SETS, WHICH MAY NOT BE OVERLAYED, ARE
# USED FOR THE PERMANENT STATE VECTORS FOR THE CSM AND LM.  THE THIRD SET, WHICH MAY BE OVERLAYED WHEN INTEGRATION
# IS NOT BEING DONE, IS USED IN THE COMPUTATIONS.
#
# THE PERMANENT STATE VECTORS WILL BE PERIODICALLY UPDATED SO THAT THE VECTORS WILL NOT BE OLDER THAN 4 TIMESTEPS.
# THE PERMANENT STATE VECTORS WILL ALSO BE UPDATED WHENEVER THE W-MATRIX IS INTEGRATED OR WHEN A CALLER OF INTEGRV
# SETS STATEFLG (THE NAVIGATION PROGRAMS P20, P22.)
#
```

```
# APPENDIX B OF THE USERS' GUIDE LISTS THE STATE VECTOR QUANTITIES.
#
# 2.1 RESTARTS
#
# PHASE CHANGES WILL BE MADE IN THE INTEGRATION PROGRAM ONLY FOR THE INTEGRV ENTRANCE (I.E., WHEN THE W-MATRIX IS
# INTEGRATED OR PERMANENT STATE VECTOR IS UPDATED.)  THE GROUP NUMBER USED WILL BE THAT FOR THE P20-25 PROGRAMS
# (I.E., GROUP2) WINCE THE INTEGRV ENTRANCE WILL ONLY BE USED BY THESE PROGRAMS.  IF A RESTART OCCURS DURING AN
# INTEGRATION OF THE STATE VECTOR ONLY, THE RECOVERY WILL BE TO THE LAST PHASE IN THE CALLING PROGRAM.  CALLING
# PROGRAMS WHICH USE THE INTEGRV OR INTEGRVS ENTRANCE OF INTEGRATION WHOULD ENSURE THAT IF PHASE CHANGING IS DONE
# THAT IT IS PRIOR TO SETTING THE INTEGRATION INPUTS IN THE PUSHLIST.
# THIS IS BECAUSE THE PUSHLIST IS LOST DURING A RESTART.
#
# 2.2 SCALING
#
# THE INTEGRATION ROUTINE WILL MAINTAIN THE PERMANENT MEMORY STATE VECTORS IN THE SCALING AND UNITS DEFINED IN
# APPENDIX B OF THE USERS GUIDE.  THE SCALING OF THE OUTPUT POSITION VECTOR DEPENDS ON THE ORIGIN OF THE COORDINATE
# SYSTEM AT THE DESIRED INTEGRATION TIME.  THE COORDINATE SYSTEM TRANSFORMATION WILL BE DONE AUTOMATICALLY ON
# MULTIPLE TIMESTEP ENCKE INTEGRATION ONLY.  THUS IT IS POSSIBLE TO HAVE OUTPUT FROM SUCCESSIVE INTEGRATIONS IN
# DIFFERENT SCALING.
# HOWEVER, RATT, VATT WILL ALWAYS BE SCALED THE SAME.
#
# 3.0 INPUT/OUTPUT
# ----------------
#
# PROGRAM INPUTS ARE THE FLAGS DESCRIBED IN APPENDIX A AND THE PERMANENT STATE VECTOR QUANTITIES DESCRIBED IN
# APPENDIX B OF THE USERS GUIDE, PLUS THE DESIRED TIME TO INTEGRATE TO IN TDEC1 (A PUSH LIST LOCATION).
# FOR INTEGRVS, THE RCV,VCV,TET OR THE TEMPORARY STATE VECTOR MUST BE SET, PLUS MOONFLAG AND MIDFLAG
#
# FOR SIMULATION THE FOLLOWING QUANTITIES MUST BE PRESET ---
#                                                                        EARTH   MOON
#                                                                         29      27
#      RRECTCSM(LEM)           RECTIFIED POSITION VECTOR       METERS    2       2
#
#                                                                         7       5
#      VRECTCSM(LEM)           RECTIFIED VELOCITY VECTOR       M/CSEC    2       2
#
#                                                                         28      28
#      TETCSM(LEM)             TIME STATE VECTOR IS VALID      CSEC      2       2
#                              CUSTOMARILY 0, BUT NOTE LUNAR
#                              ORBIT DEPENDENCE ON REAL TIME.
#
#                                                                         22      18
#      DELTAVCSM(LEM)          POSITION DEVIATION              METERS    2       2
#                              0 IF TCCSM(LEM) = 0
#
#                                                                         3       -1
#      NUVCSM(LEM)             VELOCITY DEVIATION              M/CSEC    2       2
#                              0 IF TCCSM(LEM) = 0
```

```

#                                                                          29      27
#       RCVSM(LEM)              CONIC POSITION              METERS        2       2
#                               EQUALS RRECTCSM(LEM) IF
#                               TCCSM(LEM) = 0
#
#                                                                          7       5
#       VCVCSM(LEM)             CONIC VELOCITY              M/CSEC        2       2
#                               EQUALS VRECTCSM(LEM) IF
#                               TCCSM(LEM) = 0
#
#                                                                          28      28
#       TCCSM(LEM)              TIME SINCE RECTIFICATION    CSECS         2       2
#                               CUSTOMARILY 0
#
#                                                            1/2           17      16
#       XKEPCSM(LEM)            ROOT OF KEPLER'S EQUATION   M             2       2
#                               0 IF TCCSM(LEM) = 0
#
#       CMOONFLG                PERMANENT FLAGS CORRESPONDING             0       0
#       CMIDFLAG                TO MOONFLAG AND MIDFLAG                   0,1     0,1
#       LMOONFLG                C = CSM, L = LM                           0       0
#       LMIDFLG                                                           0,1     0,1
#
#       SURFFLAG                LUNAR SURFACE FLAG                        0,1     0,1
#
# IN ADDITION, IF (L)CMIDFLAG IS SET, THE INITIAL INPUT VALUES FOR LUNAR
# SOLAR EPHEMERIDES SUBROUTINE AND PLANETARY INERTIAL ORIENTATION SUB-
# ROUTINE MUST BE PRESET.
#
# OUTPUT
#       AFTER EVERY CALL TO INTEGRATION
#                                                                 EARTH   MOON
#                                                                  29      29
#       0D      RATT    POSITION                        METERS    2       2
#
#                                                                  7       7
#       6D      VATT    VELOCITY                        M/CSEC    2       2
#
#                                                                  28      28
#       12D     TAT     TIME                                      2       2
#
#                                                                  29      27
#       14D     RATT1   POSITION                        METERS    2       2
#
#                                                                  7       5
#       20D     VATT1   VELOCITY                        M/CSEC    2       2
#
#                                                       3    2     36      30
#       26D     MU(P)   MU                              M /CS     2       2
#
#       X1              MUTABLE ENTRY                             -2      -10D
#
#       X2              COORDINT
#       X2              COORDINATE SYSTEM ORIGIN                  0       2
#                       (THIS, NOT MOONFLAG, SHOULD BE
```

```
#                      USED TO DETERMINE ORIGIN.)
#
# IN ADDITION TO THE ABOVE, THE PERMANENT STATE VECTOR IS UPDATED WHENEVER
# STATEFLG WAS SET AND WHENEVER A W-MATRIX IS TO BE INTEGRATED.  THE PUSH
# COUNTER IS SET TO 0 AND OVERFLOW IS CLEARED BEFORE RETURNING TO THE
# CALLING PROGRAM.
#
# 4.0 CALLING SEQUENCES AND SAMPLE CODE
# -------------------------------------
#
#       A) PRECISION ORBITAL INTEGRATION.  CSMPREC, LEMPREC ENTRANCES
#            L-X      STORE TIME TO 96T5791T5 T 95 PUS L9ST (T4531)
#            L        CALL
#            L+1              CSMPREC (OR LEMPREC)
#            L+2      RETURN
#          INPUT                                                     28
#            TDEC1 (PD 32D) TIME TO INTEGRATE TO...CENTISECONDS SCALED 2
#          OUTPUT
#            THE DATA LISTED IN SECTION 3.0 PLUS
#            RQVV     POSITION VECTOR OF VEHICLE WITH RESPECT TO SECONDARY
#            BODY... METERS B-29 ONLY IF MIDFLAG = DIMOFLAG = 1
#       B) CONIC INTEGRATION.  CSMCONIC, LEMCONIC ENTRANCES
#            L-X      STORE TIME IN PUSH LIST (TDEC1)
#            L        CALL
#            L+1              CSMCONIC (OR LEMCONIC)
#          INPUT/OUTPUT
#            SAME AS PRECISION INTEGRATION, EXCEPT RQVV NOT SET
#       C) INTEGRATE GIVEN STATE VECTOR.  INTEGRVS ENTRANCE
#            CALL
#                             INTSTALL
#            VLOAD
#                             POSITION VECTOR
#            STOVL            RCV
#                             VELOCITY VECTOR
#            STODL            VCV
#                             TIME STATE VECTOR VALID
#            STODL            TET
#                             FINAL RADIUS
#            STORE            RFINAL
#            SET(CLEAR)       SET(CLEAR)
#                             INTYPFLAG
#                             MOONFLAG
#            SET(CLEAR)       DLOAD
#                             DESIRED TIME
#            STCALL           TDEC1
#                             INTEGRVS
#         INPUT
#            RCV      POSITION VECTOR                  METERS
#            VCV      VELOCITY VECTOR                  M/CSEC
#            TET      TIME OF STATE VECTOR (MAY = 0)   CSEC B-28
```

```
#               TDEC1   TIME TO INTEGRATE TO            CSEC B-28 (PD 32D)
#                       (MAY BE INCREMENT IF TET=0)
#         OUTPUT
#               SAME AS FOR PRECISION OR CONIC INTEGRATION,
#               DEPENDING ON INTYPFLG.
#       D) INTEGRATE STATE VECTOR.  INTGRV ENTRANCE
#               L-X     STORE TIME IN PUSH LIST (TDEC1) (MAY BE DONE AFTER CALL TO INTSTALL)
#               L-8     CALL
#               L-7
#               L-6     SET(CLEAR)      SET(CLEAR)
#               L-5                     VINTFLAG        1=CSM, 0=LM
#               L-4                     INTYPFLAG       1=CONIC, 0=PRECISION
#               L-3     SET(CLEAR)      SET(CLEAR)
#               L-2                     DIM0FLAG        1=W-MATRIX, 0=NO W-MATRIX
#               L-1                     D6OR9FLG        1=9X9, 0=6X6
#               L       SET             DLOAD
#               L+1                     STATEFLG        DESIRE PERMANENT UPDATE
#               L+2                     FINAL RAD.      OF STATE VECTOR
#               L+3     STCALL          RFINAL
#               L+4                     INTEGRV
#               L       CALL                            NORMAL USE -- WILL UPDATE STATE
#               L+1                     INTEGRV         VECTOR IF DIM0FLAG=1. (STATEFLG IS
#               L+2     RETURN                          ALWAYS RESET IN INTEGRATION AFTER
#                                                       IT USED.)
#         INPUT
#               TDEC1 (PD 32D) TIME TO INTEGRATE TO     CSEC B-28
#         OUTPUT
#               SAME AS FOR PRECISION OR CONIC INTEGRATION
#         THE PROGRAM WILL SET MOONFLAG, MIDFLAG DEPENDING ON
#         THE PERMANENT STATE VECTOR REPRESENTATION.

                BANK    11
                SETLOC  INTINIT
                BANK
                EBANK=  RRECTCSM
                COUNT*  $$/INTIN
STATEINT        TC      PHASCHNG
                OCT     00052
                CAF     PRIO5
                TC      FINDVAC
                EBANK=  RRECTCSM
                2CADR   STATINT1

                TC      TASKOVER
STATINT1        TC      INTPRET
                BON     RTB
                        QUITFLAG        # KILL INTEGRATION UNTIL NEXT P00.
                        NOINT
                        LOADTIME
                STORE   TDEC1
```

```
                CALL
                        INTSTALL
                SET     CALL
                        NODOFLAG
                        SETIFLGS
                GOTO
                        STATEUP
600SECS         2DEC    60000

ENDINT          CLEAR   EXIT
                        STATEFLG
                TC      PHASCHNG
                OCT     20032
                EXTEND
                DCA     600SECS
                TC      LONGCALL
                EBANK=  RRECTHIS
                2CADR   STATEINT

                TC      ENDOFJOB
SETIFLGS        SET     CLEAR
                        STATEFLG
                        INTYPFLG
                CLEAR   CLEAR
                        DIMOFLAG
                        D6OR9FLG
                RVQ
NOINT           EXIT
                TC      PHASCHNG
                OCT     00002

                TC      DOWNFLAG
                ADRES   QUITFLAG
                TC      ENDOFJOB

# ATOPCSM TRANSFERS RRECT TO RRECT +41 TO RRECTCSM TO RRECTCSM +41
#
# CALLING SEQUENCE
#       L       CALL
#       L+1             ATOPCSM
#
# NORMAL EXIT AT L+2

ATOPCSM         STQ     RTB
                        S2
                        MOVEACSM
                SET     CALL
                        CMOONFLG
                        SVDWN1
                BON     CLRGO
```

```
                                MOONFLAG
                                S2
                                CMOONFLG
                                S2
MOVEACSM        TC              SETBANK
                TS              DIFEQCNT        # INITIALIZE INDEX
                INDEX           DIFEQCNT
                CA              RRECT
                INDEX           DIFEQCNT
                TS              RRECTCSM
                CCS             DIFEQCNT        # IS TRANSFER COMPLETE
                TCF             MOVEACSM +1     # NO-LOOP
                TC              DANZIG          # COMPLETE -- RETURN

# PTOACSM TRANSFERS RRECTCSM TO RRECTCSM +41 TO RRECT TO RRECT +41
#
# CALLING SEQUENCE
#       L       CALL
#                       PTOACSM
#
# NORMAL EXIT AT L+2

PTOACSM         RTB             BON
                                MOVEPCSM
                                CMOONFLG
                                SETMOON
CLRMOON         CLEAR           SSP
                                MOONFLAG
                                PBODY
                                0
                RVQ
SETMOON         SET             SSP
                                MOONFLAG
                                PBODY
                                2
                RVQ
MOVEPCSM        TC              SETBANK
                TS              DIFEQCNT
                INDEX           DIFEQCNT
                CA              RRECTCSM
                INDEX           DIFEQCNT
                TS              RRECT
                CCS             DIFEQCNT
                TCF             MOVEPCSM +1
                TC              DANZIG

# ATOPLEM TRANSFERS RRECT TO RRECT +41 TO RRECTLEM TO RRECTLEM +41
ATOPLEM         STQ             RTB
```

```
                          S2
                          MOVEALEM
                SET       CALL
                          LMOONFLG
                          SVDWN2
                BON       CLRGO
                          MOONFLAG
                          S2
                          LMOONFLG
                          S2
MOVEALEM        TC        SETBANK
                TS        DIFEQCNT
                INDEX     DIFEQCNT
                CA        RRECT
                INDEX     DIFEQCNT
                TS        RRECTLEM
                CCS       DIFEQCNT
                TCF       MOVEALEM +1
                TC        DANZIG

# PTOALEM TRANSFERS RRECTLEM TO RRECTLEM +41 TO RRECT TO RRECT +41

PTOALEM         BON       RTB
                          SURFFLAG
                          USEPIOS
                          MOVEPLEM
                BON       GOTO
                          LMOONFLG
                          SETMOON
                          CLRMOON
MOVEPLEM        TC        SETBANK
                TS        DIFEQCNT
                INDEX     DIFEQCNT
                CA        RRECTLEM
                INDEX     DIFEQCNT
                TS        RRECT
                CCS       DIFEQCNT
                TCF       MOVEPLEM +1
                TC        DANZIG

USEPIOS         SETPD     VLOAD
                          0
                          RLS
                PDDL      PUSH
                          TDEC1
                STODL     TET
                          5/8
                CALL
```

```
                                RP-TO-R
                        STOVL   RCV
                                ZUNIT
                        STODL   OD
                                TET
                        STODL   6D
                                5/8
                        SET     CALL            # NEEDED FOR SETTING X1 ON EXIT
                                MOONFLAG
                                RP-TO-R
                        VXV     VXSC
                                RCV
                                OMEGMOON
                        STOVL   VCV
                                ZEROVEC
                        STORE   TDELTAV
                        AXT,2   SXA,2
                                2
                                PBODY
                        STCALL  TNUV
                                A-PCHK
SETBANK                 CAF     INTBANK
                        TS      BBANK
                        CAF     FORTYONE
                        TC      Q
                        EBANK=  RRECTCSM
INTBANK                 BBCON   INTEGRV

# SPECIAL PURPOSE ENTRIES TO ORBITAL INTEGRATION.  THESE ROUTINES PROVIDE ENTRANCES TO INTEGRATION WITH
# APPROPRIATE SWITCHES SET OR CLEARED FOR THE DESIRED INTEGRATION.
#
# CSMPREC AND LEMPREC PERFORM ORBIT INTEGRATION BY THE ENCKE METHOD TO THE TIME INDICATED IN TDEC1.
# ACCELERATIONS DUE TO OBLATENESS ARE INCLUDED.  NO W-MATRIX INT. IS DONE.
# THE PERMANENT STATE VECTOR IS NOT UPDATED.
#
# CSMCONIC AND LEMCONIC PERFORM ORBIT INTEG. BY KEPLER'S METHOD TO THE TIME INDICATED IN TDEC1.
# NO DISTURBING ACCELERATIONS ARE INCLUDED.  IN THE PROGRAM FLOW THE GIVEN
# STATE VECTOR IS RECTIFIED BEFORE SOLUTION OF KEPLER'S EQUATION.
#
# THE ROUTINES ASSUME THAT THE CSM (LEM) STATE VECTOR IN P-MEM IS VALID.
# SWITCHES SET PRIOR TO ENTRY TO THE MAIN INTEG. PROG ARE AS FOLLOWS:
#                               CSMPREC         CSMCONIC        LEMPREC         LEMCONIC
#       VINTFLAG                SET             SET             CLEAR           CLEAR
#       INTYPFLG                CLEAR           SET             CLEAR           SET
#       DIMOFLAG                CLEAR           CLEAR           CLEAR           CLEAR
#
# CALLING SEQUENCE
#       L-X     STORE   TDEC1
#       L       CALL                    (STCALL TDEC1)
```

```
#       L+1             CSMPREC         (CSMCONIC, LEMPREC, LEMCONIC)
#
# NORMAL EXIT TO L+2
#
# SUBROUTINES CALLED
#       INTEGRV1
#       PRECOUT FOR CSMPREC AND LEMPREC
#       CONICOUT FOR CSMCONIC AND LEMCONIC
#
# OUTPUT -- SEE PAGE 2 OF THIS LOG SECTION
#
# INPUT
#       TDEC1           TIME TO INTEGRATE TO.  CSECS B-28

CSMPREC         STQ     CALL
                        X1
                        INTSTALL
                SXA,1   SET
                        IRETURN
                        VINTFLAG

IFLAGP          SET     CLEAR
                        PRECIFLG
                        DIMOFLAG
                CLRGO
                        INTYPFLG
                        INTEGRV1
LEMPREC         STQ     CALL
                        X1
                        INTSTALL
                SXA,1   CLRGO
                        IRETURN
                        VINTFLAG
                        IFLAGP

CSMCONIC        STQ     CALL
                        X1
                        INTSTALL
                SXA,1   SET
                        IRETURN
                        VINTFLAG
IFLAGC          CLEAR   SETGO
                        DIMOFLAG
                        INTYPFLG
                        INTEGRV1
LEMCONIC        STQ     CALL
                        X1
                        INTSTALL
                SXA,1   CLRGO
                        IRETURN
```

```
                                VINTFLAG
                                IFLAGC

INTEGRVS        SET             SSP
                                PRECIFLG
                                PBODY
                                0
                BOF             SSP
                                MOONFLAG
                                +3
                                PBODY
                                2
                STQ             VLOAD
                                IRETURN
                                ZEROVEC
                STORE           TDELTAV
                STCALL          TNUV
                                RECTIFY
                CLEAR           SET
                                DIMOFLAG
                                NEWIFLG
                SETGO
                                RPQFLAG
                                ALOADED

# INTEGRV IS AN ENTRY TO ORBIT INTEGRATION WHICH PERMITS THE CALLER,
# NORMALLY THE NAVIGATION PROGRAM, TO SET THE INTEG. FLAGS.  THE ROUTINE
# IS ENTERED AT INTEGRV1 BY CSMPREC ET AL. AND AT ALOADED BY INTEGRVS.
# THE ROUTINE SETS UP A-MEMORY IF ENTERED AT INTEGRV,1 AND SETS THE INTEG.
# PROGRAM FOR PRECISION OR CONIC.
#
# THE CALLER MUST FIRST CALL INTSTALL TO CHECK IF INTEG. IS IN USE BEFORE
# SETTING ANY FLAGS.
#
# THE FLAGS WHICH SHOULD BE SET OR CLEARED ARE
#       VINTFLAG        (IGNORED WHEN ENTERED FROM INTEGRVS)
#       INTYPFLG
#       DIMOFLAG
#       D6OR9FLG
#
# CALLING SEQUENCE
#       L-X     CALL
#       L-Y             INTSTALL
#       L-1     SET OR CLEAR ALL FOUR FLAGS.  ALSO CAN SET STATEFLG IF DESIRED
#               AND DIMOFLAG IS CLEAR.
#       L       CALL
#       L+1             INTEGRV
#
# INITIALIZATION
#       FLAGS AS ABOVE
#       STORE TIME TO INTEGRATE TO IN TDEC1
#
# OUTPUT
#       RATT    AS
#       VATT          DEFINED
```

```
#       TAT                     BEFORE

INTEGRV         STQ
                        IRETURN
INTEGRV1        SET     SET
                        RPQFLAG
                        NEWIFLG
INTEGRV2        SSP
                        QPRET
                        ALOADED
                BON     GOTO
                        VINTFLAG
                        PTOACSM
                        PTOALEM
ALOADED         DLOAD
                        TDEC1
                STORE   TDEC
                BOFF    GOTO
                        INTYPFLG
                        TESTLOOP
                        RVCON
A-PCHK          BOF     EXIT
                        STATEFLG
                        RECTOUT
                TC      PHASCHNG
                OCT     04022
                TC      UPFLAG          # PHASE CHANGE HAS OCCURRED BETWEEN
                ADRES   REINTFLG        # INTSTALL AND INTWAKE
                TC      INTPRET
                SSP
                        QPRET
                        PHEXIT
                BON     GOTO
                        VINTFLAG
                        ATOPCSM
                        ATOPLEM
PHEXIT          CALL
                        GRP2PC
RECTOUT         SETPD   CALL
                        0
                        RECTIFY
                VLOAD   VSL*
                        RRECT
                        0,2
                PDVL    VSL*            # RATT TO PD0
                        VRECT
                        0,2
                PDDL    PDVL            # VATT TO PD6   TAT TO PD12
                        TET
```

```
                                RRECT
                        PDVL    PDDL*
                                VRECT
                                MUEARTH,2
                        PUSH    AXT,1
                        DEC     -10
                        BON     AXT,1
                                MOONFLAG
                                +2
                        DEC     -2
INTEXIT                 SETPD   BOV
                                0
                                +1
                        CLEAR   CLEAR
                                AVEMIDSW        # ALLOW UPDATE OF DOWNLINK STATE VECTOR
                                PRECIFLG
                        CLEAR
                                STATEFLG
                        SLOAD   EXIT
                                IRETURN
                        CA      MPAC
                        INDEX   FIXLOC
                        TS      QPRET
                        TC      INTWAKE

# RVCON SETS UP ORBIT INTEGRATION TO DO A CONIC SOLUTION FOR POSITION AND
# VELOCITY FOR THE INTERVAL (TET-TDEC)

RVCON                   DLOAD   DSU
                                TDEC
                                TET
                        STCALL  TAU.
                                RECTIFY
                        CALL
                                KEPPREP
                        DLOAD   DAD
                                TC
                                TET
                        STCALL  TET
                                RECTOUT
```

```
TESTLOOP        BOF     CLRGO
                        QUITFLAG
                        +3
                        STATEFLG
                        INTEXIT         # STOP INTEGRATION
 +3             SETPD   LXA,2
                        10D
                        PBODY
                VLOAD   ABVAL
                        RCV
                PUSH    CLEAR           # RC TO 10D
                        MIDFLAG
                DSU*    BMN             # MIDFLAG=0 IF R G.T. RMP
                        RME,2
                        +3
                SET
                        MIDFLAG
NORFINAL        DLOAD   DMP
                        10D
                        34D
                SR1R    DDV*
                        MUEARTH,2
                SQRT    DMP
                        .3D
                SR3     SR4             # DT IS TRUNCATED TO A MULTIPLE
                DLOAD   SL
                        MPAC
                        15D             #       OF 128 CSECS.
                PUSH    BOV
                        MAXDT
                BDSU    BMN
                        DT/2MAX
                        MAXDT
DT/2COMP        DLOAD   DSU
                        TDEC
                        TET
                RTB     SL
                        SGNAGREE
                        8D
                STORE   DT/2            # B-19
                BOV     ABS
                        GETMAXDT
                DSU     BMN             # IS TIME TO INTEG. TO GR THAN MAXTIME
                        12D
                        POOHCHK
USEMAXDT        DLOAD   SIGN
                        12D
                        DT/2
```

```
                        STCALL  DT/2
                                POOHCHK
        MAXDT           DLOAD   PDDL            # EXCHANGE DT/2MAX WITH COMPUTED MAX.
                                DT/2MAX
                        GOTO
                                DT/2COMP
        GETMAXDT        RTB
                                SIGNMPAC
                        STCALL  DT/2
                                USEMAXDT
        POOHCHK         DLOAD   ABS
                                DT/2
                        DSU     BMN
                                DT/2MIN
                                A-PCHK
                        SLOAD   BHIZ
                                MODREG
                                +3
                        GOTO
                                TIMESTEP
                        BON                     # WAS THIS CALL VIA CSM(LEM)PREC
                                PRECIFLG
                                TIMESTEP        # YES
                        DLOAD   DSU
                                DT/2
                                12D
                        BMN     BOFCLR
                                A-PCHK
                                NEWIFLG
                                TIMESTEP
                        DLOAD   DSU
                                TDEC
                                TET
                        BMN                     # NO BACKWARD INTEGRATION
                                INTEXIT
                        PDDL    SR4
                                DT/2            # IS 4(DT) LS (TDEC - TET)
                        SR2R    BDSU
                        BMN     GOTO
                                INTEXIT
                                TIMESTEP
        DT/2MIN         2DEC    3 B-20

        DT/2MAX         2DEC    4000 E2 B-20

        INTSTALL        EXIT
                        CA      RASFLAG
                        MASK    INTBITAB        # IS THIS STALL AREA FREE
                        EXTEND
                        BZF     OKTOGRAB        # YES
```

```
                CAF     WAKESTAL
                TC      JOBSLEEP
INTWAKE0        EXIT
                TCF     INTWAKE1

INTWAKE         CS      RASFLAG         # IS THIS INTSTALLED ROUTINE TO BE
                MASK    REINTBIT        #       RESTARTED
                CCS     A
                TC      INTWAKE1        # NO

                INDEX   FIXLOC
                CA      QPRET
                TS      TBASE2          # YES, DON'T RESTART WITH SOMEONE ELSE'S Q

                TC      PHASCHNG
                OCT     04022

                CA      TBASE2
                INDEX   FIXLOC
                TS      QPRET

                CAF     REINTBIT
                MASK    RASFLAG
                EXTEND
                BZF     GOBAC           # DON'T INTWAKE IF WE CAME HERE VIA RESTART

INTWAKE1        CAF     WAKESTAL
                INHINT
                TC      JOBWAKE
                CCS     LOCCTR
                TCF     INTWAKE1
FORTYONE        DEC     41
                CS      INTBITAB
                MASK    RASFLAG
                TS      RASFLAG         # RELEASE STALL AREA
                RELINT
                TCF     GOBAC
OKTOGRAB        CAF     INTFLBIT
                INHINT
                ADS     RASFLAG
GOBAC           TC      INTPRET
                RVQ
WAKESTAL        CADR    INTSTALL +1
INTBITAB        OCT     20100
```

```
# AVETOMID
#
# THIS ROUTINE PERFORMS THE TRANSITION FROM A THRUSTING PHASE TO THE COAST
# PHASE BY INITIALIZING THIS VEHICLE'S PERMANENT STATE VECTOR WITH THE
# VALUES LEFT BY THE AVERAGEG ROUTINE IN RN,VN,PIPTIME.
#
# BEFORE THIS IS DONE THE W-MATRIX, IF ITS VALID (OR WFLAG OR RENDWFLT IS
# SET) IS INTEGRATED FORWARD TO PIPTIME WITH THE PRE-THRUST STATE VECTOR.
#
# IN ADDITION, THE OTHER VEHICLE IS INTEGRATED (PERMANENT) TO PIPTIME.
#
# FINALLY TRKMKCNT IS ZEROED.

                SETLOC  INTINIT
                BANK

                COUNT*  $$/INTIN
AVETOMID        STQ     BON
                        EGRESS
                        RENDWFLG
                        INT/W           # W-MATRIX VALID, GO INTEGRATE IT
                BON
                        ORBWFLAG
                        INT/W           # W-MATRIX VALID, GO INTEGRATE IT.

OTHERS          DLOAD   CALL            # GET SET FOR OTHER VEHICLE INTEGRATION
                        PIPTIME         # DESIRED TIME
                        INTSTALL
                SET     CALL
                        VINTFLAG        # CM
                        SETIFLGS        # SETS UP NONE W-MAT. PERMANENT INTEG.
                STCALL  TDEC1
                        INTEGRV

                AXT,2   CALL            # NOW MOVE PROPERLY SCALE RN,UN AS WELL AS
                        2               # PIPTIME TO INTEGRATION ERASABLES.
                        INTSTALL
                BON     AXT,2
                        MOONTHIS
                        +2
                        0
                VLOAD   VSR*
                        RN
                        0,2
                STORE   RRECT
                STODL   RCV
                        PIPTIME
                STOVL   TET
                        VN
```

```
                VSR*    CALL
                        0,2
                        MINIRECT        # FINISH SETTING UP STATE VECTOR
                RTB     SSP
                        MOVATHIS        # PUT TEMP STATE VECTOR INTO PERMANENT
                        TRKMKCNT
                        0
                GOTO
                        FAZAB5

INT/W           DLOAD   CALL
                        PIPTIME         # INTEGRATE W THRU BURN
                        INTSTALL
                SET     SET
                        DIMOFLAG        # DO W-MATRIX
                        AVEMIDSW        # SO WON'T CLOBBER RN,VN,PIPTIME
                SET     CLEAR
                        D6OR9FLG        # 9X9 FOR LM
                        VINTFLAG        # LM
                STCALL  TDEC1
                        INTEGRV
                GOTO
                        OTHERS          # NOW GO DO THE OTHER VEHICLE
```

```
# MIDTOAV1
#
# THIS ROUTINE INTEGRATES (PRECISION) TO THE TIME SPECIFIED IN TDEC1.
# IF, AT THE END OF AN INTEGRATION TIME STEP, CURRENT TIME PLUS A DELTA
# TIME (SEE TIMEDELT.....BASED ON THE COMPUTATION TIME FOR ONE TIME STEP)
# IS GREATER THAN THE DESIRED TIME, ALARM 1703 IS SET AND THE INTEGRATION
# IS DONE TO THE CURRENT TIME.
# RETURN IS IN BASIC TO THE RETURN ADDRESS PLUS ONE.
#
# IF THE INTEGRATION IS FINISHED TO THE DESIRED TIME, RETURN IS IN BASIC
# TO THE RETURN ADDRESS.
#
# IN EITHER CASE, BEFORE RETURNING, THE EXTRAPOLATED STATE VECTOR IS TRANSFERRED
# FROM R,VATT TO R,VN1-PIPTIME1 IS SET TO THE FINISHING INTEGRATION
# TIME AND MPAC IS SET TO THE DELTA TIME --
#                       TAT MINUS CURRENT TIME

# MIDTOAV2
#
# THIS ROUTINE INTEGRATES THIS VEHICLE'S STATE VECTOR TO THE CURRENT TIME.
# NO INPUTS ARE REQUIRED OF THE CALLER.  RETURN IS IN BASIC TO THE RETURN
# ADDRESS WITH THE ABOVE TRANSFERS TO R,VN1-PIPTIME1-AND MPAC DONE

                EBANK=  IRETURN1
MIDTOAV2        STQ     CLRGO           # INTEGRATE TO PRESENT TIME PLUS TIMEDELT
                        IRETURN1
                        MID1FLAG
                        ENTMID2

MIDTOAV1        STQ     SET             # INTEGRATE TO TDEC1
                        IRETURN1
                        MID1FLAG
                RTB     DAD             # INITIAL CHECK, IS TDEC1 IN THE FUTURE
                        LOADTIME
                        TIMEDELT
                BDSU    BPL
                        TDEC1
                        ENTMID1         # Y5S
                CALL
                        NOTIME          # NO, SET ALARM, SWITCH TO MIDTOAV2

ENTMID2         RTB     DAD
                        LOADTIME
                        TIMEDELT
                STORE   TDEC1

ENTMID1         CALL
                        INTSTALL
                CLEAR   CALL
```

```
                        DIMOFLAG        # NO W-MATRIX
                        THISVINT
                CLEAR   SET
                        INTYPFLG
                        MIDAVFLG        # LET INTEG. KNOW THE CALL IS FOR MIDTOAV.
                CALL
                        INTEGRV         # GO INTEGRATE
                CLEAR   VLOAD
                        MIDAVFLG
                        RATT
                STOVL   RN1
                        VATT
                STODL   VN1
                        TAT
                STORE   PIPTIME1
                SXA,2   SXA,1
                        RTX2
                        RTX1
                EXIT

                INHINT
                EXTEND
                DCS     TIME2
                DAS     MPAC
                TC      TPAGREE

                CA      IRETURN1
                TC      BANKJUMP
CKMID2          BOF     RTB
                        MID1FLAG
                        MID2
                        LOADTIME
                DAD     BDSU
                        TIMEDELT
                        TDEC
                BPL     CALL
                        TESTLOOP        # YES
                        NOTIME

TIMEINC         RTB     DAD
                        LOADTIME
                        TIMEDELT
                STCALL  TDEC
                        TESTLOOP

MID2            DLOAD   DSU
                        TDEC
                        TET
                ABS     DSU
                        3CSECS
```

```
                BMN     GOTO
                        A-PCHK
                        TIMEINC

NOTIME          CLEAR   EXIT            # TOO LATE
                        MID1FLAG
                INCR    IRETURN1        # SET ERROR EXIT (CALLOC +2)
                TC      ALARM           # INSUFFICIENT TIME FOR INTEGRATION --
                OCT     1703            #       TIG WILL BE SLIPPED...
                TC      INTPRET
                RVQ

3CSECS          2DEC    3

TIMEDELT        2DEC    2000

                BANK    27
                SETLOC  UPDATE2
                BANK
                EBANK=  INTWAKUQ

                COUNT*  $$/INTIN

INTWAKUQ        =       INTWAK1Q        # TEMPORARY UNTIL NAME OF INTWAK1Q IS CHNG

INTWAKEU        RELINT
                EXTEND
                QXCH    INTWAKUQ        # SAVE Q FOR RETURN

                TC      INTPRET

                SLOAD   BZE             # IS THIS A CSM/LEM STATE VECTOR UPDATE
                        UPSVFLAG        # REQUEST.  IF NOT GO TO INTWAKUP.
                        INTWAKUP

                VLOAD                   # MOVE PRECT(6) AND VRECT(6) INTO
                        RRECT           #       RCV(6) AND VCV(6) RESPECTIVELY.
                STOVL   RCV
                        VRECT           # NOW GO TO `RECTIFY +13D' TO
                CALL                    # STORE VRECT INTO VCV AND ZERO OUT
                        RECTIFY +13D    # TDELTAV(6),TNUV(6),TC(2), AND XKEP(2)
                SLOAD   ABS             # COMPARE ABSOLUTE VALUE OF `UPSVFLAG'
                        UPSVFLAG        # TO `UPDATE MOON STATE VECTOR CODE'
                DSU     BZE             # TO DETERMINE WHETHER THE STATE VECTOR TO
                        UPMNSVCD        # BE UPDATED IS IN THE EARTH OR LUNAR
                        INTWAKEM        # SPHERE OF INFLUENCE........
                AXT,2   CLRGO           # EARTH SPHERE OF INFLUENCE.
                DEC     0
                        MOONFLAG
```

```
                                INTWAKEC
INTWAKEM        AXT,2   SET             # LUNAR SPHERE OF INFLUENCE.
                DEC     2
                        MOONFLAG
INTWAKEC        SLOAD   BMN             # COMMON CODING AFTER X2 INITIALIZED AND
                                        # MOONFLAG SET (OR CLEARED).
                        UPSVFLAG        # IS THIS A REQUEST FOR A LEM OR CSM
                        INTWAKLM        #       STATE VECTOR UPDATE......
                CALL                    # UPDATE CSM STATE VECTOR
                        ATOPCSM

                CLEAR   GOTO
                        ORBWFLAG
                        INTWAKEX

INTWAKLM        CALL                    # UPDATE LM STATE VECTOR
                        ATOPLEM

INTWAKEX        CLEAR
                        RENDWFLG

INTWAKUP        SSP     CALL            # REMOVE `UPDATE STATE VECTOR INDICATOR'
                        UPSVFLAG
                        0
                        INTWAKE0        # RELEASE `GRAB' OF ORBIT INTEG.
                EXIT

                TC      PHASCHNG
                OCT     04026
                TC      INTWAKUQ

UPMNSVCD        OCT     2
                OCT     0

GRP2PC          STQ     EXIT
                        GRP2SVQ
                TC      PHASCHNG
                OCT     04022
                TC      INTPRET
                GOTO
                        GRP2SVQ
```

```
# ORBITAL INTEGRATION

# DELETE
                BANK    13
                SETLOC  ORBITAL
                BANK
                COUNT*  $$/ORBIT

# DELETE
KEPPREP         LXA,2   SETPD
                        PBODY
                        0
                DLOAD*  SQRT            # SQRT(MU) (+18 OR +15)         0D      PL 2D
                        MUEARTH,2
                PDVL    UNIT            #                                       PL 8D
                        RCV
                PDDL    NORM            # NORM R (+29 OR +27 - N1)      2D      PL 4D
                        36D
                        X1
                PDVL
                DOT     PDDL            # F*SQRT(MU) (+7 OR +5)         4D      PL 6D
                        VCV
                        TAU.            # (+28)
                DSU     NORM
                        TC
                        S1
                SR1
                DDV     PDDL
                        2D
                DMP     PUSH            # FS (+6 +N1-N2)                6D      PL 8D
                        4D
                DSQ     PDDL            # (FS)SQ (+12 +2(N1-N2))        8D      PL 10D
                        4D
                DSQ     PDDL*           # SSQ/MU (-2OR +2(N1-N2))       10D     PL 12D
                        MUEARTH,2
                SR3     SR4
                PDVL    VSQ             # PREALIGN MU (+43 OR +37)      12D     PL 14D
                        VCV
                DMP     BDSU            #                                       PL 12D
                        36D
                DDV     DMP             #                                       PL 10D
                        2D              # -(1/R-ALPHA) (+12 +3N1-2N2)
                DMP     SL*
                        DP2/3
                        0       -3,1    # 10L(1/R-ALPHA) (+13 +2(N1-N2))
                XSU,1   DAD             # 2(FS)SQ - ETCETERA                    PL 8D
                        S1              # X1 = N2-N1
                SL*     DSU             # -FS+2(FS)SQ ETC (+6 +N1-N2)           PL 6D
                        8D,1
                DMP     DMP
                        0D
                        4D
                SL*     SL*
```

```
                        8D,1
                        0,1             # S(-FS(1-2FS)-1/6...) (+17 OR +16)
                DAD     PDDL            #                                       PL 6D
                        XKEP
                DMP     SL*             # S(+17 OR +16)
                        OD
                        1,1
                BOVB    DAD
                        TCDANZIG
                STADR
                STORE   XKEPNEW
                STQ     AXC,1
                        KEPRTN
                DEC     10
                BON     AXC,1
                        MOONFLAG
                        KEPLERN
                DEC     2
                GOTO
                        KEPLERN
```

```
FBR3            LXA,1   SSP
                        DIFEQCNT
                        S1
                DEC     -13
                DLOAD   SR
                        DT/2
                        9D
                TIX,1   ROUND
                        +1
                PUSH    DAD
                        TC
                STODL   TAU.
                DAD
                        TET
                STCALL  TET
                        KEPPREP
```

```
# AGC ROUTINE TO COMPUTE ACCELERATION COMPONENTS.

ACCOMP          LXA,1   LXA,2
                        PBODY
                        PBODY
                VLOAD
                        ZEROVEC
                STOVL   FV
                        ALPHAV
                VSL*    VAD
                        0       -7,2
                        RCV
                STORE   BETAV
                BOF     XCHX,2
                        DIMOFLAG
                        +5
                        DIFEQCNT
                STORE   VECTAB,2
                XCHX,2
                        DIFEQCNT
                VLOAD   UNIT
                        ALPHAV
                STODL   ALPHAV
                        36D
                STORE   ALPHAM
                CALL
                        GAMCOMP
                VLOAD   SXA,1
                        BETAV
                        S2
                STODL   ALPHAV
                        BETAM
                STORE   ALPHAM
                BOF     DLOAD
                        MIDFLAG
                        OBLATE
                        TET
                CALL
                        LSPOS
                AXT,2   LXA,1
                        2
                        S2
                BOF
                        MOONFLAG
                        +3
                VCOMP   AXT,2
                        0
                STORE   BETAV
                STOVL   RPQV
```

```
                                2D
                STORE           RPSV
                SLOAD           DSU
                                MODREG
                                OCT27
                BHIZ            BOF
                                +3
                                DIMOFLAG
                                GETRPSV
                VLOAD           VXSC
                                ALPHAV
                                ALPHAM
                VSR*            VSU
                                1,2
                                BETAV
                XCHX,2
                                DIFEQCNT
                STORE           VECTAB  +6,2
                STORE           RQVV
                XCHX,2
                                DIFEQCNT
GETRPSV         VLOAD           INCR,1
                                RPQV
                                4
                CLEAR           BOF
                                RPQFLAG
                                MOONFLAG
                                +5
                VSR             VAD
                                9D
                                RPSV
                STORE           RPSV
                CALL
                                GAMCOMP
                AXT,2           INCR,1
                                4
                                4
                VLOAD
                                RPSV
                STCALL          BETAV
                                GAMCOMP
                GOTO
                                OBLATE
GAMCOMP         VLOAD           VSR1
                                BETAV
                VSQ             SETPD
                                0
                NORM            ROUND
                                31D
                PDDL            NORM            # NORMED B SQUARED TO PD LIST
```

```
                        ALPHAM          # NORMALIZE (LESS ONE) LENGTH OF ALPHA
                        32D             # SAVING NORM SCALE FACTOR IN X1
                SR1     PDVL
                        BETAV           # C(PDL+2) = ALMOST NORMED ALPHA
                UNIT
                STODL   BETAV
                        36D
                STORE   BETAM
                NORM    BDDV            # FORM NORMALIZE QUOTIEN ALPHAM/BETAM
                        33D
                SR1R    PUSH            # C(PDL+2) = ALMOST NORMALIZE RHO.
                DLOAD*
                        ASCALE,1
                STORE   S1
                XCHX,2  XAD,2
                        S1
                        32D
                XSU,2   DLOAD
                        33D
                        2D
                SR*     XCHX,2
                        0       -1,2
                        S1
                PUSH    SR1R            # RHO/4 TO 4D
                PDVL    DOT
                        ALPHAV
                        BETAV
                SL1R    BDSU            # (RHO/4) - 2(ALPHAV/2.BETAV/2)
                PUSH    DMPR            # TO PDL+6
                        4
                SL1
                PUSH    DAD
                        DQUARTER
                PUSH    SQRT
                DMPR    PUSH
                        10D
                SL1     DAD
                        DQUARTER
                PDDL    DAD             # (1/4)+2((Q+1)/4)      TO PD+14D
                        10D
                        HALFDP
                DMPR    SL1
                        8D
                DAD     DDV
                        THREE/8
                        14D
                DMPR    VXSC
                        6
                        BETAV           #                     _
                PDVL    VSR3            # (G/2)(C(PD+4))B/2 TO PD+16D
```

```
                        ALPHAV
                VAD     PUSH            # A12 + C(PD+16D) TO PD+16D
                DLOAD   DMP
                        0
                        12D
                NORM    ROUND
                        30D
                BDDV    DMP*
                        2
                        MUEARTH,2
                DCOMP   VXSC
                XCHX,2  XAD,2
                        S1
                        S2
                XSU,2   XSU,2
                        30D
                        31D
                BOV                     # CLEAR OVIND
                        +1
                VSR*    XCHX,2
                        0       -1,2
                        S1
                VAD
                        FV
                STORE   FV
                BOV     RVQ             # RETURN IF NO OVERFLOW
                        +1
GOBAQUE         VLOAD   ABVAL
                        TDELTAV
                BZE
                        INT-ABRT
                DLOAD   SR
                        H
                        9D
                PUSH    BDSU
                        TC
                STODL   TAU.
                        TET
                DSU     STADR
                STCALL  TET
                        KEPPREP
                CALL
                        RECTIFY
                SETGO
                        RPQFLAG
                        TESTLOOP

INT-ABRT        EXIT
                TC      POODOO
                OCT     00430
```

```
# THE OBLATE ROUTINE COMPUTES THE ACCELERATION DUE TO OBLATENESS.  IT USES THE UNIT OF THE VEHICLE
# POSITION VECTOR FOUND IN ALPHAV AND THE DISTANCE TO THE CENTER IN ALPHAM.  THIS IS ADDED TO THE SUM OF THE
# DISTURBING ACCELERATIONS IN FV AND THE PROPER DIFEQ STAGE IS CALLED VIA X1.

OBLATE          LXA,2   DLOAD
                        PBODY
                        ALPHAM
                SETPD   DSU*
                        0
                        RDE,2
                BPL     BOF             # GET URPV
                        NBRANCH
                        MOONFLAG
                        COSPHIE
                VLOAD   PDDL
                        ALPHAV
                        TET
                PDDL    CALL
                        3/5
                        R-TO-RP
                STORE   URPV
                VLOAD   VXV
                        504LM
                        ZUNIT
                VAD     VXM
                        ZUNIT
                        MMATRIX
                UNIT                    # POSSIBLY UNNECESSARY
COMTERM         STORE   UZ
                DLOAD   DMPR
                        COSPHI/2
                        3/32
                PDDL    DSQ             # P2/64 TO PD0
                        COSPHI/2
                DMPR    DSU
                        15/16
                        3/64
                PUSH    DMPR            # P3/32 TO PD2
                        COSPHI/2
                DMP     SL1R
                        7/12
                PDDL    DMPR
                        0
                        2/3
                BDSU    PUSH            # P4/128 TO PD4
                DMPR    DMPR
                        COSPHI/2        # BEGIN COMPUTING P5/1024
                        9/16
                PDDL    DMPR
                        2
                        5/128
```

```
                BDSU
                DMP*
                        J4REQ/J3,2
                DDV     DAD             #              -3
                        ALPHAM          # (((P5/256)B 2   /R+P4/32)   /R+P3/8)ALPHAV
                        4               #            4              3
                DMPR*   DDV
                        2J3RE/J2,2
                        ALPHAM
                DAD     VXSC
                        2
                        ALPHAV
                STODL   TVEC
                DMP*    SR1
                        J4REQ/J3,2
                DDV     DAD
                        ALPHAM
                DMPR*   SR3
                        2J3RE/J2,2
                DDV     DAD
                        ALPHAM
                VXSC    VSL1
                        UZ
                BVSU
                        TVEC
                STODL   TVEC
                        ALPHAM
                NORM    DSQ
                        X1
                DSQ     NORM
                        S1              #          4
                PUSH    BDDV*           # NORMED R   TO OD
                        J2REQSQ,2
                VXSC    BOV
                        TVEC
                        +1              # (RESET OVERFLOW INDICATOR)
                XAD,1   XAD,1
                        X1
                        X1
                XAD,1   VSL*
                        S1
                        0 -22D,1
                VAD     BOV
                        FV
                        GOBAQUE
                STCALL  FV
                        QUALITY1

QUALITY3        DSQ                     # J22 TERM X R**4 IN 2D, SCALED B61
                                        # AS VECTOR.
```

```
                PUSH    DMP             # STORE COSPHI**2 SCALED B2 IN 8D
                        5/8             # 5 SCALED B3
                PDDL    SR2             # PUT 5 COSPHI**2, D5, IN 8D.  GET
                                        # COSPHI**2 D2 FROM 8D
                DAD     BDSU            # END UP WITH (1-7 COSPHI**2), B5
                        8D              # ADDING COSPHI**2 B4 SAME AS COSPHI**2
                                        # X 2 D5
                        D1/32           # 1 SCALED B5
                DMP     DMP
                        URPV            # X COMPONENT
                        5/8             # 5 SCALED B3
                VXSC    VSL5            # AFTER SHIFT, SCALED B5
                        URPV            # VECTOR, B1.
                PDDL                    # VECTOR INTO 8D, 10D, 12D, SCALED B5.
                                        # GET 5 COSPHI**2 OUT OF 8D
                DSU     DAD
                        D1/32           # 1 B5
                        8D              # X COMPONENT (SAME AS MULTIPLYING
                                        # BY UNITX)
                STODL   8D
                        URPV            # X COMPONENT
                DMP     DMP
                        URPV    +4      # Z COMPONENT
                        5/8             # 5 B3 ANSWER B5
                SL1     DAD             # FROM 12D FOR Z COMPONENT (SL1 GIVES 10
                                        # INSTEAD OF 5 FOR COEFFICIENT)
                PDDL    NORM            # BACK INTO 12D FOR Z COMPNENT.
                        ALPHAM          # SCALED B27 FOR MOON
                        X2
                PUSH    SLOAD           # STORE IN 14D, DESTROYING URPV
                                        # X COMPONENT
                        E32C31RM
                DDV     VXSC            # IF X2 = 0, DIVISION GIVES B53, VXSC
                                        # OUT OF 8D B5 GIVES B58
                VSL*    VAD             # SHIFT MAKES B61, FOR ADDITION OF
                                        # VECTOR IN 2D
                        0       -3,2
                VSL*    V/SC            # OPERAND FROM 0D.  B108 FOR X1 = 0
                        0       -27D,1  # FOR X1 = 0, MAKES B88, GIVING B-20
                                        # FOR RESULT.
                PDDL    PDDL
                        TET
                        5/8             # ANY NON-ZERO CONSTANT
                LXA,2   CALL            # POSITION IN 0D, TIME IN 6D.  X2 LEFT
                                        # ALONE.
                        PBODY
                        RP-TO-R
                VAD     BOV             # OVERFLOW INDICATOR RESET IN "RP-TO-R"
                        FV
                        GOBAQUE
```

```
                STORE   FV
NBRANCH         SLOAD   LXA,1
                        DIFEQCNT
                        MPAC
                DMP     CGOTO
                        -1/12
                        MPAC
                        DIFEQTAB
COSPHIE         DLOAD
                        ALPHAV  +4
                STOVL   COSPHI/2
                        ZUNIT
                GOTO
                        COMTERM
DIFEQTAB        CADR    DIFEQ+0
                CADR    DIFEQ+1
                CADR    DIFEQ+2
TIMESTEP        BOF     VLOAD
                        MIDFLAG
                        RECTEST
                        RCV
                DOT     DMP
                        VCV
                        DT/2            # (R.V) X (DELTA T)
                BMN
                        RECTEST
                BON     BOF
                        MOONFLAG
                        LUNSPH
                        RPQFLAG
                        EARSPH
                DLOAD   CALL
                        TET
                        LSPOS           # RPQV IN MPAC
                STORE   RPQV            # RPQV
                LXA,2
                        PBODY
INLUNCHK        BVSU    ABVAL
                        RCV
                DSU     BMN
                        RSPHERE
                        DOSWITCH
RECTEST         VLOAD   ABVAL           # RECTIFY IF
                        TDELTAV
                BOV
                        CALLRECT
                DSU     BPL             #       1) EITHER TDELTAV OR TNUV EQUALS OR
                        3/4             #          EXCEEDS 3/4 IN MAGNITUDE
                        CALLRECT        #
                DAD     SL*             #                          OR
```

```
                                3/4             #
                                0       -7,2    #       2) ABVAL(TDELTAV) EQUALS OR EXCEEDS
                        DDV     DSU             #          .01(ABVAL(RCV))
                                10D
                                RECRATIO
                        BPL     VLOAD
                                CALLRECT
                                TNUV
                        ABVAL   DSU
                                3/4
                        BOV
                                CALLRECT
                        BMN
                                INTGRATE
        CALLRECT        CALL
                                RECTIFY
        INTGRATE        VLOAD
                                TNUV
                        STOVL   ZV
                                TDELTAV
                        STORE   YV
                        CLEAR
                                JSWITCH
        DIFEQ0          VLOAD   SSP
                                YV
                                DIFEQCNT
                                0
                        STODL   ALPHAV
                                DPZERO
                        STORE   H               # START H AT ZERO.  GOES 0(DELT/2)DELT.
                        BON     GOTO
                                JSWITCH
                                DOW..
                                ACCOMP
        EARSPH          VLOAD   GOTO
                                RPQV
                                INLUNCHK
        LUNSPH          DLOAD   SR2
                                10D
                        DSU     BMN
                                RSPHERE
                                RECTEST
                        BOF     DLOAD
                                RPQFLAG
                                DOSWITCH
                                TET
                        CALL
                                LUNPOS
                        VCOMP
                        STORE   RPQV
```

```
DOSWITCH        CALL
                        ORIGCHNG
                GOTO
                        INTGRATE
ORIGCHNG        STQ     CALL
                        ORIGEX
                        RECTIFY
                VLOAD   VSL*
                        RCV
                        0,2
                VSU     VSL*
                        RPQV
                        2,2
                STORE   RRECT
                STODL   RCV
                        TET
                CALL
                        LUNVEL
                BOF     VCOMP
                        MOONFLAG
                        +1
                PDVL    VSL*
                        VCV
                        0,2
                VSU
                VSL*
                        0       +2,2
                STORE   VRECT
                STORE   VCV
                LXA,2   SXA,2
                        ORIGEX
                        QPRET
                BON     GOTO
                        MOONFLAG
                        CLRMOON
                        SETMOON
```

```
# THE RECTIFY SUBROUTINE IS CALLED BY THE INTEGRATION PROGRAM AND OCCASIONALLY BY THE MEASUREMENT INCORPORATION
# ROUTINES TO ESTABLISH A NEW CONIC.

RECTIFY         LXA,2   VLOAD
                        PBODY
                        TDELTAV
                VSL*    VAD
                        0       -7,2
                        RCV
                STORE   RRECT
                STOVL   RCV
                        TNUV
                VSL*    VAD
                        0       -4,2
                        VCV
MINIRECT        STORE   VRECT
                STOVL   VCV
                        ZEROVEC
                STORE   TDELTAV
                STODL   TNUV
                        ZEROVEC
                STORE   TC
                STORE   XKEP
                RVQ
```

```
# THE THREE DIFEQ ROUTINES -- DIFEQ+0, DIFEQ+12, DIFEQ+24 -- ARE ENTERED TO PROCESS THE CONTRIBUTIONS AT THE
# BEGINNING, MIDDLE, AND END OF THE TIMESTEP, RESPECTIVELY.  THE UPDATING IS DONE BY THE NYSTROM METHOD.

DIFEQ+0         VLOAD   VSR3
                        FV
                STCALL  PHIV
                        DIFEQCOM
DIFEQ+1         VLOAD   VSR1
                        FV
                PUSH    VAD
                        PHIV
                STOVL   PSIV
                VSR1    VAD
                        PHIV
                STCALL  PHIV
                        DIFEQCOM
DIFEQ+2         DLOAD   DMPR
                        H
                        DP2/3
                PUSH    VXSC
                        PHIV
                VSL1    VAD
                        ZV
                VXSC    VAD
                        H
                        YV
                STOVL   YV
                        FV
                VSR3    VAD
                        PSIV
                VXSC    VSL1
                VAD
                        ZV
                STORE   ZV
                BOFF    CALL
                        JSWITCH
                        ENDSTATE
                        GRP2PC
                LXA,2   VLOAD
                        COLREG
                        ZV
                VSL3                            # ADJUST W-POSITION FOR STORAGE
                STORE   W       +54D,2
                VLOAD
                        YV
                VSL3    BOV
                        WMATEND
                STORE   W,2

                CALL
                        GRP2PC
```

```
                LXA,2   SSP
                        COLREG
                        S2
                        0
                INCR,2  SXA,2
                        6
                        YV
                TIX,2   CALL
                        RELOADSV
                        GRP2PC
                LXA,2   SXA,2
                        YV
                        COLREG

NEXTCOL         CALL
                        GRP2PC
                LXA,2   VLOAD*
                        COLREG
                        W,2
                VSR3                    # ADJUST W-POSITION FOR INTEGRATION
                STORE   YV
                VLOAD*  AXT,1
                        W       +54D,2
                        0
                VSR3                    # ADJUST W-VELOCITY FOR INTEGRATION
                STCALL  ZV
                        DIFEQ0

ENDSTATE        BOV     VLOAD
                        GOBAQUE
                        ZV
                STOVL   TNUV
                        YV
                STORE   TDELTAV
                BON     BOFF
                        MIDAVFLG
                        CKMID2          # CHECK FOR MID2 BEFORE GOING TO TIMEINC
                        DIMOFLAG
                        TESTLOOP
                EXIT
                TC      PHASCHNG
                OCT     04022           # PHASE 1
                TC      UPFLAG          # PHASE CHANGE HAS OCCURRED BETWEEN
                ADRES   REINTFLG        # INSTALL AND INTWAKE
                TC      INTPRET
                SSP
                        QPRET
                        AMOVED
                BON     GOTO
                        VINTFLAG
```

```
                        ATOPCSM
                        ATOPLEM
AMOVED          SET     SSP
                        JSWITCH
                        COLREG
                DEC     -30
                BOFF    SSP
                        D6OR9FLG
                        NEXTCOL
                        COLREG
                DEC     -48
                GOTO
                        NEXTCOL

RELOADSV        DLOAD                   # RELOAD TEMPORARY STATE VECTOR
                        TDEC            # FROM PERMANENT IN CASE OF
                STCALL  TDEC1
                        INTEGRV2        # BY STARTING AT INTEGRV2.
DIFEQCOM        DLOAD   DAD             # INCREMENT H AND DIFEQCNT.
                        DT/2
                        H
                INCR,1  SXA,1
                DEC     -12
                        DIFEQCNT        # DIFEQCNT SET FOR NEXT ENTRY.
                STORE   H
                VXSC    VSR1
                        FV
                VAD     VXSC
                        ZV
                        H
                VAD

                        YV
                STORE   ALPHAV
                BON     GOTO
                        JSWITCH
                        DOW..
                        FBR3

WMATEND         CLEAR   CLEAR
                        DIMOFLAG        # DON'T INTEGRATE W THIS TIME
                        ORBWFLAG        # INVALIDATE W
                CLEAR
                        RENDWFLG
                SET     EXIT
                        STATEFLG        # PICK UP STATE VECTOR UPDATE
                TC      ALARM
                OCT     421
                TC      INTPRET
```

```
                        GOTO
                                TESTLOOP        # FINISH INTEGRATING STATE VECTOR
```

```
# ORBITAL ROUTINE FOR EXTRAPOLATION OF THE W MATRIX.  IT COMPUTES THE SECOND DERIVATIVE OF EACH COLUMN POSITION
# VECTOR OF THE MATRIX AND CALLS THE NYSTROM INTEGRATION ROUTINES TO SOLVE THE DIFFERENTIAL EQUATIONS.  THE PROGRAM
# USES A TABLE OF VEHICLE POSITION VECTORS COMPUTED DURING THE INTEGRATION OF THE VEHICLE'S POSITION AND VELOCITY.

DOW..           LXA,2   DLOAD*
                        PBODY
                        MUEARTH,2
                STCALL  BETAM
                        DOW..1
                STORE   FV
                BOF     INCR,1
                        MIDFLAG
                        NBRANCH
                DEC     -6
                LXC,2   DLOAD*
                        PBODY
                        MUEARTH -2,2
                STCALL  BETAM
                        DOW..1
                BON     VSR6
                        MOONFLAG
                        +1
                VAD
                        FV
                STCALL  FV
                        NBRANCH
DOW..1          VLOAD   VSR4
                        ALPHAV
                PDVL*   UNIT
                        VECTAB,1
                PDVL    VPROJ
                        ALPHAV
                VXSC    VSU
                        3/4
                PDDL    NORM
                        36D
                        S2
                PUSH    DSQ
                DMP
                NORM    PDDL
                        34D
                        BETAM
                SR1     DDV
                VXSC
                LXA,2   XAD,2
                        S2
                        S2
                XAD,2   XAD,2
                        S2
                        34D
                VSL*    RVQ
```

```
                                O       -8D,2

# ********************************************************************************
# ********************************************************************************
SETITCTR        SSP     BOFF            # SET ITERCTR FOR LAMBERT CALLS.  THIS
                        ITERCTR         # CODING BELONGS IN INITVEL AND IS HERE
                        20D             # FOR PURPOSES OF A ONE-MODULE
                        AVEGFLAG        # REMANUFACTURE ONLY.  CODING SHOULD
                        LAMBERT         # BE MOVED BACK TO INITVEL FOR LUMINARY 1B
                SSP     GOTO
                        ITERCTR
                        5
                        LAMBERT
# ********************************************************************************
# ********************************************************************************

                SETLOC  ORBITAL1
                BANK

3/5             2DEC    .6 B-2

THREE/8         2DEC    .375

.3D             2DEC    .3 B-2

3/64            2DEC    3 B-6

DP1/4           2DEC    .25

DQUARTER        EQUALS  DP1/4
POS1/4          EQUALS  DP1/4
3/32            2DEC    3 B-5

15/16           2DEC    15. B-4

3/4             2DEC    3.0 B-2

7/12            2DEC    .5833333333

9/16            2DEC    9 B-4

5/128           2DEC    5 B-7

DPZERO          EQUALS  ZEROVEC
DP2/3           2DEC    .6666666667

2/3             EQUALS  DP2/3
OCT27           OCT     27
```

```
                BANK    13
                SETLOC  ORBITAL2
                BANK
# IT IS VITAL THAT THE FOLLOWING CONSTANTS NOT BE SHUFFLED
                DEC     -11
                DEC     -2
                DEC     -9
                DEC     -6
                DEC     -2
                DEC     -2
                DEC     0
                DEC     -12
                DEC     -9
                DEC     -4
ASCALE          DEC     -7
                DEC     -6
5/8             2DEC    5 B-3

-1/12           2DEC    -.1

RECRATIO        2DEC    .01

RSPHERE         2DEC    64373.76 E3 B-29

RDM             2DEC    16093.44 E3 B-27

RDE             2DEC    80467.20 E3 B-29

RATT            EQUALS  00
VATT            EQUALS  6D
TAT             EQUALS  12D
RATT1           EQUALS  14D
VATT1           EQUALS  20D
MU(P)           EQUALS  26D
TDEC1           EQUALS  32D
URPV            EQUALS  14D
COSPHI/2        EQUALS  URPV    +4
UZ              EQUALS  20D
TVEC            EQUALS  26D

QUALITY1        BOF     DLOAD
                        MOONFLAG
                        NBRANCH
                        URPV
                DSQ
QUALITY2        PDDL    DSQ             # SQUARE INTO 2D, B2
                        URPV    +2      # Y COMPONENT, B1
                DSU
                DMP     VXSC            # 5(Y**2-X**2)UR
                        5/8             # CONSTANT, 5B3
                        URPV            # VECTOR.  RESULT MAXIMUM IS 5, SCALING
```

```
                                        # HERE B6
          VSL3    PDDL                  # STORE SCALED B3 IN 2D, 4D, 6D FOR XYZ
                  URPV                  # X COMPONENT, B1
          SR1     DAD                   # 2 X X COMPONENT FOR B3 SCALING
                  2D                    # ADD TO VECTOR X COMPONENT OF ANSWER,
                                        # SAME AS MULTIPLYING BY UNITX.  MAX IS 7.
          STODL   2D
                  URPV    +2            # Y COMPONENT, B1
          SR1     BDSU                  # 2 X Y COMPONENT FOR B3 SCALING
                  4D                    # SUBTRACT FROM VECTOR Y COMPONENT OF
                                        # ANSWER, SAME AS MULTIPLYING BY UNITY.
                                        # MAX IS 7.
          STORE   4D                    # 2D HAS VECTOR, B3.
          SLOAD   VXSC                  # MULTIPLY COEFFICIENT TIMES VECTOR IN 2D
                  E3J22R2M
          PDDL    RVQ                   # J22 TERM X R**4 IN 2D, SCALED B61
                  COSPHI/2              # SAME AS URPV +4  Z COMPONENT
```

```
                BANK    22
                SETLOC  INFLIGHT
                BANK

                EBANK=  XSM

# CALCGTA COMPUTES THE GYRO TORQUE ANGLES REQUIRED TO BRING THE STABLE MEMBER INTO THE DESIRED ORIENTATION.
#
# THE INPUT IS THE DESIRED STABLE MEMBER COORDINATES REFERRED TO PRESENT STABLE MEMBER COORDINATES.  THE THREE
# HALF-UNIT VECTORS ARE STORED AT XDC, YDC, AND ZDC.
#
# THE OUTPUTS ARE THE THREE GYRO TORQUE ANGLES TO BE APPLIED TO THE Y, Z, AND X GYROS AND ARE STORED DP AT IGC,
# MGC, AND OGC RESPECTIVELY.

                COUNT*  $$/INFLT
CALCGTA         ITA     DLOAD           # PUSHDOWN 00-03, 16D-27D, 34D-37D
                        S2              # XDC = (XD1 XD2 XD3)
                        XDC             # YDC = (YD1 YD2 YD3)
                PDDL    PDDL            # ZDC = (ZD1 ZD2 ZD3)
                        HI6ZEROS
                        XDC     +4
                DCOMP   VDEF
                UNIT
                STODL   ZPRIME          # ZP = UNIT(-XD3 0 XD1) = (ZP1 ZP2 ZP3)
                        ZPRIME

                SR1
                STODL   SINTH           # SIN(IGC) = ZP1
                        ZPRIME  +4
                SR1
                STCALL  COSTH           # COS(IGC) = ZP3
                        ARCTRIG

                STODL   IGC             # Y GYRO TORQUING ANGLE   FRACTION OF REV.
                        XDC     +2
                SR1
                STODL   SINTH           # SIN(MGC) = XD2
                        ZPRIME

                DMP     PDDL
                        XDC     +4      # PD00 = (ZP1)(XD3)
                        ZPRIME  +4

                DMP     DSU
                        XDC             # MPAC = (ZP3)(XD1)
                STADR
                STCALL  COSTH           # COS(MGC) = MPAC - PD00
                        ARCTRIG
```

```
                STOVL   MGC             # Z GYRO TORQUING ANGLE   FRACTION OF REV.
                        ZPRIME
                DOT
                        ZDC
                STOVL   COSTH           # COS(OGC) = ZP . ZDC
                        ZPRIME
                DOT
                        YDC
                STCALL  SINTH           # SIN(OGC) = ZP . YDC
                        ARCTRIG

                STCALL  OGC             # X GYRO TORQUING ANGLE   FRACTION OF REV.
                        S2
```

```
# ARCTRIG COMPUTES AN ANGLE GIVEN THE SINE AND COSINE OF THIS ANGLE.
#
# THE INPUTS ARE SIN/4 AND COS/4 STORED DP AT SINTH AND COSTH.
#
# THE OUTPUT IS THE CALCULATED ANGLE BETWEEN +.5 AND -.5 REVOLUTIONS AND STORED AT THETA.  THE OUTPUT IS ALSO
# AVAILABLE AT MPAC.

ARCTRIG         DLOAD   ABS             # PUSHDOWN  16D-21D
                        SINTH
                DSU     BMN
                        QTSN45          # ABS(SIN/4) - SIN(45)/4
                        TRIG1           # IF (-45,45) OR (135,-135)

                DLOAD   SL1             # (45,135) OR (-135,-45)
                        COSTH
                ACOS    SIGN
                        SINTH
                STORE   THETA           # X = ARCCOS(COS) WITH SIGN(SIN)
                RVQ

TRIG1           DLOAD   SL1             # (-45,45) OR (135,-135)
                        SINTH
                ASIN
                STODL   THETA           # X = ARCSIN(SIN) WITH SIGN(SIN)
                        COSTH
                BMN
                        TRIG2           # IF (135,-135)

                DLOAD   RVQ
                        THETA           # X = ARCSIN(SIN)   (-45,45)

TRIG2           DLOAD   SIGN            # (135,-135)
                        HIDPHALF
                        SINTH
                DSU
                        THETA
                STORE   THETA           # X = .5 WITH SIGN(SIN) - ARCSIN(SIN)
                RVQ                     #       (+) - (+) OR (-) - (-)
```

```
# SMNB, NBSM, AND AXISROT, WHICH USED TO APPEAR HERE, HAVE BEEN
# COMBINED IN A ROUTINE CALLED AX*SR*T, WHICH APPEARS AMONG THE POWERED
# FLIGHT SUBROUTINES.
```

```
# CALCGA COMPUTES THE CDU DRIVING ANGLES REQUIRED TO BRING THE STABLE MEMBER INTO THE DESIRED ORIENTATION.
#
# THE INPUTS ARE  1) THE NAVIGATION BASE COORDINATES REFERRED TO ANY COORDINATE SYSTEM.  THE THREE HALF-UNIT
# VECTORS ARE STORED AT XNB, YNB, AND ZNB.  2) THE DESIRED STABLE MEMBER COORDINATES REFERRED TO THE SAME
# COORDINATE SYSTEM ARE STORED AT XSM, YSM, AND ZSM.
#
# THE OUTPUTS ARE THE THREE CDU DRIVING ANGLES AND ARE STORED SP AT THETAD, THETAD +1, AND THETAD +2.

CALCGA          SETPD                   # PUSHDOWN 00-05, 16D-21D, 34D-37D
                        0
                VLOAD   VXV
                        XNB             # XNB = OGA (OUTER GIMBAL AXIS)
                        YSM             # YSM = IGA (INNER GIMBAL AXIS)
                UNIT    PUSH            # PD0 = UNIT(OGA X IGA) = MGA

                DOT     ITA
                        ZNB
                        S2
                STOVL   COSTH           # COS(OG) = MGA . ZNB
                        0
                DOT
                        YNB
                STCALL  SINTH           # SIN(OG) = MGA . YNB
                        ARCTRIG
                STOVL   OGC
                        0

                VXV     DOT             # PROVISION FOR MG ANGLE OF 90 DEGREES
                        XNB
                        YSM
                SL1
                STOVL   COSTH           # COS(MG) = IGA . (MGA X OGA)
                        YSM
                DOT
                        XNB
                STCALL  SINTH           # SIN(MG) = IGA . OGA
                        ARCTRIG
                STORE   MGC

                ABS     DSU
                        .166...
                BPL
                        GIMLOCK1        # IF ANGLE GREATER THAN 60 DEGREES

CALCGA1         VLOAD   DOT
                        ZSM
                        0
                STOVL   COSTH           # COS(IG) = ZSM . MGA
                        XSM
```

```
                DOT     STADR
                STCALL  SINTH           # SIN(IG) = XSM . MGA
                        ARCTRIG

                STOVL   IGC
                        OGC
                RTB
                        V1STO2S
                STCALL  THETAD
                        S2

GIMLOCK1        EXIT
                TC      ALARM
                OCT     00401
                TC      UPFLAG          # GIMBAL LOCK HAS OCCURRED
                ADRES   GLOKFAIL

                TC      INTPRET
                GOTO
                        CALCGA1
```

```
# AXISGEN COMPUTES THE COORDINATES OF ONE COORDINATE SYSTEM REFERRED TO ANOTHER COORDINATE SYSTEM.
#
# THE INPUTS ARE  1) THE STAR1 VECTOR REFERRED TO COORDINATE SYSTEM A STORED AT STARAD.  2) THE STAR2 VECTOR
# REFERRED TO COORDINATE SYSTEM A STORED AT STARAD +6.  3) THE STAR1 VECTOR REFERRED TO COORDINATE SYSTEM B STORED
# AT LOCATION 6 OF THE VAC AREA.  4) THE STAR2 VECTOR REFERRED TO COORDINATE SYSTEM B STORED AT LOCATION 12D OF
# THE VAC AREA.
#
# THE OUTPUT DEFINES COORDINATE SYSTEM A REFERRED TO COORDINATE SYSTEM B.  THE THREE HALF-UNIT VECTORS ARE STORED
# AT LOCATIONS XDC, XDC +6, XDC +12D, AND STARAD, STARAD +6, STARAD +12D.

AXISGEN         AXT,1   SSP             # PUSHDOWN 00-30D, 34D-37D
                        STARAD  +6
                        S1
                        STARAD  -6

                SETPD
                        0
AXISGEN1        VLOAD*  VXV*            # 06D   UA = S1
                        STARAD  +12D,1  #       STARAD +00D     UB = S1
                        STARAD  +18D,1
                UNIT                    # 12D   VA = UNIT(S1 X S2)
                STORE   STARAD  +18D,1  #       STARAD +06D     VB = UNIT(S1 X S2)
                VLOAD*
                        STARAD  +12D,1

                VXV*    VSL1
                        STARAD  +18D,1  # 18D   WA = UA X VA
                STORE   STARAD  +24D,1  #       STARAD +12D     WB = UB X VB

                TIX,1
                        AXISGEN1

                AXC,1   SXA,1
                        6
                        30D

                AXT,1   SSP
                        18D
                        S1
                        6

                AXT,2   SSP
                        6
                        S2
                        2

AXISGEN2        XCHX,1  VLOAD*
                        30D             # X1=-6 X2=+6   X1=-6 X2=+4     X1=-6 X2=+2
                        0,1
```

```
                VXSC*   PDVL*           # J=(UA)(UB1)   J=(UA)(UB2)     J=(UA)(UB3)
                        STARAD  +6,2
                        6,1
                VXSC*
                        STARAD  +12D,2
                STOVL*  24D             # K=(VA)(VB1)   J=(VA)(VB2)     J=(VA)(VB3)
                        12D,1

                VXSC*   VAD
                        STARAD  +18D,2  # L=(WA)(WB1)   J=(WA)(WB2)     J=(WA)(WB3)
                VAD     VSL1
                        24D
                XCHX,1  UNIT
                        30D
                STORE   XDC     +18D,1  # XDC = L+J+K   YDC = L+J+K     ZDC = L+J+K

                TIX,1
                        AXISGEN3

AXISGEN3        TIX,2
                        AXISGEN2

                VLOAD
                        XDC
                STOVL   STARAD
                        YDC
                STOVL   STARAD  +6
                        ZDC
                STORE   STARAD  +12D

                RVQ
```

```
QTSN45          2DEC    .1768
.166...         2DEC    .1666666667
```

```
                BANK    14              # SAME FBANK AS THE FINDCDUD SUB-PROGRAM
                SETLOC  POWFLITE
                BANK

                EBANK=  DEXDEX
                COUNT*  $$/POWFL

# CDUTRIG, CDUTRIG1, CDUTRIG2, AND CD*TR*GS ALL COMPUTE THE SINES AND
# COSINES OF THREE 2'S COMPLEMENT ANGLES AND PLACE THE RESULT, DOUBLE
# PRECISION, IN THE SAME ORDER AS THE INPUTS, AT SINCDU AND COSCDU.  AN
# ADDITIONAL OUTPUT IS THE 1'S COMPLEMENT ANGLES AT CDUSPOT.  THESE
# ROUTINES GO OUT OF THEIR WAY TO LEAVE THE MPAC AREA AS THEY FIND IT.
# EXCEPT FOR THE GENERALLY UNIMPORTANT MPAC +2.  THEY DIFFER ONLY IN
# WHERE THEY GET THE ANGLES, AND IN METHOD OF CALLING.
#
# CDUTRIG (AND CDUTRIG1, WHICH CAN BE CALLED IN BASIC) COMPUTE THE
# SINES AND COSINES FROM THE CURRENT CONTENTS OF THE CDU REGISTERS.
# THE CONTENTS OF CDUTEMP, ETC., ARE NOT TOUCHED SO THAT THEY MAY
# CONTINUE TO FORM A CONSISTENT SET WITH THE LATEST PIPA READINGS.
#
# CDUTRIG1 IS LIKE CDUTRIG EXCEPT THAT IT CAN BE CALLED IN BASIC.
#
# CD*TR*GS FINDS CDU VALUES IN CDUSPOT RATHER THAN IN CDUTEMP.  THIS
# ALLOWS USERS TO MAKE TRANSFORMATIONS USING ARBITRARY ANGLES, OR REAL
# ANGLES IN AN ORDER OTHER THAN X Y Z.  A CALL TO THIS ROUTINE IS
# NECESSARY IN PREPARATION FOR A CALL TO AX*SR*T IN EITHER OF ITS TWO
# MODES (SMNB OR NBSM).  SINCE AX*SR*T EXPECTS TO FIND THE SINES AND
# COSINES IN THE ORDER Y Z X THE ANGLES MUST HAVE BEEN PLACED IN CDUSPOT
# IN THIS ORDER.  CD*TR*GS NEED NOT BE REPEATED WHEN AX*SR*T IS CALLED
# MORE THAN ONCE, PROVIDED THE ANGLES HAVE NOT CHANGED.  NOTE THAT SINCE
# IT CLOBBERS BUF2 (IN THE SINE AND COSINE ROUTINES) CD*TR*GS CANNOT BE
# CALLED USING BANKCALL.  SORRY.
#
# CD*TR*G IS LIKE CD*TR*GS EXCEPT THAT IT CAN BE CALLED IN
# INTERPRETIVE.

CDUTRIG         EXIT
                TC      CDUTRIGS
                TC      INTPRET
                RVQ

CD*TR*G         EXIT
                TC      CD*TR*GS
                TC      INTPRET
                RVQ

CDUTRIGS        CA      CDUX
                TS      CDUSPOT +4
                CA      CDUY
                TS      CDUSPOT
```

```
                CA      CDUZ
                TS      CDUSPOT +2

CD*TR*GS        EXTEND
                QXCH    TEM2
                CAF     FOUR
TR*GL**P        MASK    SIX             # MAKE IT EVEN AND SMALLER
                TS      TEM3
                INDEX   TEM3
                CA      CDUSPOT
                DXCH    MPAC            # STORING 2'S COMP ANGLE, LOADING MPAC
                DXCH    VBUF    +4      # STORING MPAC FOR LATER RESTORATION
                TC      USPRCADR
                CADR    CDULOGIC
                EXTEND
                DCA     MPAC
                INDEX   TEM3
                DXCH    CDUSPOT         # STORING 1'S COMPLEMENT ANGLE
                TC      USPRCADR
                CADR    COSINE
                DXCH    MPAC
                INDEX   TEM3
                DXCH    COSCDU          # STORING COSINE
                EXTEND
                INDEX   TEM3
                DCA     CDUSPOT         # LOADING 1'S COMPLEMENT ANGLE
                TC      USPRCADR
                CADR    SINE    +1      # SINE +1 EXPECTS ARGUMENT IN A AND L
                DXCH    VBUF    +4      # BRINGING UP PRIOR MPAC TO BE RESTORED
                DXCH    MPAC
                INDEX   TEM3
                DXCH    SINCDU
                CCS     TEM3
                TCF     TR*GL**P
                TC      TEM2
```

```
# ****************************************************************************************************
# QUICTRIG, INTENDED FOR QUIDANCE CYCLE USE WHERE TIME IS CRITICAL, IS A MUCH FASTER VERSION OF CD*TR*GS.
# QUICTRIG COMPUTES AND STORES THE SINES AND COSINES OF THE 2'S COMPLEMENT ANGLES AT CDUSPOT, CDUSPOT +2,
# AND CDUSPOT +4.  UNLIKE CD*TR*GS, QUICTRIG DOES NOT LEAVE THE 1'S COMPLEMENT VERSIONS OF THE ANGLES IN
# CDUSPOT.  QUICTRIG'S EXECUTION TIME IS 4.1 MS;  THIS IS 10 TIMES AS FAST AS CD*TR*GS.  QUICTRIG MAY BE
# CALLED FROM INTERPRETIVE AS AN RTB OP-CODE, OR FROM BASIC VIA BANKCALL OR IBNKCALL.

QUICTRIG        INHINT                  # INHINT SINCE DAP USES THE SAME TEMPS
                EXTEND
                QXCH    ITEMP1
                CAF     FOUR
 +4             MASK    SIX
                TS      ITEMP2
                INDEX   ITEMP2
                CA      CDUSPOT
                TC      SPSIN
                EXTEND
                MP      BIT14           # SCALE DOWN TO MATCH INTERPRETER OUTPUTS
                INDEX   ITEMP2
                DXCH    SINCDU
                INDEX   ITEMP2
                CA      CDUSPOT
                TC      SPCOS
                EXTEND
                MP      BIT14
                INDEX   ITEMP2
                DXCH    COSCDU
                CCS     ITEMP2
                TCF     QUICTRIG +4
                CA      ITEMP1
                RELINT
                TC      A
```

```
#***************************************************************************
# THESE INTERFACE ROUTINES MAKE IT POSSIBLE TO CALL AX*SR*T, ETC., IN
# INTERPRETIVE.  LATER, WHERE POSSIBLE, THEY WILL BE ELIMINATED.
#
# THESE INTERFACE ROUTINES ARE PERMANENT.  ALL RESTORE USER'S EBANK
# SETTING. ALL ARE STRICT INTERPRETIVE SUBROUTINES, CALLED USING "CALL",
# RETURNING VIA QPRET.  ALL EXPECT AND RETURN THE VECTOR TO BE TRANSFORMED
# INTERPRETER-STYLE IN MPAC; COMPONENTS AT MPAC, MPAC +3, AND MPAC +5.
#
# TRG*SMNB AND TRG*NBSM BOTH EXPECT TO SEE THE 2'S COMPLEMENT ANGLES
# AT CDUSPOT (ORDER Y Z X, AT CDUSPOT, CDUSPOT +2, AND CDUSPOT +4; ODD
# LOCATIONS NEED NOT BE ZEROED).  TRG*NBSM DOES THE NB TO SM TRANSFORMATION;
# TRG*SMNB, VICE VERSA.
#
# CDU*NBSM DOES ITS TRANSFORMATION USING THE PRESENT CONTENTS OF
# THE CDL COUNTERS.  OTHERWISE IT IS LIKE TRG*NBSM.
#
# CDU*SMNB IS THE COMPLEMENT OF CDU*NBSM.

CDU*SMNB        EXIT
                TC      CDUTRIGS
                TCF     C*MM*N1

TRG*SMNB        EXIT
                TC      CD*TR*GS
C*MM*N1         TC      MPACVBUF        # AX*SR*T EXPECTS VECTOR IN VBUF
                CS      THREE           # SIGNAL FOR SM TO NB TRANSFORMATION.
C*MM*N2         TC      AX*SR*T
                TC      INTPRET
                VLOAD   RVQ
                        VBUF

CDU*NBSM        EXIT
                TC      CDUTRIGS
                TCF     C*MM*N3

TRG*NBSM        EXIT
                TC      CD*TR*GS
C*MM*N3         TC      MPACVBUF        # FOR AX*SR*T
                CA      THREE           # SIGNAL FOR NB TO SM TRANSFORMATION
                TCF     C*MM*N2

# *NBSM* AND *SMNB* EXPECT TO SEE THE SINES AND COSINES (AT SINCDU
# AND COSCDU) RATHER THAN THE ANGLES THEMSELVES.  OTHERWISE THEY ARE
# LIKE TRG*NBSM AND TRG*SMNB.
#
# NOTE THAT JUST AS CD*TR*GS NEED BE CALLED ONLY ONCE FOR EACH SERIES
# OF TRANSFORMATIONS USING THE SAME ANGLES, SO TOO ONLY ONE OF TRG*NBSM
```

```
# AND TRG*SMNB NEED BE CALLED FOR EACH SERIES.  FOR SUBSEQUENT TRANFOR-
# MATIONS USE *NBSM* AND *SMNB*.

*SMNB*          EXIT
                TCF     C*MM*N1

*NBSM*          EXIT
                TCF     C*MM*N3

# AX*SR*T COMBINES THE OLD SMNB AND NBSM.  FOR THE NB TO SM
# TRANSFORMATION, ENTER WITH +3 IN A.  FOR SM TO NB, ENTER WITH -3.
# THE VECTOR TO BE TRANSFORMED ARRIVES, AND IS RETURNED, IN VBUF.
# AX*SR*T EXPECTS TO FIND THE SINES AND COSINES OF THE ANGLES OF ROTATION
# AT SINCDU AND COSCDU, IN THE ORDER Y Z X.  A CALL TO CD*TR*GS, WITH
# THE 2'S COMPLEMENT ANGLES (ORDER Y Z X) AT CDUSPOT, WILL TAKE CARE OF
# THIS.  HERE IS A SAMPLE CALLING SEQUENCE:--
#               TC      CDUTRIGS
#               CS      THREE           # ("CA THREE" FOR NBSM)
#               TC      AX*SR*T
# THE CALL TO CD*TR*GS NEED NOT BE REPEATED, WHEN AX*SR*T IS CALLED MORE
# THAN ONCE, UNLESS THE ANGLES HAVE CHANGED.
#
# AX*SR*T IS GUARANTEED SAFE ONLY FOR VECTORS OF MAGNITUDE LESS THAN
# UNITY.  A LOOK AT THE CASE IN WHICH A VECTOR OF GREATER MAGNITUDE
# HAPPENS TO LIE ALONG AN AXIS OF THE SYSTEM TO WHICH IT IS TO BE TRANS-
# FORMED CONVINCES ONE THAT THIS IS A RESTRICTION WHICH MUST BE ACCEPTED.

AX*SR*T         TS      DEXDEX          # WHERE IT BECOMES THE INDEX OF INDEXES.
                EXTEND
                QXCH    RTNSAVER

R*TL**P         CCS     DEXDEX          #               +3 --> 0        -3 --> 2
                CS      DEXDEX          # THUS:         +2 --> 1        -2 --> 1
                AD      THREE           #               +1 --> 2        -1 --> 0
                EXTEND
                INDEX   A
                DCA     INDEXI
                DXCH    DEXI

                CA      ONE
                TS      BUF
                EXTEND
                INDEX   DEX1
                DCS     VBUF
                TCF     LOOP1           # REALLY BE A SUBTRACT, AND VICE VERSA

LOOP2           DXCH    BUF             # LOADING VECTOR COMPONENT, STORING INDEX
```

```
LOOP1           DXCH    MPAC
                CA      SINSLOC
                AD      DEX1
                TS      ADDRWD

                TC      DMPSUB          # MULTIPLY AT SIN(CDUANGLE)
                CCS     DEXDEX
                DXCH    MPAC            # NBSM CASE
                TCF     +3
                EXTEND                  # SMNB CASE
                DCS     MPAC
                DXCH    TERM1TMP

                CA      SIX             # SINCDU AND COSCDU (EACH 6 WORDS) MUST
                ADS     ADDRWD          #        BE CONSECUTIVE AND IN THAT ORDER

                EXTEND
                INDEX   BUF
                INDEX   DEX1
                DCA     VBUF
                DXCH    MPAC
                TC      DMPSUB          # MULTIPLY BY COS(CDUANGLE)
                DXCH    MPAC
                DAS     TERM1TMP
                DXCH    TERM1TMP
                DDOUBL
                INDEX   BUF
                INDEX   DEX1
                DXCH    VBUF
                DXCH    BUF             # LOADING INDEX, STORING VECTOR COMPONENT

                CCS     A               # 'CAUSE THAT'S WHERE THE INDEX NOW IS
                TCF     LOOP2

                EXTEND
                DIM     DEXDEX          # DECREMENT MAGNITUDE PRESERVING SIGN

TSTPOINT        CCS     DEXDEX          # ONLY THE BRANCHING FUNCTION IS USED
                TCF     R*TL**P
                TC      RTNSAVER
                TCF     R*TL**P
                TC      RTNSAVER

SINSLOC         ADRES   SINCDU          # FOR USE IN SETTING ADDRWD

INDEXI          DEC     4               # **********   DON'T   ***********
                DEC     2               # **********   TOUCH   ***********
                DEC     0               # **********   THESE   ***********
```

```
                DEC     4               # ********** CONSTANTS ***********

# ********************************************************************************

                BANK    10
                SETLOC  FLESHLOC
                BANK
                COUNT*  $$/POWFL

# ROUTINE FLESHPOT COMPUTES THE BODY-STABLE MEMBER TRANSFORMATION MATRIX (COMMONLY CALLED XNB) AND STORES
# IT IN THE LOCATIONS SPECIFIED BY THE ECADR ENTERING IN A.

CALCSMSC        EXIT
                TC      BANKCALL
                CADR    FLESHPOT -1
                TC      INTPRET
                RVQ

XNBECADR        ECADR   XNB

 -1             CAF     XNBECADR

FLESHPOT        TS      TEM2
                XCH     EBANK
                XCH     TEM2
                MASK    LOW8
                AD      OCT1400
                TS      TEM1

                EXTEND
                DCA     COSCDUY
                DXCH    MPAC
                TC      DMP
                ADRES   COSCDUZ
                DXCH    MPAC
                DDOUBL
                INDEX   TEM1
                DXCH    0               # = COSY COSZ

                EXTEND
                DCA     SINCDUZ
                INDEX   TEM1
                DXCH    2               # = SINZ

                EXTEND
                DCS     SINCDUY
                DXCH    MPAC
                TC      DMPSUB          # ADDRWD SET TO COSCDUZ
```

```
                DXCH    MPAC
                DDOUBL
                INDEX   TEM1
                DXCH    4               # = - SINY COSZ

                EXTEND
                DCS     SINCDUX
                DXCH    MPAC
                TC      DMPSUB          # ADDRWD SET TO COSCDUZ STILL
                DXCH    MPAC
                DDOUBL
                DXCH    MPAC    +3

                EXTEND
                DCS     SINCDUX
                DXCH    MPAC
                TC      DMP
                ADRES   SINCDUZ
                EXTEND
                DCS     MPAC
                DXCH    MPAC    +5
                TC      DMP
                ADRES   SINCDUY
                DXCH    MPAC
                DDOUBL
                DDOUBL
                DXCH    MPAC    +5

                DXCH    MPAC
                TC      DMP
                ADRES   COSCDUY
                DXCH    MPAC
                DDOUBL
                DDOUBL
                DXCH    BUF

                EXTEND
                DCA     COSCDUY
                DXCH    MPAC
                TC      DMP
                ADRES   COSCDUX
                DXCH    MPAC
                DDOUBL
                DAS     MPAC    +5

                EXTEND
                DCA     SINCDUY
                DXCH    MPAC
                TC      DMPSUB          # ADDRWD SET TO COSCDUX
                DXCH    MPAC
```

```
                DDOUBL
                DAS     BUF

                DXCH    BUF
                DXCH    MPAC

                EXTEND
                DCA     MPAC
                INDEX   TEM1
                DXCH    14              # = - SINY COSX + SINX SINZ COSY

                EXTEND
                DCA     MPAC    +3
                INDEX   TEM1
                DXCH    16              # = - SINX COSZ

                EXTEND
                DCA     MPAC    +5
                INDEX   TEM1
                DXCH    20              # = COSX COSY - SINX SINY SINZ

                CA      TEM1
                TS      ADDRWD
                EXTEND
                DCA     Z
                AD      FOUR
                DXCH    LOC
                CAF     BIT8
                TS      EDOP
                TCF     VXV
                DXCH    MPAC
                DDOUBL
                INDEX   TEM1
                DXCH    6

                DXCH    MPAC    +3
                DDOUBL
                INDEX   TEM1
                DXCH    10

                DXCH    MPAC    +5
                DDOUBL
                INDEX   TEM1
                DXCH    12

                CA      TEM2
                TS      EBANK
                TCF     SWRETURN
```

```
# THE TFF SUBROUTINES MAY BE USED IN EITHER EARTH OR MOON CENTERED COORDINATES.  THE TFF ROUTINES NEVER
# KNOW WHICH ORIGIN APPLIES.  IT IS THE USER WHO KNOWS, AND WHO SUPPLIES RONE, VONE, AND 1/SQRT(MU) AT THE
# APPROPRIATE SCALE LEVEL FOR THE PROPER PRIMARY BODY.
#
#       EARTH ORIGIN    POSITION        -29     METERS
#                       VELOCITY        -7      METERS/CENTISECOND
#                       1/SQRT(MU)      +17     SQRT(CS SQ/METERS CUBED)
#
#       MOON ORIGIN     POSITION        -27     METERS
#                       VELOCITY        -5      METERS/CENTISECONDS
#                       1/SQRT(MU)      +14     SQRT(CS SQ/METERS CUBED)
#
# ALL DATA PROVIDED TO AND RECEIVED FROM ANY TFF SUBROUTINE WILL BE AT ONE OF THE LEVELS ABOVE.  IN ALL CASES,
# THE FREE FALL TIME IS RETURNED IN CENTISECONDS AT (-28).  PROGRAM TFF/CONIC WILL GENERATE VONE/RTMU AND
# LEAVE IT IN VONE' AT (+10) IF EARTH ORIGIN AND (+9) IF MOON ORIGIN.
#
# THE USER MUST STORE THE STATE VECTOR IN RONE, VONE, AND MU IN THE FORM 1/SQRT(MU) IN TFF/RTMU
# AT THE PROPER SCALE BEFORE CALLING TFF/CONIC.  SINCE RONE, VONE ARE IN THE EXTENDED VERB STORAGE AREA,
# THE USER MUST ALSO LOCK OUT THE EXTENDED VERBS, AND RELEASE THEM WHEN FINISHED.
#
# PROGRAMS CALC/TFF AND CALC/TPER ASSUME THAT THE TERMINAL RADIUS IS LESS THAN THE PRESENT
# RADIUS.  THIS RESTRICTION CAN BE REMOVED BY A 15 W CODING CHANGE, BUT AT PRESENT IT IS NOT DEEMED NECESSARY.
#
# THE FOLLOWING ERASABLE QUANTITIES ARE USED BY THE TFF ROUTINES, AND ARE LOCATED IN THE PUSH LIST.
#
#                                       BELOW   E:  IS USED FOR EARTH ORIGIN SCALE
#                                               M:  IS USED FOR MOON  ORIGIN SCALE
#
#TFFSW          =       119D    # BIT1  0 = CALCTFF             1 = CALCTPER
TFFDELQ         =       10D     #       Q2-Q1                   E: (-16)  M: (-15)
RMAG1           =       12D     #       ABVAL(RN)  M            E: (-29)  M: (-27)
#RPER           =       14D     #       PERIGEE RADIUS  M       E: (-29)  M: (-27)
TFFQ1           =       14D     #       R.V / SQRT(MUE)         E: (-16)  M: (-15)
#SDELF/2                        #       SIN(THETA) /2
CDELF/2         =       14D     #       COS(THETA) /2
#RAPO           =       16D     #       APOGEE RADIUS  M        E: (-29)  M: (-27)
NRTERM          =       16D     #       TERMINAL RADIUS  M      E: (-29+NR)
                                #                                         M: (-27+NR)
RTERM           =       18D     #       TERMINAL RADIUS  M      E: (-29)  M: (-27)
TFFVSQ          =       20D     #       -(V SQUARED/MU)  1/M    E: (20)   M: (18)
TFF1/ALF        =       22D     #       SEMI MAJ AXIS  M        E: (-22-2 NA)
                                #                                         M: (-20-2 NA)
TFFRTALF        =       24D     #       SQRT(ALFA)              E:(10+NA) M: (9+NA)
TFFALFA         =       26D     #       ALFA  1/M               E:(26-NR) M: (24-NR)
TFFNP           =       28D     #       SEMI LATUS RECTUM  M    E: (-38+2 NR)
                                #                                         M: (-36+2 NR)
TFF/RTMU        =       30D     #       1/SQRT(MU)              E: (17)   M: (14)
NRMAG           =       32D     #       PRESENT RADIUS  M       E: (-29+NR)
                                #                                         M: (-27+NR)
TFFX            =       34D     #
TFFTEM          =       36D     #       TEMPORARY
```

```
#                REGISTERS S1, S2 ARE UNTOUCED BY ANY TFF SUBROUTINE
#                INDEX REGISTERS X1, X2 ARE USED BY ALL TFF SUBROUTINES.  THEY ARE ESTAB-
#                LISHED IN TFF/CONIC AND MUST BE PRESERVED BETWEEN CALLS TO SUBSEQUENT
#                SUBROUTINES.
#                -NR                             C(X1) = NORM COUNT OF RMAG
#                -NA                             C(X2) = NORM COUNT OF SQRT(ABS(ALFA))
```

```
# SUBROUTINE NAME:  TFFCONIC                                                DATE:  01.29.67
# MOD NO:  0                                                                LOG SECTION:  TIME OF FREE FALL
# MOD BY:  RR BAIRNSFATHER
# MOD NO:  1    MOD BY:  RR BAIRNSFATHER        DATE: 11 APR 67
# MOD NO:  2    MOD BY:  RR BAIRNSFATHER        DATE: 21 NOV 67         ADD MOON MU.
# MOD NO:  3    MOD BY:  RR BAIRNSFATHER        DATE: 21 MAR 68         ACCEPT DIFFERENT EARTH/MOON SCALES
#
# FUNCTIONAL DESCRIPTION:  THIS SUBROUTINE IS CALLED TO COMPUTE THOSE CONIC PARAMETERS REQUIRED BY THE TFF
#       SUBROUTINES AND TO ESTABLISH THEM IN THE PUSH LIST AREA.  THE PARAMETERS ARE LISTED UNDER OUTPUT.
#       THE EQUATIONS ARE:
#               _   __ __
#               H = RN*VN                       ANGULAR MOMENTUM
#
#               LCP = H.H / MU                  SEMI LATUS RECTUM
#
#               ALFA = 2/RN - VN.VN / MU        RECIPROCAL SEMI-MAJOR AXIS, SIGNED
#
#       AND ALFA IS POS FOR ELLIPTIC ORBITS
#                    0 FOR PARABOLIC ORBITS
#                  NEG FOR HYPERBOLIC ORBITS
#       SUBROUTINE ALSO COMPUTES AND SAVES RMAG.
#
# CALLING SEQUENCE:
#       TFFCONIC EXPECTS CALLER TO ENTER WITH CORRECT GRAVITATIONAL CONSTANT IN MPAC, IN THE FORM
#       1/SQRT(MU).  THE PROGRAM WILL SAVE IN TFF/RTMU.  THE SCALE IS DETERMINED BY WHETHER EARTH OR MOON
#       ORIGIN IS USED.  THE CALLER MUST LOCK OUT THE EXTENDED VERBS BEFORE PROVIDING STATE VECTOR IN RONE,
#       VONE AT PROPER SCALE.  THE EXTENDED VERBS MUST BE RESTORED WHEN THE CALLER IS FINISHED USING THE
#       TFF ROUTINES.
#
#       ENTRY POINT TFFCONMU EXPECTS THAT TFF/RTMU IS ALREADY LOADED.
#
#       TO SPECIFY MU:  DLOAD   CALL                            IF MU ALREADY STORED:   CALL
#                               YOURMU  1/RTMU E:(17) M:(14)                                    TFFCONMU
#                               TFFCONIC
#       PUSHLOC = PDL+0, ARBITRARY IF LEQ 18D
#
# SUBROUTINES CALLED:  NONE
#
# NORMAL EXIT MODES:  RVQ
#
# ALARMS:  NONE
#
# OUTPUT:       THE FOLLOWING ARE STORED IN THE PUSH LIST AREA.
#               RMAG1           E:(-29) M:(-27) M  RN, PRESENT RADIUS LENGTH.
#               NRMAG           E:(-29+NR)      M  RMAG, NORMALIZED
#                               M:(-27+NR)
#               X1                              -NR, NORM COUNT
#               TFFNP           E:(-38+2NR)     M  LCP, SEMI LATUS RECTUM, WEIGHTED BY NR.      FOR VGAMCALC.
#                               M:(-36+2NR)
#               TFF/RTMU        E:(17) M:(14)   1/SQRT(MU)
#               TFFVSQ          E:(20) M:(18)   1/M  -(V SQ/MU):  PRESENT VELOCITY, NORMALIZED. FOR VGAMCALC
#               TFFALFA         E:(26-NR)       1/M  ALFA, WEIGHTED BY NR
#                               M:(24-NR)
#               TFFRTALF        E:(10+NA)       SQRT(ALFA), NORMALIZED
#                               M:(9+NA)
```

```
#               X2                              -NA, NORMCOUNT
#               TFF1/ALF        E:(-22-2NA)     SIGNED SEMI MAJ AXIS, WEIGHTED BY NA
#                               M:(-20-2NA)
#               PUSHLOC AT PDL+0
#
#       THE FOLLOWING IS STORED IN GENERAL ERASABLE
#               VONE'           E:(10) M:(9)    V/RT(MU), NORMALIZED VELOCITY
#
# ERASABLE INITIALIZATION REQUIRED:
#               RONE            E:(-29) M:(-27) M  STATE VECTOR         LEFT BY CALLER
#               VONE            E:(-7) M:(-5)   M/CS  STATE VECTOR      LEFT BY CALLER
#               TFF/RTMU        E:(17) M:(14)   1/RT(CS SQ/M CUBE)      IF ENTER VIA TFFCONMU.
#
# DEBRIS:       QPRET           PDL+0 ... PDL+3

                BANK    33
                SETLOC  TOF-FF
                BANK

                COUNT*  $$/TFF

TFFCONIC        STORE   TFF/RTMU        # 1/SQRT(MU)    E:(17) M:(14)

TFFCONMU        VLOAD   UNIT            # COME HERE WITH TFFRTMU LOADED.
                        RONE            # SAVED RN.   M  E:(-29) M:(-27)
                PDDL                    # UR/2 TO PDL+0, +5
                        36D             # MAGNITUDE
                STORE   RMAG1           # M  E:(-29) M:(-27)

                NORM
                        X1              # -NR
                STOVL   NRMAG           # RMAG  M  E:(-29+NR) M:(-27+NR)
                        VONE            # SAVED VN.  M/CS  E:(-7) M:(-5)
                VXSC
                        TFF/RTMU        # E:(17) M:(14)
                STORE   VONE'           # VN/SQRT(MU)  E:(10) M:(9)

                VXSC    VXV
                        NRMAG           # E:(-29+NR) M:(-27+NR)
                                        # UR/2 FROM PDL
                VSL1    VSQ             # BEFORE:  E:(-19+NR) M:(-18+NR)
                STODL   TFFNP           # LC P  M  E:(-38+2NR) M:(-36+2NR)
                                        # SAVE ALSO FOR VGAMCALC
                        TFF1/4
                DDV     PDVL            # (2/RMAG)  1/M  E:(26-NR) M:(24-NR)
                        NRMAG           # RMAG  M  E:(-29+NR) M:(-27+NR)
                        VONE'           # SAVED VN.  E:(10) M:(9)
                VSQ     DCOMP           # KEEP MPAC+2 HONEST FOR SQRT.
                STORE   TFFVSQ          # -(V SQ/MU)  E:(20) M:(18)
                                        # SAVE FOR VGAMCALC
                SR*     DAD
```

```
                        0        -6,1    # GET -VSQ/MU  E:(26-NR) M:(24-NR)
                STADR
                                         # 2/RMAG  FROM PDL+2
                STORE   TFFALFA          # ALFA  1/M  E:(26-NR) M:(24-NR)
                SL*     PUSH             # TEMP SAVE ALFA  E:(20) M:(18)
                        0        -6,1
                ABS     SQRT             # E:(10) M:(9)
                NORM
                        X2               # X2 = -NA
                STORE   TFFRTALF         # SQRT( ABS(ALFA) )  E:(10+NA) M:(9+NA)
                DSQ     SIGN             # NOT SO ACCURATE, BUT OK
                                         # ALFA FROM PDL+2  E:(20) M:(18)
                BZE     BDDV             # SET 1/ALFA =0, TO SHOW SMALL ALFA
                        +2
                        TFF1/4
 +2             STORE   TFF1/ALF         # 1/ALFA  E:(-22-2NA) M:(-20-2NA)
DUMPCNIC        RVQ

#                                                              39 W
```

```
# SUBROUTINE NAME:  TFFRP/RA                                             DATE: 01.17.67
# MOD NO:  0                                                             LOG SECTION:  TIME OF FREE FALL
# MOD NO:  1     MOD BY:  RR BAIRNSFATHER         DATE: 11 APR 67
# MOD NO:  2     MOD BY:  RR BAIRNSFATHER         DATE: 21 MAR 68        ACCEPT DIFFERENT EARTH/MOON SCALES
#                                                                        ALSO IMPROVE ACCURACY OF RAPO.
#
# FUNCTIONAL DESCRIPTION:  USED BY CALCTPER AND TFF DISPLAYS TO CALCULATE PERIGEE RADIUS AND ALSO
#        APOGEE RADIUS FOR A GENERAL CONIC.
#        PROGRAM GIVES PERIGEE RADIUS AS          APOGEE RADIUS IS GIVEN BY
#                RP = P/(1+E)                             RA = (1+E) / ALFA
#        WHERE    2
#                E  = 1 - P ALFA
#        IF RA IS NEGATIVE OR SHOWS DIVIDE OVERFLOW, THEN RA = POSMAX BECAUSE
#                1. APOGEE RADIUS IS NOT MEANINGFUL FOR HYPERBOLA
#                2. APOGEE RADIUS IS NOT DEFINED FOR PARABOLA
#                3. APOGEE RADIUS EXCEEDS THE SCALING FOR ELLIPSE.
#
#        THIS SUBROUTINE REQUIRED THE SIGNED RECIPROCAL SEMI MAJ AXIS, ALFA, AND SEMI-LATUS RECTUM AS DATA.
#
# CALLING SEQUENCE:     CALL
#                               TFFRP/RA
#        PUSHLOC = PDL+0, ARBITRARY IF LEQ 10D
#        C(MPAC) UNSPECIFIED
#
# SUBROUTINES CALLED:   NONE
#
# NORMAL EXIT MODE:     RVQ
#        IF ELLIPSE, WITHIN NORMAL SCALING, RAPO IS CORRECT.
#        OTHERWISE, RAPO = POSMAX.
#
# ALARMS:        NONE
#
# OUTPUT:        STORED IN PUSH LIST AREA.  SCALE OF OUTPUT AGREES WITH DATA SUPPLIED TO TFF/CONIC.
#        RPER    E:(-29) M:(-27)         M       PERIGEE RADIUS          DESTROYED BY CALCTFF/CALCTPER, TFFTRIG.
#        RAPO    E:(-29) M:(-27)         M       APOGEE RADIUS           WILL BE DESTROYED BY CALCTFF/CALCTPER
#        PUSHLOC AT PDL+0
#
# ERASABLE INITIALIZATION REQUIRED:
#        TFFALFA E:(26-NR)               M       1/SEMI MAJ AXIS         LEFT BY TFFCONIC
#                M:(24-NR)
#        TFFNP   E:(-38+2NR)             M       LC P, SEMI LATUS RECTUM LEFT BY TFFCONIC
#                M:(-36+2NR)
#        X1                              -NR, NORM COUNT OF RMAG         LEFT BY TFFCONIC
#        X2                              -NA, NORM COUNT OF ALFA         LEFT BY TFFCONIC
#
# DEBRIS:        QPRET, PDL+0 ... PDL+1
```

```
RAPO            =       16D             # APOGEE RADIUS  M  E:(-29) M:(-27)
RPER            =       14D             # PERIGEE RADIUS  M  E:(-29) M:(-27)

TFFRP/RA        DLOAD   DMP
                        TFFALFA         # ALFA  1/M  E:(26-NR) M:(24-NR)
                        TFFNP           # LC P  M E:(-38+2NR) M:(-36+2NR)
                SR*     DCOMP           # ALFA P (-12+NR)
                        0        -8D,1  # ALFA P (-4)
                DAD     ABS             # (DCOMP GIVES VALID TP RESULT FOR SQRT)
                                        # (ABS PROTECTS SQRT IF E IS VERY NEAR 0)
                        DP2(-4)
                SQRT    DAD             # E SQ = (1- P ALFA) (-4)
                        TFF1/4
                PUSH    BDDV            # (1+E)  (-2)  TO PDL+0
                        TFFNP           # LCP  M  E:(-38+2NR) M:(-36+2NR)
                SR*     SR*             # (DOES SR THEN SL TO AVOID OVFL)
                        0,1             # X1=-NR
                        0       -7,1    # (EFFECTIVE SL)
                STODL   RPER            # PERIGEE RADIUS  M  E:(-29) M:(-27)
                                        # (1+E)  (-2)  FROM PDL+0
                DMP     BOVB
                        TFF1/ALF        # E:(-22-2NA) M:(-20-2NA)
                        TCDANZIG        # CLEAR OVFIND, IF ON.
                BZE     SL*
                        MAXRA           # SET POSMAX IF ALFA=0
                        0       -5,2    # -5+NA
                SL*     BOV
                        0,2
                        MAXRA           # SET POSMAX IF OVFL.
                BPL                     # CONTINUE WITH VALID RAPO.
                        +3
MAXRA           DLOAD                   # RAPO CALC IS NOT VALID.  SET RAPO =
                        NEARONE         # POSMAX AS A TAG.
 +3             STORE   RAPO            # APOGEE RADIUS  M  E:(-29) M:(-27)
DUMPRPRA        RVQ

#                                                               30 W
```

```
# SUBROUTINE NAME:  CALCTPER / CALCTFF                                               DATE:  01.29.67
# MOD NO:  0                                                                         LOG SECTION:  TIME OF FREE FALL
# MOD BY:  RR BAIRNSFATHER
# MOD NO:  1    MOD BY:  RR BAIRNSFATHER        DATE: 21 MAR 67
# MOD NO:  2    MOD BY:  RR BAIRNSFATHER        DATE: 14 APR 67
# MOD BY:  3    MOD BY:  RR BAIRNSFATHER        DATE: 8 JUL 67           NEAR EARTH MUE AND NEG TFF (GONEPAST)
# MOD BY:  4    MOD BY:  RR BAIRNSFATHER        DATE: 21 NOV 67          ADD VARIABLE MU.
# MOD BY:  5    MOD BY:  RR BAIRNSFATHER        DATE: 21 MAR 68          ACCEPT DIFFERENT EARTH/MOON SCALES
#
# FUNCTIONAL DESCRIPTION:  PROGRAM CALCULATES THE FREE-FALL TIME OF FLIGHT FROM PRESENT POSITION RN AND
#       VELOCITY VN TO A RADIUS LENGTH SPECIFIED BY RTERM, SUPPLIED BY THE USER.  THE POSITION VECTOR
#       RN MAY BE ON EITHER SIDE OF THE CONIC, BUT RTERM IS CONSIDERED ON THE INBOUND SIDE.
#       THE EQUATIONS ARE:
#
#               Q2 = -SQRT(RTERM (2-RTERM ALFA) - LCP)  (INBOUND SIDE)  LEQ +- LCE/SQRT(ALFA)
#                         __ __
#               Q1 = RN.VN / SQRT(MU)                                    LEQ +- LCE/SQRT(ALFA)
#
#               Z = NUM / DEN                                            LEQ +- 1/SQRT(ALFA)
#
#       WHERE, IF INBOUND
#               NUM = RTERM -RN                                          LEQ +- 2 LCE/ALFA
#               DEN = Q2+Q1                                              LEQ +- 2 LCE/SQRT(ALFA)
#
#       AND, IF OUTBOUND
#               NUM = Q2-Q1                                              LEQ +- 2 LCE/SQRT(ALFA)
#               DEN = 2 - ALFA (RTERM + RN).                             LEQ +- 2 LCE
#
#       IF      ALFA ZZ < 1.0           (FOR ALL CONICS EXCEPT ELLIPSES HAVING ABS(DEL ECC ANOM) G 90 DEG)
#       THEN    X = ALFA Z Z
#       AND     TFF = (RTERM +RN -2 ZZ T(X) ) Z/SQRT(MU)
#               EXCEPT  IF ALFA PNZ, AND IF TFF NEG,
#               THEN    TFF = 2 PI /(ALFA SQRT(ALFA)) + TFF
#       OR      IF ALFA ZZ GEQ 1.0      (FOR ELLIPSES HAVING ABS(DEL ECC ANOM) GEQ 90 DEG)
#       THEN    X = 1/ALFA Z Z
#       AND     TFF = (PI/SQRT(ALFA) -Q2 +Q1 +2(X T(X) -1) /ALFA Z) /ALFA SQRT(MU)
#       WHERE   T(X) IS A POLYNOMIAL APPROXIMATION TO THE SERIES
#                             2      3             2
#               1/3 - X/5 + X /7 - X /8 ...     (X  < 1.0)
#
# CALLING SEQUENC:      TIME TO RTERM                   TIME TO PERIGEE
#                       CALL                            CALL
#                               CALCTFF                         CALCTPER
#                       C(MPAC) = TERMNL RAD M          C(MPAC) = PERIGEE RAD M
#       FOR EITHER, E:(-29) M:(-27)
#       FOR EITHER, PUSHLOC = PDL+0, ARBITRARY IF LEQ 8D.
```

```
#
# SUBROUTINES CALLED:   T(X), VIA RTB
#
# NORMAL EXIT MODE:     RVQ
#       HOWEVER, PROGRAM EXITS WITH ONE OF THE FOLLOWING VALUES FOR TFF (-28) CS IN MPAC.  USER MUST STORE.
#               A. TFF = FLIGHT TIME.  NORMAL CASE FOR POSITIVE FLIGHT TIME LESS THAN ONE ORBITAL PERIOD.
#               B. (THIS OPTION IS NO LONGER USED.)
#               C. TFF = POSMAX.  THIS INDICATES THAT THE CONIC FROM THE PRESENT POSITION WILL NOT RETURN TO
#                  THE SPECIFIED ALTITUDE.  ALSO INDICATES OUTBOUND PARABOLA OR HYPERBOLA.
#
# OUTPUT:       C(MPAC)         (-28) CS        TIME OF FLIGHT, OR TIME TO PERIGEE
#               TFFX            (0)             X                                       LEFT FOR ENTRY DISPLAY TFF ROUTINES
#               NRTERM          E:(-29+NR) M    RTERM, WEIGHTED BY NR                   LEFT FOR ENTRY DISPLAY TFF ROUTINES
#                               M:(-27+NR)
#               TFFTEM          E:(-59+2NR)     LCP Z Z SGN(SDELF)                      LEFT FOR ENTRY DISPLAY TFF ROUTINES
#                               M:(-55+2NR)     LCP /ALFA SGN(SDELF)                    LEFT FOR ENTRY DISPLAY TFF ROUTINES
#               NOTE:   TFFTEM = PDL 36D AND WILL BE DESTROYED BY .:UNIT:.
#               RMAG1           E:(-29) M:(-27) PDL 12 NOT TOUCHED.
#               TFFQ1           E:(-16) M:(-15) PDL 14D
#               TFFDELQ         E:(-16) M:(-15) PDL 10D
#               PUSHLOC AT PDL+0
#
# ERASABLE INITIALIZATION REQUIRED:
#               RONE            E:(-29) M:(-27) M  STATE VECTOR                         LEFT BY USER
#               VONE'           E:(+10) M:(+9)  VN/SQRT(NU)                             LEFT BY TFF/CONIC
#               RMAG1           E:(-29) M:(-27) PRESENT RADIUS, M                       LEFT BY TFFCONIC
#               C(MPAC)         E:(-29) M:(-27) RTERM, TERMINAL RADIUS LENGTH, M        LEFT BY USER
#
#               THE FOLLOWING ARE STORED IN THE PUSH LIST AREA.
#               TFF/RTMU        E:(17) M:(14)   1/SQRT(MU)                              LEFT BY TFFCONIC.
#               NRMAG           E:(-29+NR)      M  RMAG, NORMALIZED                     LEFT BY TFFCONIC
#                               M:(-27+NR)
#               X1                              -NR, NORM COUNT                         LEFT BY TFFCONIC
#               TFFNP           E:(-38+2NR)     M  LCP, SEMI LATUS RECTUM, WEIGHT NR    LEFT BY TFFCONIC
#                               M:(-36+2N4)
#               TFFALFA         E:(26-NR)       1/M  ALFA, WEIGHT NR                    LEFT BY TFFCONIC
#                               M:(24-NR)
#               TFFRTALF        E:(10+NA)       SQRT(ALFA), NORMALIZED                  LEFT BY TFFCONIC
#                               M:(9+NA)
#               X2                              -NA, NORMCOUNT                          LEFT BY TFFCONIC
#               TFF1/ALF        E:(-22-2NA)     SIGNED SEMI-MAJOR AXIS, WEIGHTED BY NA  LEFT BY TFFCONIC
#                               M:(-20-2NA)
#
# DEBRIS:       QPRET, PDL+0 ... PDL+3
#               RTERM           E:(-29) M(-27)  RTERM, TERMINAL RADIUS LENGTH
#               RAPO            E:(-29) M(-27)  PDL 16D (=NRTERM)
#               RPER            E:(-29) M(-27)  PDL 14D (=TFFQ1)
```

```
CALCTPER        SETGO                           # ENTER WITH RPER IN MPAC
                        TFFSW
                        +3
CALCTFF         CLEAR                           # ENTER WITH RTERM IN MPAC
                        TFFSW
 +3             STORE   RTERM                   # E:(-29) M:(-27)
                SL*
                        0,1                     # X1=-NR
                STORE   NRTERM                  # RTERM  E:(-29+NR) M:(-27+NR)
                DMP     BDSU
                        TFFALFA                 # ALFA  E:(26-NR) M:(24-NR)
                        TFF1/4
                PUSH    DMP                     # (2-ALFA RTERM)  (-3)  TO PDL+0
                        NRTERM                  # E:(-29+NR) M:(-27+NR)
                PDDL    SR*                     # RTERM(2-ALFA RTERM) TO PDL+2
                                                # E:(-32+NR) M:(-30+NR)
                        TFFNP                   # LC P  E:(-38+2NR) M:(-36+2NR)
                        0       -6,1            # X1 = -NR
                DCOMP   DAD                     # DUE TO SHIFTS, KEEP PRECISION FOR SQRT
                                                # RTERM(2-ALFA RTERM) FROM PDL +2
                                                # E:(-32+NR) M:(-30+NR)
                SR*                             # LEAVE  E:(-32) M:(-30)
                        0,1                     # X1 = -NR
                BOFF    DLOAD                   # CHECK TFF /TPER SWITCH
                        TFFSW
                        +2                      # IF TFF, CONTINUE
                        TFFZEROS                # IF TPER, SET Q2 = 0
 +2             BMN     SQRT                    # E:(-16) M:(-15)

                        MAXTFF1                 # NO FREE FALL CONIC TO RTERM FROM HERE
                                                # RESET PDL, SET TFF=POSMAX, AND EXIT.

                DCOMP   BOVB                    # RT IS ON INBOUND SIDE.  ASSURE OVFIND=0
                        TCDANZIG                # ANY PORT IN A STORM.
                STOVL   TFFTEM                  # Q2  E:(-16) M:(-15)
                        VONE'                   # VN/SQRT(MU)  E:(10) M:(9)
                DOT     SL3
                        RONE                    # SAVED RN.  E:(-29) M:(-27)
                STORE   TFFQ1                   # Q1, SAVE FOR GONEPAST TEST.
                                                # E:(-16) M:(-15)
                BMN     BDSU
                        INBOUND                 # USE ALTERNATE Z
                        TFFTEM                  # Q2  E:(-16) M:(-15)

# OUTBOUND Z CALC CONTINUES HERE

                STODL   TFFX                    # NUM=Q2-Q1  E:(-16) M:(-15)
                        TFFALFA                 # ALFA  E:(26-NR) M:(24-NR)
                DMP     BDSU
```

```
                                NRMAG           # RMAG  E:(-29+NR) M:(-27+NR)
                                                # (2-RTERM ALFA)  (-3) FROM PDL+0
        SAVEDEN         PUSH    ABS             # DEN TO PDL+0  E:(-3) OR (-16)
                                                #               M:(-3) OR (-15)
                        DAD     BOV             # INDETERMINANCY TEST
                                LIM(-22)        # =1.0-B(-22)
                                TFFXTEST        # GO IF DEN >/= B(-22)
                        DLOAD   PDDL            # SET DEN=0 OTHERWISE
                                TFFZEROS
                                                # XCH ZERO WITH PDL+0
                        DLOAD   DCOMP
                                TFFALFA         # ALFA  E:(26-NR) M:(24-NR)
                        BMN     DLOAD           # FOR TPER:  Z INDET AT DELE/2=0 AND 90.
                                TFFEL1          # ASSUME 90, AND LEAVE 0 IN PDL: 1/Z=D/N

                                                # Z INDET. AT PERIGEE FOR PARAB OR HYPERB.
        DUMPTFF1        RVQ                     # RETURN TFF =0

# INBOUND Z CALC CONTINUES HERE

        INBOUND         DLOAD                   # RESET PDL+0
                        DLOAD   DSU             # ALTERNATE Z CALC
                                RTERM           # E:(-29) M:(-27)
                                RMAG1           # E:(-29) M:(-27)
                        STODL   TFFX            # NUM=RTERM-RN  E:(-29) M:(-27)
                                TFFTEM          # Q2  E:(-16) M:(-15)
                        DAD     GOTO
                                TFFQ1           # Q1  E:(-16) M:(-15)
                                SAVEDEN         # DEN = Q2+Q1  E:(-16) M:(-15)

        TFFXTEST        DAD     PDDL            # (ABS(DEN) TO PDL+2)   E:(-3) OR (-16)
                                                #                       M:(-3) OR (-15)
                                DP(-22)         # RESTORE ABS(DEN) TO MPAC
                                TFFX            # NUM  E:(-16) OR (-29)  M:(-15) OR (-27)
                        DMP     SR*
                                TFFRTALF        # SQRT(ALFA)  E:(10+NA) M:(9+NA)
                                0       -3,2    # X2=-NA
                        DDV                     # C(MPAC) =NUM SQRT(ALFA)       E:(-3) OR (-16)
                                                #                               M:(-3) OR (-15)
                                                # ABS(DEN) FROM PDL+2   E:(-3) OR (-16)
                                                #                       M:(-3) OR (-15)
                        DLOAD   BOV             # (THE DLOAD IS SHARED WITH TFFELL)
                                TFFX            # NUM  E:(-16) OR (-29)  M:(-15) OR (-27)
                                TFFELL          # USE EQN FOR DELE GEQ 90, LEQ -90

# OTHERWISE, CONTINUE FOR GENERAL CONIC FOR TFF EQN

                        DDV     STADR
                                                # DEN FROM PDL+0        E:(-3) OR (-16)
                                                #                       M:(-3) OR (-15)
                        STORE   TFFTEM          # Z SAVE FOR SIGN OF SDELF.
```

```
                                               # E:(-13) M:(-12)
                PUSH     DSQ                   # Z TO PDL+0
                PUSH     DMP                   # Z SQ TO PDL+2  E:(-26) M:(-24)
                         TFFNP                 # LC P  E:(-38+2NR) M:(-36+NR)
                SL       SIGN
                         5
                         TFFTEM                # AFFIX SIGN FOR SDELF (ENTRY DISPLAY)
                STODL    TFFTEM                # P ZSQ  E:(-59+2NR) M:(-55+2NR)
                                               # (ARG IS USED IN TFF/TRIG)
                                               # ZSQ FROM PDL+2  E:(-26) M:(-24)
                PUSH     DMP                   # RESTORE PUSH LOC
                         TFFALFA               # ALFA  E:(26-NR) M:(24-NR)
                SL*
                         0,1                   # X1=-NR
                STORE    TFFX                  # X
                RTB      DMP
                         T(X)                  # POLY
                                               # ZSQ FROM PDL+2  E:(-26) M:(-24)
                SR2      BDSU                  # 2 ZSQ T(X)  E:(-29) M:(-27)
                         RTERM                 # RTERM  E:(-29) M:(-27)
                DAD      DMP
                         RMAG1                 # E:(-29) M:(-27)
                                               # Z FROM PDL+0  E:(-13) M:(-12)
                SR3      BPL                   # TFF SQRT(MU)  E:(-45) M:(-42)
                         ENDTFF                # (NO PUSH UP)
                PUSH     SIGN                  # TFF SQRT(MU) TO PDL+0
                         TFFQ1                 # Q1 FOR GONEPAST TEST
                BPL      DLOAD                 # GONE PAST ?
                         NEGTFF                # YES. TFF < 0.
                         TFF1/ALF              # 1/ALFA  E:(-22-2NA) M:(-20-2NA)
                DCOMP    BPL                   # ALFA > 0 ?
                         NEGTFF                # NO. TFF IS NEGATIVE.

# CORRECT FOR ORBITAL PERIOD.

                DCOMP                          # YES.  CORRECT FOR ORB PERIOD.
                DMP      DDV
                         PI/16                 # 2 PI (-5)
                         TFFRTALF              # SQRT(ALFA)  E:(10+NA) M:(9+NA)
                SL*      SL*
                         0        -4,2         # X2=-NA
                         0        -4,2
                SL*      DAD
                         0,2
                                               # TFF SQRT(MU) FROM PDL+0       E:(-45) M:(-42)
ENDTFF          DMP      BOV                   # TFF SQRT(MU) IN MPAC          E:(-45) M:(-42)
                         TFF/RTMU              # E:(17) M:(14)
                         MAXTFF                # SET POSMAX IN OVFL.

DUMPTFF2        RVQ                            # RETURN TFF (-28) CS IN MPAC.
```

```
NEGTFF          DLOAD
                                        # TFF SQRT(MU) FROM PDL+0, NEGATIVE.
                GOTO
                        ENDTFF

MAXTFF1         DLOAD                   # RESET PDL
MAXTFF          DLOAD   RVQ
                        NEARONE

# TIME OF FLIGHT ELLIPSE WHEN DEL (ECCENTRIC ANOM) GEQ 90 AND LEQ -90.

                                        # NUM FROM TFFX.        E:(-16) OR (-29)
                                        #                       M:(-15) OR (-27)
TFFELL          SL2                     # NUM  E:(-14) OR (-27)  M:(-13) OR (-25)
                BDDV    PUSH            # TEMP SAVE D/N IN PDL+0
                                        # DEN FROM PDL+0  E:(-3)/(-16)  M:(-3)/(-15)
                                        # N/D TO PDL+0  E:(11) M:(10)
TFFEL1          DLOAD   DSU             # (ENTER WITH D/N=0 IN PDL+0)
                        TFFTEM          # Q2  E:(-16) M:(-15)
                        TFFQ1           # Q1  E:(-16) M:(-15)
                STODL   TFFDELQ         # Q2-Q1  E:(-16) M:(-15)
                                        # D/N FROM PDL+0
                STADR
                STORE   TFFTEM          # D/N  E:(11) M:(10)
                DMP     SL*
                        TFF1/ALF        # 1/ALFA  E:(-22-2NA) M:(-20-2NA)
                        0,2             # 1/ALFA Z  E:(-11-NA) M:(-10-NA)
                PUSH    DMP             # TO PDL+0
                        TFFTEM          # 1/Z  E:(11) M:(10
                SL*     BOVB
                        0,2             # X2= -NA
                        SIGNMPAC        # IN CASE X= 1.0, CONTINUE
                STORE   TFFX            # X=1/ALFA ZSQ
                RTB     DMP
                        T(X)            # POLY
                        TFFX
                SR3     DSU
                        DP2(-3)
                DMP     PUSH            # 2(X T(X)-1) /Z ALFA   E:(-15-NA)
                                        #                       M:(-14-NA)
                                        # 1/ALFA Z FROM PDL+0   E:(-11-NA)
                                        #                       M:(-10-NA)
                DLOAD   DMP             # GET SIGN FOR SDELF
                        TFFTEM          # 1/Z  E:(11) M:(10)
                        RMAG1           # E:(-29) M:(-27)
                SL2     DAD
                        TFFQ1           # Q1  E:(-16) M:(-15)
                STODL   TFFTEM          # (Q1+R 1/Z) =SGN OF SDELF  E:(-16) M:(-15)
                        TFFNP           # LC P  E:(-38+2NR) M:(-36+2NR)
                DMP     SL*             # CALC FOR ARG FOR TFF/TRIG.
```

```
                TFF1/ALF        # 1/ALFA  E:(-22-2NA) M:(-20-2NA)
                1,2             # X2=-NA
        SIGN    SL*
                TFFTEM          # AFFIX SIGN FOR SDELF
                0,2
        STODL   TFFTEM          # P/ALFA  E:(-59+2NR) M:(-55+2NR)
                                # (ARG FOR USE IN TFF/TRIG)
                TFF1/ALF        # 1/ALFA  E:(-22-2NA) M:(-20-2NA)
        SQRT    DMP
                PI/16           # PI (-4)
        DAD
                                # 2(XT(X)-1)/Z ALFA FROM PDL    E:(-15-NA)
                                #                               M:(-14-NA)
        SL*     DSU
                0       -1,2
                TFFDELQ         # Q2-Q1  E:(-16) M:(-15)
        DMP     SL*
                TFF1/ALF        # 1/ALFA  E:(-22-2NA) M:(-20-2NA)
                0       -3,2
        SL*     GOTO
                0       -4,2
                ENDTFF          # TFF SQRT(MU) IN MPAC E:(-45) M:(-42)
```

```
# PROGRAM NAME:  T(X)                             DATE:  01.17.67
# MOD NO:  0                                      LOG SECTION:  TIME OF FREE FALL
# MOD BY:  RR BAIRNSFATHER
#
# FUNCTIONAL DESCRIPTION:  THE POLYNOMIAL T(X) IS USED BY TIME OF FLIGHT SUBROUTINES CALCTFF AND
#       CALCTPER TO APPROXIMATE THE SERIES
#                          2     3
#               1/3 -X/5 +X /7 -X /9 ...
#
#       WHERE   X = ALFA Z Z            IF ALFA Z Z LEQ 1
#               X = 1/(ALFA Z Z)        IF ALFA Z Z G 1
#
#       ALSO    X IS NEG FOR HYPERBOLIC ORBITS
#               X = 0 FOR PARABOLIC ORBITS
#               X IS POSITIVE FOR ELLIPTIC ORBITS
#
#       FOR FLIGHT 278, THE POLYNOMIAL T(X) IS FITTED OVER THE RANGE (0,+1) AND HAS A MAXIMUM
#       DEVIATION FROM THE SERIES OF 2 E-5.  (T(X) IS A CHEBYCHEV TYPE FIT AND WAS OBTAINED USING
#       MAX PROGRAM AUTOCURFIT294RRB AND IS VALID TO THE SAME TOLERANCE OVER THE RANGE (-.08,+1).)
#
# CALLING SEQUENCE:     RTB
#                               T(X)
#       C(MPAC) = X
#
# SUBROUTINE CALLED:  NONE
#
# NORMAL EXIT MODE:  TC TANZIG
#
# ALARMS:  NONE
#
# OUTPUT:  C(MPAC) = T(X)
#
# ERASABLE INITIALIZATION REQUIRED:
#       C(MPAC) = X
#
# DEBRIS:  NONE

T(X)            TC      POLY
                DEC     4               # N-1
                2DEC    3.333333333 E-1

                2DEC*   -1.999819135 E-1*

                2DEC*   1.418148467 E-1*

                2DEC*   -1.01310997 E-1*

                2DEC*   5.609004986 E-2*

                2DEC*   -1.536156925 E-2*

ENDT(X)         TC      DANZIG

TCDANZIG        =       ENDT(X)
```

```
# TFF CONSTANTS

                BANK    32

                SETLOC  TOF-FF1
                BANK

#                                       # NOTE:  ADJUSTED MUE FOR NEAR EARTH TRAJ.
#MUE            =       3.990815471 E10         # M CUBE/CS SQ
#RTMUE          =       1.997702549 E5 B-18*    # MODIFIED EARTH MU
#
#                                       # NOTE:  ADJUSTED MUE FOR NEAR EARTH TRAJ.
#MUM            =       4.902778 E8             # M CUBE/CS SQ
#RTMUM          2DEC*   2.21422176 E4 B-18*

PI/16           2DEC    3.141592653 B-4

LIM(-22)        2OCT    3777737700              # 1.0 -B(-22)

DP(-22)         2OCT    0000000100              # B(-22)

DP2(-3)         2DEC    1 B-3

DP2(-4)         2DEC    1 B-4                   # 1/16

# RPAD1         2DEC    6373338 B-29            # M (-29) = 20909901.57 FT

RPAD1           =       RPAD

R300K           2DEC    6464778 B-29            # (-29) M

NEARONE         2DEC    .999999999

TFFZEROS        EQUALS  HI6ZEROS
TFF1/4          EQUALS  HIDP1/4
```

```
# PROGRAM DESCRIPTION                           DATE:  20 DECEMBER 1967
# PROGRAM NAME -- SELF-CHECK                    LOG SECTION:  AGC BLOCK TWO SELF-CHECK
# MOD NO -- 1                                   ASSEMBLY SUBROUTINE UTILITYM REV 25
# MOD BY -- GAUNTT
#
#
# FUNCTIONAL DESCRIPTION
#
#       PROGRAM HAS TWO MAIN PARTS.  THE FIRST IS SELF-CHECK WHICH RUNS AS A ZERO PRIORITY JOB WITH NO CORE SET, AS
#       PART OF THE BACK-UP IDLE LOOP.  THE SECOND IS SHOW-BANKSUM WHICH RUNS AS A REGULAR EXECUTIVE JOB WITH ITS OWN
#       STARTING VERB.
#       THE PURPOSE OF SELF-CHECK IS TO CHECK OUT VARIOUS PARTS OF THE COMPUTER AS OUTLINED BELOW IN THE OPTIONS.
#       THE PURPOSE OF SHOW-BANKSUM IS TO DISPLAY THE SUM OF EACH BANK, ONE AT A TIME.
#       IN ALL THERE ARE 7 POSSIBLE OPTIONS IN THIS BLOCK II VERSION OF SELF-CHECK.  MORE DETAIL DESCRIPTION MAY BE
#       FOUND IN E-2065 BLOCK II AGC SELF-CHECK AND SHOW BANKSUM BY EDWIN D. SMALLY DECEMBER 1966, AND ADDENDA 2 AND 3.
#       THE DIFFERENT OPTIONS ARE CONTROLLED BY PUTTING DIFFERENT NUMBERS IN THE SMODE REGISTER (NOUN 27).  BELOW IS
#       A DESCRIPTION OF WHAT PARTS OF THE COMPUTER THAT ARE CHECKED BY THE OPTIONS, AND THE CORRESPONDING NUMBER, IN
#       OCTAL, TO LOAD INTO SMODE.
#               +-4             ERASABLE MEMORY
#               +-5             FIXED MEMORY
#               +-1,2,3,6,7,10  EVERYTHING IN OPTIONS 4 AND 5.
#               -0              SAME AS +-10 UNTIL AN ERROR IS DETECTED.
#               +0              NO CHECK, PUTS COMPUTER INTO THE BACKUP IDLE LOOP.
#
#
# WARNINGS
#
#       USE OF E MEMORY RESERVED FOR SELF-CHECK (EVEN IN IDLE LOOP) AS TEMP STORAGE BY OTHER PROGRAMS IS DANGEROUS.
#       SMODE SET GREATER THAN OCT 10 PUTS COMPUTER INTO BACKUP IDLE LOOP.
#
#
# CALLING SEQUENCE
#
#       TO CALL SELF-CHECK KEY IN
#               V 21 N 27 E     OPTION NUMBER E
#       TO CALL SHOW-BANKSUM KEY IN
#               V 91 E          DISPLAYS FIRST BANK
#               V 33 E          PROCEED, DISPLAYS NEXT BANK
#
#
# EXIT MODES, NORMAL AND ALARM
#       SELF-CHECK NORMALLY CONTINUES INDEFINITELY UNLESS THERE IS AN ERROR DETECTED.  IF SO + OPTION NUMBERS PUT
#       COMPUTER INTO BACKUP IDLE LOOP, - OPTION NUMBERS RESTART THE OPTION.
#
#       THE -0 OPTION PROCEEDS FROM THE LINE FOLLOWING THE LINE WHERE THE ERROR WAS DETECTED.
#       SHOW-BANKSUM PROCEEDS UNTIL A TERMINATE IS KEYED IN (V 34 E).  THE COMPUTER IS PUT INTO THE BACKUP IDLE LOOP.
#
#
# OUTPUT
```

```
#       SELF-CHECK UPON DETECTING AN ERROR LOADS THE SELF-CHECK ALARM CONSTANT (01102) INTO THE FAILREG SET AND
#       TURNS ON THE ALARM LIGHT.  THE OPERATOR MAY THEN DISPLAY THE THREE FAILREGS BY KEYING IN V 05 N 09 E.  FOR FURTHER
#       INFORMATION HE MAY KEY IN V 05 N 08 E, THE DSKY DISPLAY IN R1 WILL BE ADDRESS+1 OF WHERE THE ERROR WAS DETECTED,
#       IN R2 THE BBCON OF SELF-CHECK, AND IN R3 THE TOTAL NUMBER OF ERRORS DETECTED BY SELF-CHECK SINCE THE LAST MAN
#       INITIATED FRESH START (SLAP1).
#       SHOW-BANKSUM STARTING WITH BANK 0 DISPLAYS IN R1 THE BANK SUM (A +-NUMBER EQUAL TO THE BANK NUMBER), IN R2
#       THE BANK NUMBER, AND IN R3 THE BUGGER WORD.
#
#
# ERASABLE INITIALIZATION REQUIRED
#
#       ACCOMPLISHED BY FRESH START
#               SMODE SET TO +0
#
#
# DEBRIS
#
#       ALL EXITS FROM THE CHECK OF ERASABLE (ERASCHK) RESTORE ORIGINAL CONTENTS TO REGISTERS UNDER CHECK.
#       EXCEPTION IS A RESTART.  RESTART THAT OCCURS DURING ERASCHK RESTORES ERASABLE, UNLESS THERE IS EVIDENCE TO DOUBT
#       E MEMORY, IN WHICH CASE PROGRAM THEN DOES A FRESH START (DOFSTART).


                BANK    25
                SETLOC  SELFCHEC
                BANK

                COUNT*  $$/SELF
SBIT1           EQUALS  BIT1
SBIT2           EQUALS  BIT2
SBIT3           EQUALS  BIT3
SBIT4           EQUALS  BIT4
SBIT5           EQUALS  BIT5
SBIT6           EQUALS  BIT6
SBIT7           EQUALS  BIT7
SBIT8           EQUALS  BIT8
SBIT9           EQUALS  BIT9
SBIT10          EQUALS  BIT10
SBIT11          EQUALS  BIT11
SBIT12          EQUALS  BIT12
SBIT13          EQUALS  BIT13
SBIT14          EQUALS  BIT14
SBIT15          EQUALS  BIT15

S+ZERO          EQUALS  ZERO
S+1             EQUALS  BIT1
S+2             EQUALS  BIT2
S+3             EQUALS  THREE
S+4             EQUALS  FOUR
S+5             EQUALS  FIVE
S+6             EQUALS  SIX
```

```
S+7             EQUALS  SEVEN
S8BITS          EQUALS  LOW8            # 00377
CNTRCON         =       OCT50           # USED IN CNTRCHK
ERASCON1        OCTAL   00061           # USED IN ERASCHK
ERASCON2        OCTAL   01373           # USED IN ERASCHK
ERASCON6        =       OCT1400         # USED IN ERASCHK
ERASCON3        OCTAL   01461           # USED IN ERASCHK
ERASCON4        OCTAL   01773           # USED IN ERASCHK
S10BITS         EQUALS  LOW10           # 01777, USED IN ERASCHK
SBNK03          EQUALS  PRIO6           # 06000, USED IN ROPECHK
-MAXADRS        =       HI5             # FOR ROPECHK
SIXTY           OCTAL   00060
SUPRCON         OCTAL   60017           # USED IN ROPECHK
S13BITS         OCTAL   17777
CONC+S1         OCTAL   25252           # USED IN CYCLSHFT
CONC+S2         OCTAL   52400           # USED IN CYCLSHFT
ERASCON5        OCTAL   76777
S-7             =       OCT77770
S-4             EQUALS  NEG4
S-3             EQUALS  NEG3
S-2             EQUALS  NEG2
S-1             EQUALS  NEGONE
S-ZERO          EQUALS  NEG0

                EBANK=  LST1
ADRS1           ADRES   SKEEP1
SELFADRS        ADRES   SELFCHK         # SELFCHK RETURN ADDRESS.  SHOULD BE PUT
                                        # IN SELFRET WHEN GOING FROM SELFCHK TO
                                        # SHOWSUM AND PUT IN SKEEP1 WHEN GOING
                                        # FROM SHOWSUM TO SELF-CHECK.

PRERRORS        CA      ERESTORE        # IS IT NECESSARY TO RESTORE ERASABLE
                EXTEND
                BZF     ERRORS          # NO
                EXTEND
                DCA     SKEEP5
                INDEX   SKEEP7
                DXCH    0000            # RESTORE THE TWO ERASABLE REGISTERS
                CA      S+ZERO
                TS      ERESTORE
ERRORS          INHINT
                CA      Q
                TS      SFAIL           # SAVE Q FOR FAILURE LOCATION
                TS      ALMCADR         # FOR DISPLAY WITH BBANK AND ERCOUNT
                INCR    ERCOUNT         # KEEP TRACK OF NUMBER OF MALFUNCTIONS.
TCALARM2        TC      ALARM2
                OCT     01102           # SELF-CHECK MALFUNCTION INDICATOR
                CCS     SMODE
SIDLOOP         CA      S+ZERO
                TS      SMODE
```

```
                TC      SELFCHK         # GO TO IDLE LOOP
                TC      SFAIL           # CONTINUE WITH SELF-CHECK

-1CHK           CCS     A
                TCF     PRERRORS
                TCF     PRERRORS
                CCS     A
                TCF     PRERRORS
                TC      Q

SMODECHK        EXTEND
                QXCH    SKEEP1
                TC      CHECKNJ         # CHECK FOR NEW JOB
                CCS     SMODE
                TC      SOPTIONS
                TC      SMODECHK +2     # TO BACKUP IDLE LOOP
                TC      SOPTIONS
                INCR    SCOUNT
                TC      SKEEP1          # CONTINUE WITH SELF-CHECK

SOPTIONS        AD      S-7
                EXTEND
                BZMF    +2              # FOR OPTIONS BELOW NINE.
BNKOPTN         TC      SIDLOOP         # ILLEGAL OPTION.  GO TO IDLE LOOP.
                INCR    SCOUNT          # FOR OPTIONS BELOW NINE.
                AD      S+7

                INDEX   A
                TC      SOPTION1
SOPTION1        TC      SKEEP1          # WAS TC+TCF
SOPTION2        TC      SKEEP1          # WAS IN:OUT1
SOPTION3        TC      SKEEP1          # WAS COUNTCHK
SOPTION4        TC      ERASCHK
SOPTION5        TC      ROPECHK
SOPTION6        TC      SKEEP1
SOPTION7        TC      SKEEP1
SOPTON10        TC      SKEEP1          # CONTINUE WITH SELF-CHECK

CHECKNJ         EXTEND
                QXCH    SELFRET         # SAVE RETURN ADDRESS WHILE TESTING NEWJOB
                TC      POSTJUMP        # TO SEE IF ANY JOBS HAVE BECOME ACTIVE.
                CADR    ADVAN

SELFCHK         TC      SMODECHK        # ** CHARLEY, COME IN HERE

# SKEEP7 HOLDS LOWEST OF TWO ADDRESSES BEING CHECKED.
# SKEEP6 HOLDS B(X+1).
# SKEEP5 HOLDS B(X).
# SKEEP4 HOLDS C(EBANK) DURING ERASLOOP AND CHECKNJ.
# SKEEP3 HOLDS LAST ADDRESS BEING CHECKED (HIGHEST ADDRESS).
```

```
# SKEEP2 CONTROLS CHECKING OF NON-SWITCHABLE ERASABLE MEMORY WITH BANK NUMBERS IN EB.
# ERASCHK TAKES APPROXMATELY 7 SECONDS
ERASCHK         CA      S+1
                TS      SKEEP2
OEBANK          CA      S+ZERO
                TS      EBANK
                CA      ERASCON3        # 01461
                TS      SKEEP7          # STARTING ADDRESS
                CA      S10BITS         # 01777
                TS      SKEEP3          # LAST ADDRESS CHECKED
                TC      ERASLOOP

E134567B        CA      ERASCON6        # 01400
                TS      SKEEP7          # STARTING ADDRESS
                CA      S10BITS         # 01777
                TS      SKEEP3          # LAST ADDRESS CHECKED
                TC      ERASLOOP

2EBANK          CA      ERASCON6        # 01400
                TS      SKEEP7          # STARTING ADDRESS
                CA      ERASCON4        # 01773
                TS      SKEEP3          # LAST ADDRESS CHECKED
                TC      ERASLOOP

NOEBANK         TS      SKEEP2          # +0
                CA      ERASCON1        # 00061
                TS      SKEEP7          # STARTING ADDRESS
                CA      ERASCON2        # 01373
                TS      SKEEP3          # LAST ADDRESS CHECKED

ERASLOOP        INHINT
                CA      EBANK           # STORES C(EBANK)
                TS      SKEEP4
                EXTEND
                NDX     SKEEP7
                DCA     0000
                DXCH    SKEEP5          # STORES C(X) AND C(X+1) IN SKEEP6 AND 5.
                CA      SKEEP7
                TS      ERESTORE        # IF RESTART, RESTORE C(X) AND C(X+1)
                TS      L
                INCR    L
                NDX     A
                DXCH    0000            # PUTS OWN ADDRESS IN X AND X +1
                NDX     SKEEP7
                CS      0001            # CS X+1
                NDX     SKEEP7
                AD      0000            # AD X
                TC      -1CHK
                CA      ERESTORE        # HAS ERASABLE BEEN RESTORED
                EXTEND
```

```
                BZF     ELOOPFIN        # YES, EXIT ERASLOOP.
                EXTEND
                NDX     SKEEP7
                DCS     0000            # COMPLEMENT OF ADDRESS OF X AND X+1
                NDX     SKEEP7
                DXCH    0000            # PUT COMPLEMENT OF ADDRESS OF X AND X+1
                NDX     SKEEP7
                CS      0000            # CS X
                NDX     SKEEP7
                AD      0001            # AD X+1
                TC      -1CHK
                CA      ERESTORE        # HAS ERASABLE BEEN RESTORED
                EXTEND
                BZF     ELOOPFIN        # YES, EXIT ERASLOOP.
                EXTEND
                DCA     SKEEP5
                NDX     SKEEP7
                DXCH    0000            # PUT B(X) AND B(X+1) BACK INTO X AND X+1
                CA      S+ZERO
                TS      ERESTORE        # IF RESTART, DO NOT RESTORE C(X), C(X+1)
ELOOPFIN        RELINT
                TC      CHECKNJ         # CHECK FOR NEW JOB
                CA      SKEEP4          # REPLACES B(EBANK)
                TS      EBANK
                INCR    SKEEP7
                CS      SKEEP7
                AD      SKEEP3
                EXTEND
                BZF     +2
                TC      ERASLOOP        # GO TO NEXT ADDRESS IN SAME BANK
                CCS     SKEEP2
                TC      NOEBANK
                INCR    SKEEP2          # PUT +1 IN SKEEP2.
                CA      EBANK
                AD      SBIT9
                TS      EBANK
                AD      ERASCON5        # 76777, CHECK FOR BANK E2
                EXTEND
                BZF     2EBANK
                CCS     EBANK
                TC      E134567B        # GO TO EBANKS 1,3,4,5,6, AND 7
                CA      ERASCON6        # END OF ERASCHK
                TS      EBANK
# CNTRCHK PERFORMS A CS OF ALL REGISTERS FROM OCT. 60 THROUGH OCT. 10.
# INCLUDED ARE ALL COUNTERS, T6-1, CYCLE AND SHIFT, AND ALL RUPT REGISTERS
CNTRCHK         CA      CNTRCON         # 00050
CNTRLOOP        TS      SKEEP2
                AD      SBIT4           # +10 OCTAL
                INDEX   A
                CS      0000
```

```
                CCS     SKEEP2
                TC      CNTRLOOP

# CYCLSHFT CHECKS THE CYCLE AND SHIFT REGISTERS
CYCLSHFT        CA      CONC+S1         # 25252
                TS      CYR             # C(CYR) = 12525
                TS      CYL             # C(CYL) = 52524
                TS      SR              # C(SR) = 12525
                TS      EDOP            # C(EDOP) = 00125
                AD      CYR             # 37777         C(CYR) = 45252
                AD      CYL             # 00-12524      C(CYL) = 25251
                AD      SR              # 00-25251      C(SR) = 05252
                AD      EDOP            # 00-25376      C(EDOP) = +0
                AD      CONC+S2         # C(CONC+S2) = 52400
                TC      -1CHK
                AD      CYR             # 45252
                AD      CYL             # 72523
                AD      SR              # 77775
                AD      EDOP            # 77775
                AD      S+1             # 77776
                TC      -1CHK

                INCR    SCOUNT +1
                TC      SMODECHK

# SKEEP1 HOLDS SUM
# SKEEP2 HOLDS PRESENT CONTENTS OF ADDRESS IN ROPECHK AND SHOWSUM ROUTINES
# SKEEP2 HOLDS BANK NUMBER IN LOW ORDER BITS DURING SHOWSUM DISPLAY
# SKEEP3 HOLDS PRESENT ADDRESS (00000 TO 01777 IN COMMON FIXED BANKS)
#                              (04000 TO 07777 IN FXFX BANKS)
# SKEEP3 HOLDS BUGGER WORD DURING SHOWSUM DISPLAY
# SKEEP4 HOLDS BANK NUMBER AND SUPER BANK NUMBER
# SKEEP5 COUNTS 2 SUCCESSIVE TC SELF WORDS
# SKEEP6 CONTROLS ROPECHK OR SHOWSUM OPTION
# SKEEP7 CONTROLS WHEN ROUNTINE IS IN COMMON FIXED OR FIXED FIXED BANKS

ROPECHK         CA      S-ZERO          # *
                TS      SKEEP6          # * -0 FOR ROPECHK.
STSHOSUM        CA      S+ZERO          # * SHOULD BE ROPECHK

                TS      SKEEP4          # BANK NUMBER
                CA      S+1
COMMFX          TS      SKEEP7
                CA      S+ZERO
                TS      SKEEP1
                TS      SKEEP3
                CA      S+1
                TS      SKEEP5          # COUNTS DOWN 2 TC SELF WORDS
COMADRS         CA      SKEEP4
                TS      L               # TO SET SUPER BANK
                MASK    HI5
```

```
                AD      SKEEP3
                TC      SUPDACAL        # SUPER DATA CALL
                TC      ADSUM
                AD      SBIT11          # 02000
                TC      ADRSCHK

FXFX            CS      A
                TS      SKEEP7
                EXTEND
                BZF     +3
                CA      SBIT12          # 04000, STARTING ADDRESS OF BANK 02
                TC      +2
                CA      SBNK03          # 06000, STARTING ADDRESS OF BANK 03
                TS      SKEEP3
                CA      S+ZERO
                TS      SKEEP1
                CA      S+1
                TS      SKEEP5          # COUNTS DOWN 2 TC SELF WORDS
FXADRS          INDEX   SKEEP3
                CA      0000
                TC      ADSUM
                TC      ADRSCHK

ADSUM           TS      SKEEP2
                AD      SKEEP1
                TS      SKEEP1
                CAF     S+ZERO
                AD      SKEEP1
                TS      SKEEP1
                CS      SKEEP2
                AD      SKEEP3
                TC      Q

ADRSCHK         LXCH    A
                CA      SKEEP3
                MASK    LOW10           # RELATIVE ADDRESS
                AD      -MAXADRS        # SUBTRACT MAX RELATIVE ADDRESS = 1777.
                EXTEND
                BZF     SOPTION         # CHECKSUM FINISHED IF LAST ADDRESS.
                CCS     SKEEP5          # IS CHECKSUM FINISHED
                TC      +3              # NO
                TC      +2              # NO
                TC      SOPTION         # GO TO ROPECHK SHOWSUM OPTION
                CCS     L               # -0 MEANS A TC SELF WORD.
                TC      CONTINU
                TC      CONTINU
                TC      CONTINU
                CCS     SKEEP5
                TC      CONTINU +1
                CA      S-1
```

```
                TC      CONTINU +1      # AD IN THE BUGGER WORD
CONTINU         CA      S+1             # MAKE SURE TWO CONSECUTIVE TC SELF WORDS
                TS      SKEEP5
                CCS     SKEEP6          # *
                CCS     NEWJOB          # * +1, SHOWSUM
                TC      CHANG1          # *
                TC      +2              # *
                TC      CHECKNJ         # -0 IN SKEEP6 FOR ROPECHK

ADRS+1          INCR    SKEEP3
                CCS     SKEEP7
                TC      COMADRS
                TC      COMADRS
                TC      FXADRS
                TC      FXADRS

NXTBNK          CS      SKEEP4
                AD      LSTBNKCH        # LAST BANK TO BE CHECKED
                EXTEND
                BZF     ENDSUMS         # END OF SUMMING OF BANKS.
                CA      SKEEP4
                AD      SBIT11
                TS      SKEEP4          # 37 TO 40 INCRMTS SKEEP4 BY END RND CARRY
                TC      CHKSUPR
17TO20          CA      SBIT15
                ADS     SKEEP4          # SET FOR BANK 20
                TC      GONXTBNK
CHKSUPR         MASK    HI5
                EXTEND
                BZF     NXTSUPR         # INCREMENT SUPER BANK
27TO30          AD      S13BITS
                EXTEND
                BZF     +2              # BANK SET FOR 30
                TC      GONXTBNK
                CA      SIXTY           # FIRST SUPER BANK
                ADS     SKEEP4
                TC      GONXTBNK
NXTSUPR         AD      SUPRCON         # SET BNK 30 + INCR SUPR BNK AND CANCEL
                ADS     SKEEP4          # ERC BIT OF THE 37 TO 40 ADVANCE.
GONXTBNK        CCS     SKEEP7
                TC      COMMFX
                CA      S+1
                TC      FXFX
                CA      SBIT7           # HAS TO BE LARGER THAN NO OF FXSW BANKS.
                TC      COMMFX

SOPTION         CA      SKEEP4
                MASK    HI5             # = BANK BITS
                TC      LEFT5
                TS      L               # BANK NUMBER BEFORE SUPER BANK
```

```
                CA      SKEEP4
                MASK    S8BITS          # = SUPER BANK BITS
                EXTEND
                BZF     SOPT            # BEFORE SUPER BANK
                TS      SR              # SUPER BANK NECESSARY
                CA      L
                MASK    SEVEN
                AD      SR
                TS      L               # BANK NUMBER WITH SUPER BANK
SOPT            CA      SKEEP6          # *
                EXTEND                  # *
                BZF     +2              # * ON -0 CONTINUE WITH ROPE CHECK.
                TC      SDISPLAY        # * ON +1 GO TO DISPLAY OF SUM.
                CCS     SKEEP1          # FORCE SUM TO ABSOLUTE VALUE.
                TC      +2
                TC      +2
                AD      S+1
                TS      SKEEP1
BNKCHK          CS      L               # = - BANK NUMBER
                AD      SKEEP1
                AD      S-1
                TC      -1CHK           # CHECK SUM
                TC      NXTBNK

                EBANK=  NEWJOB
LSTBNKCH        BBCON*                  # * CONSTANT, LAST BANK.
```

```
# SUBROUTINE TO UPDATE THE PROGRAM NUMBER DISPLAY ON THE DSKY.

                COUNT*  $$/PHASE
                BLOCK   02
                SETLOC  FFTAG1
                BANK

NEWMODEX        INDEX   Q               # UPDATE MODREG.  ENTRY FOR MODE IN FIXED.
                CAF     0
                INCR    Q

NEWMODEA        TS      MODREG          # ENTRY FOR MODE IN A.
MMDSPLAY        CAF     +3              # DISPLAY MAJOR MODE.
PREBJUMP        LXCH    BBANK           # PUTS BBANK IN L
                TCF     BANKJUMP        # PUTS Q INTO A
                CADR    SETUPDSP

# RETURN TO CALLER +3 IF MODE = THAT AT CALLER +1.  OTHERWISE RETURN TO CALLER +2.

CHECKMM         INDEX   Q
                CS      0
                AD      MODREG
                EXTEND
                BZF     Q+2
                TCF     Q+1             # NO MATCH

TCQ             =       Q+2 +1

                BANK    14
                SETLOC  PHASETAB
                BANK

                COUNT*  $$/PHASE
SETUPDSP        INHINT
                DXCH    RUPTREG1        # SAVE CALLER'S RETURN 2CADR
                CAF     PRIO30          #       EITHER A TASK OR JOB CAN COME TO
                TC      NOVAC           #       NEWMODE X
                EBANK=  MODREG
                2CADR   DSPMMJOB

                DXCH    RUPTREG1
                RELINT
                DXCH    Z               # RETURN

DSPMMJOB        EQUALS  DSPMMJB

                BLOCK   02
                SETLOC  FFTAG1
                BANK
```

```
# PHASCHNG IS THE MAIN WAY OF MAKING PHASE CHANGES FOR RESTARTS.  THERE ARE THREE FORMS OF PHASCHNG, KNOWN AS TYPE
# A, TYPE B, AND TYPE C.  THEY ARE ALL CALLED AS FOLLOWS, WHERE OCT XXXXX CONTAINS THE PHASE INFORMATION,
#		TC	PHASCHNG
#		OCT	XXXXX
# TYPE A IS CONCERNED WITH FIXED PHASE CHANGES, THAT IS, PHASE INFORMATION THAT IS STORED PERMANENTLY.  THESE
# OPTIONS ARE, WHERE G STANDS FOR A GROUP AND .X FOR THE PHASE,
#	G.0		INACTIVE, WILL NOT PERMIT A GROUP G RESTART
#	G.1		WILL CAUSE THE LAST DISPLAY TO BE REACTIVATED, USED MAINLY IN MANNED FLIGHTS
#	G.EVEN		A DOUBLE TABLE RESTART, CAN CAUSE ANY COMBINATION OF TWO JOBS, TASKS, AND/OR
#			LONGCALL TO BE RESTARTED.
#	G.ODD NOT .1	A SINGLE TABLE RESTART, CAN CAUSE EITHER A JOB, TASK, OR LONGCALL RESTART.
#
# THIS INFORMATION IS PUT INTO THE OCTAL WORD AFTER TC PHASCHNG AS FOLLOWS
#	TLO OOP PPP PPP GGG
# WHERE EACH LETTER OR NUMBER STANTS FOR A BIT.  THE G'S STAND FOR THE GROUP, OCTAL 1-7, THE P'S FOR THE PHASE,
# OCTAL 0 - 127.  O'S MUST BE 0.  IF ONE WISHES TO HAVE THE TBASE OF GROUP G TO BE SET AT THIS TIME,
# T IS SET TO 1, OTHERWISE IT IS SET TO 0.  SIMILARLY IF ONE WISHES TO SET LONGBASE, THEN L IS SET TO 1, OTHERWISE
# IT IS SET TO 0.  SOME EXAMPLES,
#		TC	PHASCHNG		# THIS WILL CAUSE GROUP 3 TO BE SET TO 0,
#		OCT	00003			# MAKING GROUP 3 INACTIVE
#
#		TC	PHASCHNG		# IF A RESTART OCCURS THIS WOULD CAUSE
#		OCT	00012			# GROUP 2 TO RESTART THE LAST DISPLAY
#
#		TC	PHASCHNG		# THIS SETS THE TBASE OF GROUP 4 AND IN
#		OCT	40064			# CASE OF A RESTART WOULD START UP THE TWO
#					# THINGS LOCATED IN THE DOUBLE 4.6 RESTART
#					# LOCATION.
#
#		TC	PHASCHNG		# THIS SETS LONGBASE AND UPON A RESTART
#		OCT	20135			# CAUSES 5.13 TO BE RESTARTED (SINCE
#					# LONGBASE WAS SET THIS SINGLE ENTRY
#					# SHOULD BE A LONGCALL)
#
#		TC	PHASCHNG		# SINCE BOTH TBASE4 AND LONGBASE ARE SET,
#		OCT	60124			# 4.12 SHOULD CONTAIN BOTH A TASK AND A
#					# LONGCALL TO BE RESTARTED
#
# TYPE C PHASCHNG CONTAINS THE VARIABLE TYPE OF PHASCHNG INFORMATION.  INSTEAD OF THE INFORMATION BEING IN A
# PERMANENT FORM, ONE STORES THE DESIRED RESTART INFORMATION IN A VARIABLE LOCATION.  THE BITS ARE AS FOLLOWS,
#	TLO 1AD XXX CJW GGG
# WHERE EACH LETTER OR NUMBER STANDS FOR A BIT.  THE G'S STAND FOR THE GROUP, OCTAL 1 - 7.  IF THE RESTART IS TO
# BE BY WAITLIST, W IS SET TO 1, IF IT IS A JOB, J IS SET TO 1, IF IT IS A LONGCALL, C IS SET TO 1.  ONLY ONE OF
# THESE THREE BITS MAY BE SET.  X'S ARE IGNORED, 1 MUST BE 1, AND 0 MUST BE 0.  AGAIN T STANDS FOR THE TBASE,
```

```
# AND L FOR LONGBASE.  THE BITS A AND D ARE CONCERNED WITH THE VARIABLE INFORMATION.  IF D IS SET TO 1, A PRIORITY
# OR DELTA TIME WILL BE READ FROM THE NEXT LOCATION AFTER THE OCTAL INFORMATION., IF THIS IS TO BE INDIRECT, THAT
# IS, THE NAME OF A LOCATION CONTAINING THE INFORMATION (DELTA TIME ONLY), THEN THIS IS GIVEN AS THE -GENADR OF
# THAT LOCATION WHICH CONTAINS THE DELTA TIME.  IF THE OLD PRIORITY OR DELTA TIME IS TO BE USED, THAT WHICH IS
# ALREADY IN THE VARIABLE STORAGE, THEN D IS SET TO 0.  NEXT THE A BIT IS USED.  IF IT IS SET TO 0, THE ADDRESS
# THAT WOULD BE RESTARTED DURING A RESTART IS THE NEXT LOCATION AFTER THE PHASE INFORMATION, THAT IS, EITHER
# (TC PHASCHNG) +2 OR +3, DEPENDING ON WHETHER D HAD BEEN SET OR NOT.  IF A IS SET TO 1, THEN THE ADDRESS THAT
# WOULD BE RESTARTED IS THE 2CADR THAT IS READ FROM THE NEXT TWO LOCATION.  EXAMPLES,
#       AD      TC      PHASCHNG        # THIS WOULD CAUSE LOCATION AD +3 TO BE
#       AD+1    OCT     05023           # RESTARTED BY GROUP THREE WITH A PRIORITY
#       AD+2    OCT     23000           # OF 23.  NOTE UPON RETURNING IT WOULD
#       AD+3                            # ALSO GO TO AD+3
#
#       AD      TC      PHASCHNG        # GROUP 1 WOULD CAUSE CALLCALL TO BE
#       AD+1    OCT     27441           # BE STARTED AS A LONGCALL FROM THE TIME
#       AD+2    -GENADR DELTIME         # STORED IN LONGBASE (LONGBASE WAS SET) BY
#       AD+3    2CADR   CALLCALL        # A DELTA TIME STORED IN DELTIME.  THE
#       AD+4                            # BBCON OF THE 2CADR SHOULD CONTAIN THE E
#       AD+5                            # BANK OF DELTIME.  PHASCHNG RETURNS TO
#                                       # LOCATION AD+5
#
# NOTE THAT IF A VARIABLE PRIORITY IS GIVEN FOR A JOB, THE JOB WILL BE RESTARTED AS A NOVAC IF THE PRIORITY IS
# NEGATIVE, AS A FINDVAC IF THE PRIORITY IS POSITIVE.
#
# TYPE B PHASCHNG IS A COMBINATION OF VARIABLE AND FIXED PHASE CHANGES.  IT WILL START UP A JOB AS INDICATED
# BELOW AND ALSO START UP ONE FIXED RESTART, THAT IS EITHER AN G.1 OR A G.ODD OR THE FIRST ENTRY OF G.EVEN
# DOUBLE ENTRY.  THE BIT INFORMATION IS AS FOLLOW,
#       TL1 DAP PPP PPP GGG
# WHERE EACH LETTER OR NUMBER STANDS FOR A BIT.  THE G'S STAND FOR THE GROUP, OCTAL 1 - 7, THE P'S FOR THE FIXED
# PHASE INFORMATION, OCTAL 0 - 127.  1 MUST BE 1.  AND AGAIN T STANDS FOR THE TBASE AND L FOR LONGBASE.  D THIS
# TIME STANDS ONLY FOR PRIORITY SINCE THIS WILL BE CONSIDERED A JOB, AND IT MUST BE GIVEN DIRECTLY IF GIVEN.
# AGAIN A STANDS FOR THE ADDRESS OF THE LOCATION TO BE RESTARTED, 1 IF THE 2CADR IS GIVEN, OR 0 IF IT IS TO BE
# THE NEXT LOCATION.  (THE RETURN LOCATION OF PHASCHNG) EXAMPLES,
#       AD      TC      PHASCHNG        # TBASE IS SET AND A RESTART CAUSE GROUP 3
#       AD+1    OCT     56043           # TO START THE JOB AJOBAJOB WITH PRIORITY
#       AD+2    OCT     31000           # 31 AND THE FIRST ENTRY OF 3.4SPOT (WE CAN
#       AD+3    2CADR   AJOBAJOB        # ASSUME IT IS A TASK SINCE WE SET TBASE3)
#       AD+4                            # UPON RETURN FROM PHASCHNG CONTROL WOULD
#       AD+5                            # GO TO AD+5
#
#       AD      TC      PHASCHNG        # UPON A RESTART THE LAST DISPLAY WOULD BE
#       AD+1    OCT     10015           # RESTARTED AND A JOB WITH THE PREVIOUSLY
#       AD+2                            # STORED PRIORITY WOULD BE BEGUN AT AD+2
#                                       # BY MEANS OF GROUP 5
```

```
# THE NOVAC-FINDVAC CHOICE FOR JOBS HOLDS HERE ALSO -- NEGATIVE PRIORITY CAUSES A NOVAC CALL, POSITIVE A FINDVAC.

# SUMMARY OF BITS:
#       TYPE A          TLO OOP PPP PPP GGG
#       TYPE B          TL1 DAP PPP PPP GGG
#       TYPE C          TLO 1AD XXX CJW GGG
```

```
# 2PHSCHNG IS USED WHEN ONE WISHES TO START UP A GROUP OR CHANGE A GROUP WHILE UNDER THE CONTROL OF A DIFFERENT
# GROUP.  FOR EXAMPLE, CHANGE THE PHASE OF GROUP 3 WHILE THE PORTION OF THE PROGRAM IS UNDER GROUP 5.  ALL 2PHSCHNG
# CALLS ARE MADE IN THE FOLLOWING MANNER,
#                TC      2PHSCHNG
#                OCT     XXXXX
#                OCT     YYYYY
# WHERE OCT XXXXX MUST BE OF TYPE A AND OCT YYYYY MAY BE OF EITHER TYPE A OR TYPE B OR TYPE C.  THERE IS ONE
# DIFFERENCE --- NOTE: IF LONGBASE IS TO BE SET THIS INFORMATION IS GIVEN IN THE OCT YYYYY INFORMATION, IT WILL
# BE DISREGARDED IF GIVEN WITH THE OCT XXXXX INFORMATION.  A COUPLE OF EXAMPLES MAY HELP,
#        AD      TC      2PHACHNG        # SET TBASE3 AND IF A RESTART OCCURS START
#        AD+1    OCT     40083           # THE TWO ENTRIES IN 3.8 TABLE LOCATION
#        AD+2    OCT     05025           # THIS IS OF TYPE C, SET THE JOB TO BE
#        AD+3    OCT     18000           # TO BE LOCATION AD+4, WITH A PRIORITY 18,
#        AD+4                            # FOR GROUP 5 PHASE INFORMATION.

                SBANK=  PINSUPER

                COUNT*  $$/PHASE
2PHSCHNG        INHINT                  # THE ENTRY FOR A DOUBLE PHASE CHANGE
                NDX     Q
                CA      0
                INCR    Q
                TS      TEMPP2

                MASK    OCT7
                DOUBLE
                TS      TEMPG2

                CA      TEMPP2
                MASK    OCT17770        # NEED ONLY 1770, BUT WHY GET A NEW CONST.
                EXTEND
                MP      BIT12
                XCH     TEMPP2

                MASK    BIT15
                TS      TEMPSW2         # INDICATES WHETHER TO SET TBASE OR NOT

                INDEX   Q
                CA      0
                INCR    Q
                TS      TEMPSW

                TCF     PHASJUMP

PHASCHNG        INHINT                  # NORMAL PHASCHNG ENTRY POINT.
                INDEX   Q
                CA      0
                INCR    Q
PHSCHNGA        INHINT                  # FIRST OCTAL PARAMETER IN A.
```

```
                TS      TEMPSW
                CA      ONE
                TS      TEMPSW2
PHASJUMP        EXTEND
                DCA     ADRPCHN2        # OFF TO SWITCHED BANK
                DTCB

                EBANK=  LST1
ADRPCHN2        2CADR   PHSCHNG2

ONEORTWO        LXCH    TEMPBBCN
                LXCH    BBANK
                LXCH    TEMPBBCN

                MASK    OCT14000        # SEE WHAT KIND OF PHASE CHANGE IT IS
                CCS     A
                TCF     CHECKB          # IT IS OF TYPE `B'.

                CA      TEMPP
                MASK    BIT7
                CCS     A               # SHALL WE USE THE OLD PRIORITY
                TCF     GETPRIO         # NO GET A NEW PRIORITY (OR DELTA T)

OLDPRIO         NDX     TEMPG           # USE THE OLD PRIORITY (OR DELTA T)
                CA      PHSPRDT1 -2
                TS      TEMPPR

CON1            CA      TEMPP           # SEE IF A 2CADR IS GIVEN
                MASK    BIT8
                CCS     A
                TCF     GETNEWNM

                CA      Q
                TS      TEMPNM
                CA      BB
                EXTEND                  # PICK UP USER'S SUPERBANK
                ROR     SUPERBNK
                TS      TEMPBB

TOCON2          CA      CON2ADR         # BACK TO SWITCHED BANK
                LXCH    TEMPBBCN
                DTCB

CON2ADR         GENADR  CON2

GETPRIO         NDX     Q               # DON'T CARE IF DIRECT OR INDIRECT
                CA      0               # LEAVE THAT DECISION TO RESTARTS
                INCR    Q               # OBTAIN RETURN ADDRESS
```

```
                TCF     CON1 -1

GETNEWNM        EXTEND
                INDEX   Q
                DCA     0
                DXCH    TEMPNM
                CA      TWO
                ADS     Q               # OBTAIN RETURN ADDRESS

                TCF     TOCON2

OCT14000        EQUALS  PRIO14
TEMPG           EQUALS  ITEMP1
TEMPP           EQUALS  ITEMP2
TEMPNM          EQUALS  ITEMP3
TEMPBB          EQUALS  ITEMP4
TEMPSW          EQUALS  ITEMP5
TEMPSW2         EQUALS  ITEMP6
TEMPPR          EQUALS  RUPTREG1
TEMPG2          EQUALS  RUPTREG2
TEMPP2          EQUALS  RUPTREG3

TEMPBBCN        EQUALS  RUPTREG4
BB              EQUALS  BBANK

                BANK    14
                SETLOC  PHASETAB
                BANK

                EBANK=  PHSNAME1
                COUNT*  $$/PHASE
PHSCHNG2        LXCH    TEMPBBCN
                CA      TEMPSW
                MASK    OCT7
                DOUBLE
                TS      TEMPG

                CA      TEMPSW
                MASK    OCT17770
                EXTEND
                MP      BIT12
                TS      TEMPP

                CA      TEMPSW
                MASK    OCT60000
                XCH     TEMPSW
                MASK    OCT14000
                CCS     A
```

```
                TCF     ONEORTWO

                CA      TEMPP           # START STORING THE PHASE INFORMATION
                NDX     TEMPG
                TS      PHASE1 -2

BELOW1          CCS     TEMPSW2         # IS IT A PHASCHNG OR A 2PHSCHNG
                TCF     BELOW2          # IT'S A PHASCHNG

                TCF     +1              # IT'S A 2PHSCHNG
                CS      TEMPP2
                LXCH    TEMPP2
                NDX     TEMPG2
                DXCH    -PHASE1 -2

                CCS     TEMPSW2
                NOOP                    # CAN'T GET HERE
                TCF     BELOW2

                CS      TIME1
                NDX     TEMPG2
                TS      TBASE1 -2

BELOW2          CCS     TEMPSW          # SEE IF WE SHOULD SET TBASE OR LONGBASE
                TCF     BELOW3          # SET LONGBASE ONLY
                TCF     BELOW4          # SET NEITHER

                CS      TIME1           # SET TBASE TO BEGIN WITH
                NDX     TEMPG
                TS      TBASE1 -2

                CA      TEMPSW          # SHALL WE NOW SET LONGBASE
                AD      BIT14COM
                CCS     A
                NOOP                    # ***** CAN'T GET HERE *****
BIT14COM        OCT     17777           # ***** CAN'T GET HERE *****
                TCF     BELOW4          # NO WE NEED ONLY SET TBASE

BELOW3          EXTEND                  # SET LONGBASE
                DCA     TIME2
                DXCH    LONGBASE

BELOW4          CS      TEMPP           # AND STORE THE FINAL PART OF THE PHASE
                NDX     TEMPG
                TS      -PHASE1 -2

                CA      Q
                LXCH    TEMPBBCN
                RELINT
                DTCB
```

```
CON2            LXCH    TEMPBBCN

                CA      TEMPP
                NDX     TEMPG
                TS      PHASE1 -2

                CA      TEMPPR
                NDX     TEMPG
                TS      PHSPRDT1 -2

                EXTEND
                DCA     TEMPNM
                NDX     TEMPG
                DXCH    PHSNAME1 -2

                TCF     BELOW1

                BLOCK   03
                SETLOC  FFTAG6
                BANK

                COUNT*  $$/PHASE
CHECKB          MASK    BIT12           # SINCE THIS IS OF TYPE B, THIS BIT WOULD
                CCS     A               # BE HERE IF WE ARE TO GET A NEW PRIORITY
                TCF     GETPRIO         # IT IS, SO GET NEW PRIORITY

                TCF     OLDPRIO         # IT ISN'T, USE THE OLD PRIORITY.
```

```
                BANK    01
                SETLOC  RESTART
                BANK

                EBANK=  PHSNAME1        # GOPROG MUST SWITCH TO THIS EBANK

                COUNT*  $$/RSROU
RESTARTS        CA      MPAC +5         # GET GROUP NUMBER -1
                DOUBLE                  # SAVE FOR INDEXING
                TS      TEMP2G

                CA      PHS2CADR        # SET UP EXIT IN CASE IT IS AN EVEN
                TS      TEMPSWCH        # TABLE PHASE

                CA      RTRNCADR        # TO SAVE TIME ASSUME IT WILL GET NEXT
                TS      GOLOC +2        # GROUP AFTER THIS

                CA      TEMPPHS
                MASK    OCT1400
                CCS     A               # IS IT A VARIABLE OR TABLE RESTART
                TCF     ITSAVAR         # IT:S A VARIABLE RESTART

GETPART2        CCS     TEMPPHS         # IS IT AN X.1 RESTART
                CCS     A
                TCF     ITSATBL         # NO, ITS A TABLE RESTART

                CA      PRIO14          # IT IS AN X.1 RESTART, THEREFORE START
                TC      FINDVAC         # THE DISPLAY RESTART JOB
                EBANK=  LST1
                2CADR   INITDSP

                TC      RTRNCADR        # FINISHED WITH THIS GROUP, GET NEXT ONE

ITSAVAR         MASK    OCT1400         # IS IT TYPE B ?
                CCS     A
                TCF     ITSLIKEB        # YES,IT IS TYPE B

                EXTEND                  # STORE THE JOB (OR TASK) 2CADR FOR EXIT
                NDX     TEMP2G
                DCA     PHSNAME1
                DXCH    GOLOC

                CA      TEMPPHS         # SEE IF THIS IS A JOB, TASK, OR A LONGCAL
                MASK    OCT7
                AD      MINUS2
                CCS     A
                TCF     ITSLNGCL        # ITS A LONGCALL

RTRNCADR        TC      SWRETURN        # CANT GET HERE
```

```
                TCF     ITSAWAIT

                TCF     ITSAJOB         # ITS A JOB

ITSAWAIT        CA      WTLTCADR        # SET UP WAITLIST CALL
                TS      GOLOC -1

                NDX     TEMP2G          # DIRECTLY STORED
                CA      PHSPRDT1
TIMETEST        CCS     A               # IS IT AN IMMEDIATE RESTART
                INCR    A               # NO.
                TCF     FINDTIME        # FIND OUT WHEN IT SHOULD BEGIN

                TCF     ITSINDIR        # STORED INDIRECTLY

                TCF     IMEDIATE        # IT WANTS AN IMMEDIATE RESTART

# ***** THIS MUST BE IN FIXED FIXED *****

                BLOCK   02
                SETLOC  FFTAG2
                BANK

                COUNT*  $$/RSROU
ITSINDIR        LXCH    GOLOC +1        # GET THE CORRECT E BANK IN CASE THIS IS
                LXCH    BB              # SWITCHED ERRASIBLE

                NDX     A               # GET THE TIME INDIRECTLY
                CA      1

                LXCH    BB              # RESTORE THE BB AND GOLOC
                LXCH    GOLOC +1

                TCF     FINDTIME        # FIND OUT WHEN IT SHOULD BEGIN

# ***** YOU MAY RETURN TO SWITCHED FIXED *****

                BANK    01
                SETLOC  RESTART
                BANK

                COUNT*  $$/RSROU
FINDTIME        COM                     # MAKE NEGITIVE SINCE IT WILL BE SUBTRACTD
                TS      L               # AND SAVE
                NDX     TEMP2G
                CS      TBASE1
                EXTEND
                SU      TIME1
                CCS     A
                COM
```

```
                AD      OCT37776
                AD      ONE
                AD      L
                CCS     A
                CA      ZERO
                TCF     +2
                TCF     +1
IMEDIATE        AD      ONE
                TC      GOLOC -1
ITSLIKEB        CA      RTRNCADR        # TYPE B,               SO STORE RETURN IN
                TS      TEMPSWCH        # TEMPSWCH IN CASE OF AN EVEN PHASE

                CA      PRT2CADR        # SET UP EXIT TO GET TABLE PART OF THIS
                TS      GOLOC +2        # VARIABLE TYPE OF PHASE

                CA      TEMPPHS         # MAKE THE PHASE LOOK RIGHT FOR THE TABLE
                MASK    OCT177          # PART OF THIS VARIABLE PHASE
                TS      TEMPPHS

                EXTEND
                NDX     TEMP2G          # OBTAIN THE JOB:S 2CADR
                DCA     PHSNAME1
                DXCH    GOLOC

ITSAJOB         NDX     TEMP2G          # NOW ADD THE PRIORITY AND LET:S GO
                CA      PHSPRDT1
CHKNOVAC        TS      GOLOC -1        # SAVE PRIO UNTIL WE SEE IF ITS
                EXTEND                  # A FINDVAC OR A NOVAC
                BZMF    ITSNOVAC

                CAF     FVACCADR        # POSITIVE, SET UP FINDVAC CALL.
                XCH     GOLOC -1        # PICK UP PRIO,
                TC      GOLOC -1        # AND GO

ITSNOVAC        CAF     NOVACADR        # NEGATIVE,
                XCH     GOLOC -1        # SET UP NOVAC CALL,
                COM                     # CORRECT PRIO,
                TC      GOLOC -1        # AND GO

ITSATBL         TS      CYR             # FIND OUT IF THE PHASE IS ODD OR EVEN
                CCS     CYR
                TCF     +1              # IT:S EVEN
                TCF     ITSEVEN

                CA      RTRNCADR        # IN CASE THIS IS THE SECOND PART OF A
                TS      GOLOC +2        # TYPE B RESTART, WE NEED PROPER EXIT

                CA      TEMPPHS         # SET UP POINTER FOR FINDING OUR PLACE IN
                TS      SR              # THE RESTART TABLES
                AD      SR
```

```
                NDX     TEMP2G
                AD      SIZETAB +1
                TS      POINTER

CONTBL2         EXTEND                  # FIND OUT WHAT:S IN THE TABLE
                NDX     POINTER
                DCA     CADRTAB         # GET THE 2CADR

                LXCH    GOLOC +1        # STORE THE BB INFORMATION

                CCS     A               # IS IT A JOB OR IS IT  TIMED
                INCR    A               # POSITIVE. MUST BE A JOB
                TCF     ITSAJOB2

                INCR    A               # MUST BE EITHER A WAITLIST OR LONGCALL
                TS      GOLOC           # LET-S STORE THE CORRECT CADR

                CA      WTLTCADR        # SET UP OUR EXIT TO WAITLIST
                TS      GOLOC -1

                CA      GOLOC +1        # NOW FIND OUT IF IT IS A WAITLIST CALL
                MASK    BIT10           # THIS SHOULD BE ONE IF WE HAVE -BB
                CCS     A               # FOR THAT MATTER SO SHOULD BE BITS 9,8,7,
                                        # 6,5, AND LAST BUT NOT LEAST (PERHAPS NOT
                                        # IN IMPORTANCE ANYWAY. BIT 4
                TCF     ITSWTLST        # IT IS A WAITLIST CALL

                NDX     POINTER         # OBTAIN THE ORIGINAL DELTA T
                CA      PRDTTAB         # ADDRESS FOR THIS LONGCALL

                TCF     ITSLGCL1        # NOW GO GET THE DELTA TIME

# ***** THIS MUST BE IN FIXED FIXED *****

                BLOCK   02
                SETLOC  FFTAG2
                BANK

                COUNT*  $$/RSROU
ITSLGCL1        LXCH    GOLOC +1        # OBTAIN THE CORRECT E BANK
                LXCH    BB
                LXCH    GOLOC +1        # AND PRESERVE OUR E AND F BANKS

                EXTEND                  # GET THE DELTA TIME
                NDX     A
                DCA     0

                LXCH    GOLOC +1        # RESTORE OUR E AND F BANK
                LXCH    BB              # RESTORE THE TASKS E AND F BANKS
                LXCH    GOLOC +1        # AND PRESERVE OUR L
```

```
                TCF     ITSLGCL2        # NOT LET:S PROCESS THIS LONGCALL

# ***** YOU MAY RETURN TO  SWITCHED FIXED *****

                BANK    01
                SETLOC  RESTART
                BANK

                COUNT*  $$/RSROU
ITSLGCL2        DXCH    LONGTIME

                EXTEND                  # CALCULATE TIME LEFT
                DCS     TIME2
                DAS     LONGTIME
                EXTEND
                DCA     LONGBASE
                DAS     LONGTIME

                CCS     LONGTIME        # FIND OUT HOW THIS SHOULD BE RESTARTED
                TCF     LONGCLCL
                TCF     +2
                TCF     IMEDIATE -3
                CCS     LONGTIME +1
                TCF     LONGCLCL
                NOOP                    # CAN:T GET HERE    *********
                TCF     IMEDIATE -3
                TCF     IMEDIATE

LONGCLCL        CA      LGCLCADR        # WE WILL GO TO LONGCALL
                TS      GOLOC -1

                EXTEND                  # PREPARE OUR ENTRY TO LONGCALL
                DCA     LONGTIME
                TC      GOLOC -1

ITSLNGCL        CA      WTLTCADR        # ASSUME IT WILL GO TO WAITLIST
                TS      GOLOC -1

                NDX     TEMP2G
                CS      PHSPRDT1        # GET THE DELTA T ADDRESS

                TCF     ITSLGCL1        # NOW GET THE DELTA TIME

ITSWTLST        CS      GOLOC +1        # CORRECT THE BBCON INFORMATION
                TS      GOLOC +1

                NDX     POINTER         # GET THE DT AND FIND OUT IF IT WAS STORED
                CA      PRDTTAB         # DIRECTLY OR INDIRECTLY

                TCF     TIMETEST        # FIND OUT HOW THE TIME IS STORED
```

```
ITSAJOB2        XCH     GOLOC           # STORE THE CADR

                NDX     POINTER         # ADD THE PRIORITY AND LET:S GO
                CA      PRDTTAB

                TCF     CHKNOVAC

ITSEVEN         CA      TEMPSWCH        # SET UP FOR EITHER THE SECOND PART OF THE
                TS      GOLOC +2        # TABLE, OR A RETURN FOR THE NEXT GROUP

                NDX     TEMP2G          # SET UP POINTER FOR OUR LOCATION WITHIN
                CA      SIZETAB         # THE TABLE
                AD      TEMPPHS         # THIS MAY LOOK BAD BUT LET:S SEE YOU DO
                AD      TEMPPHS         # BETTER IN TIME OR NUMBERR OF LOCATIONS
                AD      TEMPPHS
                TS      POINTER

                TCF     CONTBL2         # NOW PROCESS WHAT IS IN THE TABLE

PHSPART2        CA      THREE           # SET THE POINTER FOR THE SECOND HALF OF
                ADS     POINTER         # THE TABLE

                CA      RTRNCADR        # THIS WILL BE OUR LAST TIME THROUGH THE
                TS      GOLOC +2        # EVEN TABLE , SO AFTER IT GET THE NEXT
                                        # GROUP
                TCF     CONTBL2         # SO LET:S GET THE SECOND ENTRY IN THE TBL

TEMPPHS         EQUALS  MPAC
TEMP2G          EQUALS  MPAC +1
POINTER         EQUALS  MPAC +2
TEMPSWCH        EQUALS  MPAC +3
GOLOC           EQUALS  VAC5 +20D
MINUS2          EQUALS  NEG2
OCT177          EQUALS  LOW7

PHS2CADR        GENADR  PHSPART2
PRT2CADR        GENADR  GETPART2
LGCLCADR        GENADR  LONGCALL
FVACCADR        GENADR  FINDVAC
WTLTCADR        GENADR  WAITLIST
NOVACADR        GENADR  NOVAC
```

```
                BLOCK   02
                SETLOC  FFTAG3
                BANK

                EBANK=  COMMAND

# FIXED-FIXED ROUTINES

                COUNT*  $$/IMODE
ZEROICDU        CAF     ZERO            # ZERO ICDU COUNTERS.
                TS      CDUX
                TS      CDUY
                TS      CDUZ
                TC      Q

SPSCODE         =       BIT9
```

```
# IMU ZEROING ROUTINES

                BANK     11
                SETLOC   MODESW
                BANK

                COUNT*   $$/IMODE
IMUZERO         INHINT                  # ROUTINE TO ZERO ICDUS.
                CS       DSPTAB +11D    # DON'T ZERO CDUS IF IMU IN GIMBAL LOCK AND
                MASK     BITS4&6        # COARSE ALIGN (GIMBAL RUNAWAY PROTECTION)
                CCS      A
                TCF      IMUZEROA

                TC       ALARM          # IF SO.
                OCT      00206

                TCF      CAGETSTJ +4    # IMMEDIATE FAILURE.

IMUZEROA        TC       CAGETSTJ

                CS       IMODES33       # DISABLE DAP AUTO AND HOLD MODES
                MASK     SUPER011       #        BIT5 FOR GROUND
                ADS      IMODES33

                CS       IMODES30       # INHIBIT ICDUFAIL AND IMUFAIL (IN CASE WE
                MASK     BITS3&4        # JUST CAME OUT OF COARSE ALIGN).
                ADS      IMODES30

                CS       BITS4&6        # SEND ZERO ENCODE WITH COARSE AND ERROR
                EXTEND                  # COUNTER DISABLED.
                WAND     CHAN12

                TC       NOATTOFF       # TURN OFF NO ATT LAMP.

                CAF      BIT5
                EXTEND
                WOR      CHAN12

                TC       ZEROICDU
                CAF      BIT6           # WAIT 320 MS TO GIVE AGS ADEQUATE TIME TO
                TC       WAITLIST       # RECEIVE ITS PULSE TRAIN.
                EBANK=   CDUIND
                2CADR    IMUZERO2

                CS       IMODES30       # SEE IF IMU OPERATING AND ALARM IF NOT.
                MASK     BIT9
                CCS      A
                TCF      MODEEXIT
```

```
                TC      ALARM
                OCT     210

MODEEXIT        RELINT                  # GENERAL MODE-SWITCHING EXIT.
                TCF     SWRETURN

IMUZERO2        TC      CAGETEST
                TC      ZEROICDU        # ZERO CDUX, CDUY, CDUZ

                CS      BIT5            # REMOVE ZERO DISCRETE.
                EXTEND
                WAND    CHAN12

                CAF     BIT11           # WAIT 10 SECS FOR CTRS TO FIND GIMBALS
                TC      VARDELAY

IMUZERO3        TC      CAGETEST
                CS      BITS3&4         # REMOVE IMUFAIL AND ICDUFAIL INHIBIT.
                MASK    IMODES30
                TS      IMODES30

                CS      SUPER011        # ENABLE DAP AUTO AND HOLD MODES
                MASK    IMODES33        #       BIT5 FOR GROUND
                TS      IMODES33

                TC      IBNKCALL        # SET ISS WARNING IF EITHER OF ABOVE ARE
                CADR    SETISSW         # PRESENT.

                TCF     ENDIMU
```

```
# IMU COARSE ALIGN MODE.

IMUCOARS        INHINT
                TC      CAGETSTJ
                TC      SETCOARS

                CAF     SIX
                TC      WAITLIST
                EBANK=  CDUIND
                2CADR   COARS

                TCF     MODEEXIT

COARS           TC      CAGETEST
                CAF     BIT6            # ENABLE ALL THREE ISS CDU ERROR COUNTERS
                EXTEND
                WOR     CHAN12

                CAF     TWO             # SET CDU INDICATOR
COARS1          TS      CDUIND

                INDEX   CDUIND          # COMPUTE THETAD -- THETAA IN 1'S
                CA      THETAD          #       COMPLEMENT FORM
                EXTEND
                INDEX   CDUIND
                MSU     CDUX
                EXTEND
                MP      BIT13           # SHIFT RIGHT 2
                XCH     L               # ROUND
                DOUBLE
                TS      ITEMP1
                TCF     +2
                ADS     L

                INDEX   CDUIND          # DIFFERENCE TO BE COMPUTED
                LXCH    COMMAND
                CCS     CDUIND
                TC      COARS1

                CAF     TWO             # MINIMUM OF 4 MS WAIT
                TC      VARDELAY
```

```
COARS2          TC      CAGETEST        # DON'T CONTINUE IF CAGED.
                TS      ITEMP1          # SET TO +0.
                CAF     TWO             # SET CDU INDICATOR.
 +3             TS      CDUIND

                INDEX   CDUIND
                CCS     COMMAND         # NUMBER OF PULSES REQUIRED
                TC      COMPOS          # GREATER THAN MAX ALLOWED
                TC      NEXTCDU +1
                TC      COMNEG
                TC      NEXTCDU +1

COMPOS          AD      -COMMAX         # COMMAX = MAX NUMBER OF PULSES ALLOWED
                EXTEND                  #       MINUS ONE
                BZMF    COMZERO
                INDEX   CDUIND
                TS      COMMAND         # REDUCE COMMAND BY MAX NUMBER OF PULSES
                CS      -COMMAX-        #       ALLOWED

NEXTCDU         INCR    ITEMP1
                AD      NEGO
                INDEX   CDUIND
                TS      CDUXCMD         # SET UP COMMAND REGISTER.

                CCS     CDUIND
                TC      COARS2 +3

                CCS     ITEMP1          # SEE IF ANY PULSES TO GO OUT.
                TCF     SENDPULS

                TC      FIXDELAY        # WAIT FOR GIMBALS TO SETTLE.
                DEC     150

                CAF     TWO             # AT END OF COMMAND, CHECK TO SEE THAT
CHKCORS         TS      ITEMP1          # GIMBALS ARE WITHIN 2 DEGREES OF THETAD.
                INDEX   A
                CA      CDUX
                EXTEND
                INDEX   ITEMP1
                MSU     THETAD
                CCS     A
                TCF     COARSERR
                TCF     CORSCHK2
                TCF     COARSERR
```

```
CORSCHK2        CCS     ITEMP1
                TCF     CHKCORS
                TCF     ENDIMU          # END OF COARSE ALIGNMENT

COARSERR        AD      COARSTOL        # 2 DEGREES.
                EXTEND
                BZMF    CORSCHK2

                TC      ALARM           # COARSE ALIGN ERROR.
                OCT     211

                TCF     IMUBAD

COARSTOL        DEC     -.01111         # 2 DEGREES SCALED AT HALF-REVOLUTIONS

COMNEG          AD      -COMMAX
                EXTEND
                BZMF    COMZERO
                COM
                INDEX   CDUIND
                TS      COMMAND
                CA      -COMMAX-
                TC      NEXTCDU

COMZERO         CAF     ZERO
                INDEX   CDUIND
                XCH     COMMAND
                TC      NEXTCDU

SENDPULS        CAF     13,14,15
                EXTEND
                WOR     CHAN14
                CAF     600MS
                TCF     COARS2 -1       # THEN TO VARDELAY

CA+ECE          CAF     BIT6            # ENABLE ALL THREE ISS CDU ERROR COUNTERS
                EXTEND
                WOR     CHAN12
                TC      TASKOVER
```

```
SETCOARS        CAF     BIT4            # BYPASS IF ALREADY IN COARSE ALIGN
                EXTEND
                RAND    CHAN12
                CCS     A
                TC      Q

                CS      BIT6            # CLEAR ISS ERROR COUNTERS
                EXTEND
                WAND    CHAN12

                CS      BIT10           # KNOCK DOWN GYRO ACTIVITY
                EXTEND
                WAND    CHAN14
                CS      ZERO
                TS      GYROCMD

                CAF     BIT4            # PUT ISS IN COARSE ALIGN
                EXTEND
                WOR     CHAN12

                CS      DSPTAB +11D     # TURN ON NO ATT LAMP
                MASK    OCT40010
                ADS     DSPTAB +11D

                CS      IMODES33        # DISABLE DAP AUTO AND HOLD MODES
                MASK    BIT6
                ADS     IMODES33

                CS      IMODES30        # DISABLE IMUFAIL
                MASK    BIT4
                ADS     IMODES30

RNDREFDR        CS      TRACKBIT        # CLEAR TRACK FLAG
                MASK    FLAGWRD1
                TS      FLAGWRD1

                CS      DRFTBIT         # CLEAR DRIFT FLAG
                MASK    FLAGWRD2
                TS      FLAGWRD2

                CS      REFSMBIT        # CLEAR REFSMMAT FLAG
                MASK    FLAGWRD3
                TS      FLAGWRD3

                TC      Q

OCT40010        OCT     40010
```

```
# IMU FINE ALIGN MODE SWITCH.

IMUFINE         INHINT
                TC      CAGETSTJ        # SEE IF IMU BEING CAGED.

                CS      BITS4-5         # RESET ZERO AND COARSE
                EXTEND
                WAND    CHAN12

                CS      BIT6            # INSURE DAP AUTO AND HOLD MODES ENABLED
                MASK    IMODES33
                TS      IMODES33

                TC      NOATTOFF

                CAF     BIT10           # IMU FAIL WAS INHIBITED DURING THE
                TC      WAITLIST        # PRESUMABLY PRECEDING COARSE ALIGN.  LEAVE
                EBANK=  CDUIND
                2CADR   IFAILOK         # IT ON FOR THE FIRST 5 SECS OF FINE ALIGN

                CAF     2SECS
                TC      WAITLIST
                EBANK=  CDUIND
                2CADR   IMUFINED

                TCF     MODEEXIT

IMUFINED        TC      CAGETEST        # SEE THAT NO ONE HAS CAGED THE IMU.
                TCF     ENDIMU
```

```
IFAILOK         TC      CAGETSTQ        # ENABLE IMU FAIL UNLESS IMU BEING CAGED.
                TCF     TASKOVER        # IT IS.

                CAF     BIT4            # DON'T RESET IMU FAIL INHIBIT IF SOMEONE
                EXTEND                  # HAS GONE INTO COARSE ALIGN.
                RAND    CHAN12
                CCS     A
                TCF     TASKOVER

                CS      IMODES30        # RESET IMUFAIL.
                MASK    BIT13
                ADS     IMODES30
                CS      BIT4
PFAILOK2        MASK    IMODES30
                TS      IMODES30
                TC      IBNKCALL        # THE ISS WARNING LIGHT MAY COME ON NOW
                CADR    SETISSW         # THAT THE INHIBIT WAS BEEN REMOVED.
                TCF     TASKOVER

PFAILOK         TC      CAGETSTQ        # ENABLE PIP FAIL PROG ALARM.
                TCF     TASKOVER

                CS      IMODES30        # RESET IMU AND PIPA FAIL BITS.
                MASK    BIT10
                ADS     IMODES30

                CS      IMODES33
                MASK    BIT13
                ADS     IMODES33

                CS      BIT5
                TCF     PFAILOK2

NOATTOFF        CS      OCT40010        # SUBROUTINE TO TURN OFF NO ATT LAMP.
                MASK    DSPTAB +11D
                AD      BIT15
                TS      DSPTAB +11D
                TC      Q
```

```
# ROUTINES TO INITIATE AND TERMINATE PROGRAM USE OF THE PIPAS. NO IMUSTALL REQUIRED IN EITHER CASE.

PIPUSE          CS      ZERO
                TS      PIPAX
                TS      PIPAY
                TS      PIPAZ

PIPUSE1         TC      CAGETSTQ        # DO NOT ENABLE PIPA FAIL IF IMU IS CAGED
                TCF     SWRETURN

                INHINT
                CS      BIT1            # IF PIPA FAILS FROM NOW ON (UNTIL
                MASK    IMODES30        # PIPFREE), LIGHT ISS WARNING.
                TS      IMODES30

PIPFREE2        TC      IBNKCALL        # ISS WARNING MIGHT COME ON NOW.
                CADR    SETISSW         # (OR GO OFF ON PIPFREE).

                TCF     MODEEXIT

PIPFREE         INHINT                  # PROGRAM DONE WITH PIPAS.  DON'T LIGHT
                CS      IMODES30        # ISS WARNING.
                MASK    BIT1
                ADS     IMODES30

                MASK    BIT10           # IF PIP FAIL ON, DO PROG ALSRM AND RESET
                CCS     A               # ISS WARNING.
                TCF     MODEEXIT

                TC      ALARM
                OCT     212

                INHINT

                TCF     PIPFREE2
```

```
#          THE FOLLOWING ROUTINE TORQUES THE IRIGS ACCORDING TO DOUBLE PRECISION INPUTS IN THE SIX REGISTERS
# BEGINNING AT THE ECADR ARRIVING IN A. THE MINIMUM SIZE OF ANY PULSE TRAIN IS 16 PULSES (.25 CDU COUNTS). THE
# UNSENT PORTION OF THE COMMAND IS LEFT INTACT IN THE INPUT COMMAND REGISTERS.

                EBANK=  1400            # VARIABLE, ACTUALLY.

IMUPULSE        TS      MPAC +5         # SAVE ARRIVING ECADR.
                TC      CAGETSTJ        # DON'T PROCEED IF IMU BEING CAGED.

                CCS     LGYRO           # SEE IF GYROS BUSY.
                TC      GYROBUSY        # SLEEP.

                TS      MPAC +2
                CAF     BIT6            # ENABLE THE POWER SUPPLY.
                EXTEND
                WOR     CHAN14

                CAF     FOUR
GWAKE2          TC      WAITLIST        # (IF A JOB WAS PUT TO SLEEP, THE POWER
                EBANK=  CDUIND          # SUPPLY IS LEFT ON BY THE WAKING JOB).
                2CADR   STRTGYRO

                CA      MPAC +5         # SET UP EBANK, SAVING CALLER'S EBANK FOR
                XCH     EBANK           # RESTORATION ON RETURN.
                XCH     MPAC +5
                TS      LGYRO           # RESERVES GYROS.
                MASK    LOW8
                TS      ITEMP1

                CAF     TWO             # FORCE SIGN AGREEMENT ON INPUTS.
GYROAGRE        TS      MPAC +3
                DOUBLE
                AD      ITEMP1
                TS      MPAC +4
                EXTEND
                INDEX   A
                DCA     1400
                DXCH    MPAC
                TC      TPAGREE
                DXCH    MPAC
                INDEX   MPAC +4
                DXCH    1400

                CCS     MPAC +3
                TCF     GYROAGRE

                CA      MPAC +5         # RESTORE CALLER'S EBANK.
                TS      EBANK
                TCF     MODEEXIT
```

```
# ROUTINES TO ALLOW TORQUING ONLY ONE JOB AT A TIME.

GYROBUSY        EXTEND                  # SAVE RETURN 2FCADR.
                DCA     BUF2
                DXCH    MPAC
REGSLEEP        CAF     LGWAKE
                TCF     JOBSLEEP

GWAKE           CCS     LGYRO           # WHEN AWAKENED, SEE IF GYROS STILL BUSY.
                TCF     REGSLEEP        # IF SO, SLEEP SOME MORE.

                TS      MPAC +2
                EXTEND
                DCA     MPAC
                DXCH    BUF2            # RESTORE SWRETURN INFO.
                CAF     ONE
                TCF     GWAKE2

LGWAKE          CADR    GWAKE
```

```
# GYRO-TORQUING WAITLIST TASKS.

STRTGYRO        CS      GDESELCT        # DE-SELECT LAST GYRO.
                EXTEND
                WAND    CHAN14

                TC      CAGETEST

STRTGYR2        CA      LGYRO           # JUMP ON PHASE COUNTER IN BITS 13-14.
                EXTEND
                MP      BIT4
                INDEX   A
                TCF     +1
                TC      GSELECT         # =0.  DO Y GYRO.
                OCT     00202

                TC      GSELECT         # =1.  DO Z GYRO.
                OCT     00302

                TC      GSELECT -2      # =2.  DO X GYRO.
                OCT     00100

                CAF     ZERO            # =3.  DONE
                TS      LGYRO
                CAF     LGWAKE          # WAKE A POSSIBLE SLEEPING JOB.
                TC      JOBWAKE

NORESET         TCF     IMUFINED        # DO NOT RESET POWER SUPPLY
```

```
 -2             CS      FOUR            # SPECIAL ENTRY TO REGRESS LGYRO FOR X.
                ADS     LGYRO

GSELECT         INDEX   Q               # SELECT GYRO.
                CAF     0               # PACKED WORD CONTAINS GYRO SELECT BITS
                TS      ITEMP4          # AND INCREMENT TO LGYRO.
                MASK    SEVEN
                AD      BIT13
                ADS     LGYRO
                TS      EBANK
                MASK    LOW8
                TS      ITEMP1

                CS      SEVEN
                MASK    ITEMP4
                TS      ITEMP4

                EXTEND                  # MOVE DP COMMAND TO RUPTREGS FOR TESTING.
                INDEX   ITEMP1
                DCA     1400
                DXCH    RUPTREG1

                CCS     RUPTREG1
                TCF     MAJ+
                TCF     +2
                TCF     MAJ-

                CCS     RUPTREG2
                TCF     MIN+
                TCF     STRTGYR2
                TCF     MIN-
                TCF     STRTGYR2
```

```
MIN+            AD      -GYROMIN        # SMALL POSITIVE COMMAND.  SEE IF AT LEAST
                EXTEND                  # 16 GYRO PULSES.
                BZMF    STRTGYR2

MAJ+            EXTEND                  # DEFINITE POSITIVE OUTPUT.
                DCA     GYROFRAC
                DAS     RUPTREG1

                CA      ITEMP4          # SELECT POSITIVE TORQUING FOR THIS GYRO.
                EXTEND
                WOR     CHAN14

                CAF     LOW7            # LEAVE NUMBER OF POSSIBLE 8192 AUGMENTS
                MASK    RUPTREG2        # TO INITIAL COMMAND IN MAJOR PART OF LONG
                XCH     RUPTREG2        # TERM STORAGE AND TRUNCATED FRACTION
GMERGE          EXTEND                  # IN MINOR PART.  THE MAJOR PART WILL BE
                MP      BIT8            # COUNTED DOWN TO ZERO IN THE COURSE OF
                TS      ITEMP2          # PUTTING OUT THE ENTIRE COMMAND.
                CA      RUPTREG1
                EXTEND
                MP      BIT9
                TS      RUPTREG1
                CA      L
                EXTEND
                MP      BIT14
                ADS     ITEMP2          # INITIAL COMMAND.

                EXTEND                  # SEE IF MORE THAN ONE PULSE TRAIN NEEDED
                DCA     RUPTREG1        # (MORE THAN 16383 PULSES).
                AD      MINUS1
                CCS     A
                TCF     LONGGYRO
-GYROMIN        OCT     -176            # MAY BE ADJUSTED TO SPECIFY MINIMUM CMD
                TCF     +4

                CAF     BIT14
                ADS     ITEMP2
                CAF     ZERO

 +4             INDEX   ITEMP1
                DXCH    1400
```

```
                CA      ITEMP2          # ENTIRE COMMAND.
LASTSEG         TS      GYROCMD
                EXTEND
                MP      BIT10           # WAITLIST DT
                AD      THREE           # TRUNCATION AND PHASE UNCERTAINTIES.
                TC      WAITLIST
                EBANK=  CDUIND
                2CADR   STRTGYRO

GYROEXIT        CAF     BIT10
                EXTEND
                WOR     CHAN14
                TCF     TASKOVER

LONGGYRO        INDEX   ITEMP1
                DXCH    1400            # INITIAL COMMAND OUT PLUS N AUGMENTS OF
                CAF     BIT14           # 8192.  INITIAL COMMAND IS AT LEAST 8192.
                AD      ITEMP2
                TS      GYROCMD

AUG3            EXTEND                  # GET WAITLIST DT TO TIME WHEN TRAIN IS
                MP      BIT10           # ALMOST OUT.
                AD      NEG3
                TC      WAITLIST
                EBANK=  CDUIND
                2CADR   8192AUG

                TCF     GYROEXIT

8192AUG         TC      CAGETEST

                CAF     BIT4
                EXTEND
                RAND    CHAN12
                CCS     A
                TCF     IMUBAD
                CA      LGYRO           # ADD 8192 PULSES TO GYROCMD
                TS      EBANK
                MASK    LOW8
                TS      ITEMP1

                INDEX   ITEMP1          # SEE IF THIS IS THE LAST AUG.
                CCS     1400
                TCF     AUG2            # MORE TO COME.

                CAF     BIT14
                ADS     GYROCMD
                TCF     LASTSEG +1
```

```
AUG2            INDEX   ITEMP1
                TS      1400
                CAF     BIT14
                ADS     GYROCMD
                TCF     AUG3            # COMPUTE DT.
```

```
MIN-            AD      -GYROMIN        # POSSIBLE NEGATIVE OUTPUT.
                EXTEND
                BZMF    STRTGYR2

MAJ-            EXTEND                  # DEFINITE NEGATIVE OUTPUT.
                DCS     GYROFRAC
                DAS     RUPTREG1

                CA      ITEMP4          # SELECT NEGATIVE TORQUING FOR THIS GYRO.
                AD      BIT9
                EXTEND
                WOR     CHAN14

                CS      RUPTREG1        # SET UP RUPTREGS TO FALL INTO GMERGE.
                TS      RUPTREG1        # ALL NUMBERS PUT INTO GYROCMD ARE
                CS      RUPTREG2        # POSITIVE -- BIT9 OF CHAN 14 DETERMINES
                MASK    LOW7            # THE SIGN OF THE COMMAND.
                COM
                XCH     RUPTREG2
                COM
                TCF     GMERGE

GDESELCT        OCT     1700            # TURN OFF SELECT AND ACTIVITY BITS.

GYROFRAC        2DEC    .215 B -21
```

```
# IMU MODE SWITCHING ROUTINES COME HERE WHEN ACTION COMPLETE.

ENDIMU          EXTEND                          # MODE IS BAD IF CAGE HAS OCCURRED OR IF
                READ    DSALMOUT                # ISS WARNING IS ON.
                MASK    BIT1
                CCS     A
                TCF     IMUBAD

IMUGOOD         TCF     GOODEND                 # WITH C(A) = 0.

IMUBAD          CAF     ZERO
                TCF     BADEND

CAGETEST        CAF     BIT6                    # SUBROUTINE TO TERMINATE IMU MODE
                MASK    IMODES30                # SWITCH IF IMU HAS BEEN CAGED.
                CCS     A
                TCF     IMUBAD                  # DIRECTLY.
                TC      Q                       # WITH C(A) = +0.

CAGETSTQ        CS      IMODES30                # SKIP IF IMU NOT BEING CAGED.
                MASK    BIT6
                CCS     A
                INCR    Q
                TC      Q

CAGETSTJ        CS      IMODES30                # IF DURING MODE SWITCH INITIALIZATION
                MASK    BIT6                    # IT IS FOUND THAT THE IMU IS BEING CAGED,
                CCS     A                       # SET IMUCADR TO -0 TO INDICATE OPERATION
                TC      Q                       # COMPLETE BUT FAILED.  RETURN IMMEDIATELY

                CS      ZERO                    # TO SWRETURN.
                TS      IMUCADR
                TCF     MODEEXIT
```

```
#          GENERALIZED MODE SWITCHING TERMINATION. ENTER AT GOODEND FOR SUCCESSFUL COMPLETION OF AN I/O OPERATION
# OR AT BADEND FOR A N UNSUCCESSFUL ONE. C(A) OR ARRIVAL =0 FOR IMU, 1 FOR OPTICS.

BADEND          TS      RUPTREG2        # DEVICE INDEX.
                CS      ZERO            # FOR FAILURE.
                TCF     GOODEND +2

GOODEND         TS      RUPTREG2
                CS      ONE             # FOR SUCCESS.

                TS      RUPTREG3
                INDEX   RUPTREG2        # SEE IF USING PROGRAM ASLEEP.
                CCS     MODECADR
                TCF     +4              # YES -- WAKE IT UP.
                TCF     ENDMODE         # IF 0, PROGRAM NOT IN YET.

                EXTEND
                BZF     ENDMODE +1      # BZF = TCF IF MODECADR = -0.

                CAF     ZERO            # WAKE SLEEPING PROGRAM.
                INDEX   RUPTREG2
                XCH     MODECADR
                TC      JOBWAKE

                CS      RUPTREG3        # ADVANCE LOC IF SUCCESSFUL.
                INDEX   LOCCTR
                ADS     LOC

                TCF     TASKOVER

ENDMODE         CA      RUPTREG3        # -0 INDICATES OPERATION COMPLETE BUT
 +1             INDEX   RUPTREG2        # UNSUCCESSFUL: -1 INDICATES COMPLETE AND
                TS      MODECADR        # SUCCESSFUL.
                TCF     TASKOVER
```

```
#          GENERAL STALLING ROUTINE. USING PROGRAMS COME HERE TO WAIT FOR I/O COMPLETION.
#
# PROGRAM DESCRIPTION                                   DATE- 21 FEB 1967
#                                            LOG SECTION IMU MODE SWITCHING
# MOD BY- R.MELANSON TO ADD DOCUMENTATION       ASSEMBLY SUNDISK  REV.  82
#
# FUNCTIONAL DESCRIPTION-
#       TO DELAY FURTHER EXECUTION OF THE CALLING ROUTINE UNTIL ITS SELECTED
#       I/O FUNCTION IS COMPLETE.THE FOLLOWING CHECKS ON THE CALLING ROUTINE:S
#       MODECADR ARE MADE AND ACTED UPON.
#         1) +0 INDICATES INCOMPLETE I/O OPERATION.CALLING ROUTINE IS PUT TO
#            SLEEP.
#         2) -1 INDICATES COMPLETED I/O OPERATION. STALL BYPASSES JOBSLEEP
#            CALL AND RETURNS TO CALLING ROUTINE AT L+3
#         3) -0 INDICATES COMPLETED I/O WITH FAILURE. STALL CLEARS MODECADR
#            AND RETURNS TO CALLING ROUTINE AT L+2.
#         4) VALUE GREATER THAN 0 INDICATES TWO ROUTINES CALLING FOR USE OF
#            SAME DEVICE.  STALL EXITS TO ABORT WHICH EXECUTES A PROGRAM
#            RESTART WHICH IN TURN CLEARS ALL MODECADR REGISTERS.
#
#  CALLING SEQUENCE-
#       L       TC       BANKCALL
#       L+1     CADR (ONE OF 5 STALL ADDRESSES I.E. IMUSTALL,OPTSTALL,RADSTALL,
#                         AOTSTALL,OR ATTSTALL)
#
# NORMAL-EXIT MODE-
#       TCF  JOBSLEEP OR TCF   MODEXIT
#
# ALARM OR ABORT EXIT MODE-
#       TC   ABORT
#
# OUTPUT-
#       MODECADR= CADR IF JOBSLEEP
#       MODECADR=+0    IF I/O COMPLETE
#       BUF2=L+3       IF I/O COMPLETE AND GOOD.
#       BUF2=L+2 IF I/O COMPLETE BUT FAILED.
#
# ERASABLE INITIALIZATION-
#       BUF2 CONTAINS RETURN ADDRESS PLUS 1,(L+2)
#       BUF2+1 CONTAINS FBANK VALUE OF CALLING ROUTINE.
#       MODECADR OF CALLING ROUTINE CONTAINS +0,-1,-0 OR  CADR RETURN ADDRESS.
#
# DEBRIS-
#       RUPTREG2 AND CALLING ROUTINE MODECADR.

AOTSTALL        CAF     ONE             # AOT.
                TC      STALL

RADSTALL        CAF     TWO
                TCF     STALL
```

```
OPTSTALL        EQUALS  AOTSTALL

IMUSTALL        CAF     ZERO            # IMU.

STALL           INHINT
                TS      RUPTREG2        # SAVE DEVICE INDEX.
                INDEX   A               # SEE IF OPERATION COMPLETE.
                CCS     MODECADR
                TCF     MODABORT        # ALLOWABLE STATES ARE +0, -1, AND -0.
                TCF     MODESLP         # OPERATION INCOMPLETE.
                TCF     MODEGOOD        # COMPLETE AND GOOD IF = -1.

MG2             INDEX   RUPTREG2        # COMPLETE AND FAILED IF -0.  RESET TO +0.
                TS      MODECADR        # RETURN TO CALLER.
                TCF     MODEEXIT

MODEGOOD        CCS     A               # MAKE SURE INITIAL STATE -1.
                TCF     MODABORT

                INCR    BUF2            # IF SO, INCREMENT RETURN ADDRESS AND
                TCF     MG2             # RETURN IMMEDIATELY, SETTING CADR = +0.

MODESLP         TC      MAKECADR        # CALL FROM SWITCHABLE FIXED ONLY.
                INDEX   RUPTREG2
                TS      MODECADR
                TCF     JOBSLEEP

MODABORT        DXCH    BUF2
                TC      BAILOUT1        # TWO PROGRAMS USING THE SAME DEVICE.
                OCT     1210
```

```
# CONSTANTS FOR MODE SWITCHING ROUTINES

BITS3&4         =       OCT14
BITS4&6         =       OCT50
BITS4-5         OCT     00030
IMUSEFLG        EQUALS  BIT8            # INTERPRETER SWITCH 7.
-COMMAX         DEC     -191
-COMMAX-        DEC     -192
600MS           DEC     60
IMUFIN20        =       IMUFINE
GOMANUR         CA      ATTCADR         # IS KALCMANU FREE
                EXTEND
                BZF     +3

                TC      POODOO          # NO
                OCT     1210            # 2 TRYING TO USE SAME DEVICE

 +3             EXTEND
                DCA     BUF2
                DXCH    ATTCADR         # SAVE FINAL RETURN FOR KALCMAN3

                CA      BBANK
                MASK    SEVEN
                ADS     ATTCADR +1

                CA      PRIORITY
                MASK    PRIO37
                TS      ATTPRIO         # SAVE USERS PRIO

                CAF     KALEBCON        # SET EBANK FOR KALCMAN3
                TS      EBANK
                TC      POSTJUMP
                CADR    KALCMAN3
KALEBCON        ECADR   BCDU
```

```
# PROGRAM DESCRIPTION
# IMU STATUS CHECK ROUTINE R02 (SUBROUTINE UTILITY)
# MOD NO - 1
# MOD BY - N.BRODEUR
# FUNCTIONAL DESCRIPTION
#
# TO CHECK WHETHER IMU IS ON AND IF ON WHETHER IT IS ALIGNED TO AN
# ORIENTATION KNOWN BY THE CMC. TO REQUEST SELECTION OF THE APPROPRIATE
# PROGRAM IF THE IMU IS OFF OR NOT ALIGNED TO AN ORIENTATION KNOWN BY THE
# CMC. CALLED THROUGH BANKCALL
# CALLING SEQUENCE-
#
# L         TC      BANKCALL
# L+1       CADR    R02BOTH
# SUBROUTINES CALLED
#
#       VARALARM
#       FLAGUP
# NORMAL EXIT MODES
#
# AT L+2 OF CALLING SEQUENCE
# ALARM OR ABORT EXIT MODES
#       GOTOPOOH, WITH ALARM
# ERASABLE INITIALIZATION REQUIRED
#
# NONE
# DEBRIS
#
# CENTRALS-A,Q,L

                BANK    34
                SETLOC  R02
                BANK
                COUNT*  $$/R02
DEC51           DEC     51
R02BOTH         CAF     REFSMBIT
                MASK    FLAGWRD3
                CCS     A
                TC      R02ZERO         # ZERO IMUS

                CA      IMODES30
                MASK    BIT9            # IS ISS INITIALIZED
                EXTEND
                BZF     +2
                CS      BIT4            # SEND IMU ALARM CODE 210
                AD      OCT220          # SEND REFSMM ALARM
                TC      VARALARM

                TC      GOTOPOOH

R02ZERO         TC      UPFLAG
```

```
                ADRES   IMUSE
                TCF     SWRETURN
OCT220          OCT     220
```

```
# PROGRAM DESCRIPTION   P06   10FEB67
#
# TRANSFER THE ISS/CMC FROM THE OPERATE TO THE STANDBY CONDITION.
#
# THE NORMAL CONDITION OF READINESS OF THE GNCS WHEN NOT IN USE IS STANDBY. IN THIS CONDITION THE IMU
# HEATER POWER IS ON. THE IMU OPERATE POWER IS OFF. THE COMPUTER POWER IS ON. THE OPTICS POWER IS OFF. THE
# CMC  STANDBY ON THE MAIN AND LEB DISKYS IS ON.
#
# CALLING SEQUENCE:
#         ASTRONAUT REQUEST THROUGH DSKY     V37E 06E.
#
# SUBROUTINES CALLED:
#         GOPERF1
#         BANKCALL
#         FLAGDOWN
#
```

```
# PRESTAND PREPARES FOR STANDBY BY SNAPSHOTTING THE SCALER AND TIME1 TIME2
# THE LOW 5 BITS OF THE SCALER ARE INSPECTED TO INSURE COMPATIBILITY
# BETWEEN THE SCALER READING AND THE TIME1 TIME2 READING.

                SETLOC  P05P06
                BANK

                EBANK=  TIME2SAV
                COUNT*  $$/P06

P06             TC      UPFLAG          # SET NODOV37 BIT
                ADRES   NODOFLAG

PRESTAND        INHINT
                EXTEND
                DCA     TIME2           # SNAPSHOT TIME1TIME2
                DXCH    TIME2SAV
                TC      SCALPREP
                TC      PRESTAND        # T1,T2,SCALER NOT COMPATIBLE
                DXCH    MPAC            # T1,T2 AND SCALER OK
                DXCH    SCALSAVE        # STORE SCALER
                INHINT
                TC      BANKCALL
                CADR    RNDREFDR        # REFSMM, DRIFT, TRACK FLAGS DOWN

                TC      DOWNFLAG
                ADRES   IMUSE           # IMUSE DOWN
                TC      DOWNFLAG
                ADRES   RNDVZFLG        # RNDVZFLG DOWN

                CAF     BIT11
                EXTEND
                WOR     CHAN13          # SET STANDBY ENABLE BIT

                TC      PHASCHNG        # SET RESTART TO POSTAND WHEN STANDBY
                OCT     07024           #       RECOVERS
                OCT     20000
                EBANK=  SCALSAVE
                2CADR   POSTAND

                CAF     OCT62
                TC      BANKCALL
                CADR    GOPERF1
                TCF     -3
                TCF     -4
                TCF     -5

OCT62           EQUALS  .5SEC           # DEC 50 = OCT 62

# THE LOW 5 BITS OF THE SCALER READS 10000 FOR THE FIRST INTERVAL AFTER A
```

```
# T1 INCREMENT. IF SCALPREP DETECTS THIS INTERVAL THE T1,T2 AND SCALER
# DATA ARE NOT COMPATIBLE AND RETURN IS TO L+1 FOR ANOTHER READING OF THE
# DATA. OTHERWISE, THE RETURN IS TO L+2 TO PROCEED. ROUTINE ALSO PREPARES
# THE SCALER READING FOR COMPUTATION OF THE INCREMENT TO UPDATE T1T2. (THE
# 10 MS BIT (BIT 6) OF THE SCALER IS INCREMENTED 5 MS OUT OF PHASE FROM
# T1.) ADDITION OF 5 MS (BIT 5) TO THE SCALER READING HAS THE EFFECT OF
# ADJUSTING BIT 6 IN THE SCALER TO BE IN PHASE WITH BIT 1 OF T1. THE LOW 5
# BITS OF THE SCALER READING ARE THEN SET TO ZERO, TO TRUNCATE THE SCALER
# DATA TO 10 MS. RESULTS ARE STORED IN MPAC, +1.

SCALPREP        EXTEND
                QXCH    MPAC +2
                TC      FINETIME +1
                RELINT
                DXCH    MPAC
                CA      BIT5            # ADD 5 MS TO THE SCALER READING.
                TS      L
                CA      ZERO
                DAS     MPAC
                CS      LOW5            # SET LOW 5 BITS OF (SCALER+5MS) TO ZERO
                MASK    MPAC +1         # AND STORE RESULTS IN MPAC,+1.
                XCH     MPAC +1
                MASK    LOW5            # TEST LOW 5 BITS OF SCALER FOR THE FIRST
                                        # INTERVAL AFTER THE T1 INCREMENT
                                        # (NOW = 00000, SINCE BIT 5 ADDED).
                CCS     A               # IS IT 1ST INTERVAL AFTER T1 INCREMENT
                INCR    MPAC +2         # NO
                TC      MPAC +2         # YES

# POSTAND RECOVERS TIME AFTER STANDBY.THE SCALER IS SNAPSHOTTED AND THE
# TIME1 TIME2 COUNTER IS SET TO ZERO. THE LOW 5 BITS OF THE SCALER ARE
# INSPECTED TO INSURE COMPATIBILITY BETWEEN THE SCALER READING AND THE
# CLEARING OF THE TIME COUNTER.  IT THEN COMPUTES THE DIFFERENCE IN SCALER
# VALUES (IN DP) AND ADDS THIS TO THE PREVIOUSLY SNAPSHOTTED VALUES OF
# TIME1 TIME2 AND PLACES THIS NEW TIME INTO THE TIME1 TIME2 COUNTER.

                COUNT*  $$/P05

POSTAND         CS      BIT11           # RECOVER TIME AFTER STANDBY.
                EXTEND
                WAND    CHAN13          # CLEAR STANDBY ENABLE BIT
                INHINT
                CA      ZERO
                TS      L
                DXCH    TIME2           # CLEAR TIME1TIME2
                TC      SCALPREP        # STORE SCALER IN MPAC, MPAC+1
                TC      POSTAND +3      # T1,T2,SCALER NOT COMPATIBLE
                EXTEND                  # T1,T2 AND SCALER OK
                DCS     SCALSAVE
                DAS     MPAC            # FORM DP DIFFERENCE OF POSTSTANDBY SCALER
```

```
                CAF     BIT10           # MINUS PRESTANDBY SCALER AND SHIFT RIGHT
                TC      SHORTMP         # 5 TO ALIGN BITS WITH TIME1TIME2.
                CAF     ZERO
                TS      MPAC +2         # NEEDED FOR TP AGREE
                TC      TPAGREE         # MAKE DP DIFF AGREE
                CCS     MPAC
                TC      POSTCOM         # IF DP DIFF NET +, NO SCALER OVERFLOW
                TC      POSTCOM         # BETWEEN PRE AND POST STANDBY.
                TC      +1              # IF DP DIFF NET -, SCALER OVERFLOWED. ADD
                CAF     BIT10           # BIT 10 TO HIGH DIFF TO CORRECT.
                ADS     MPAC
POSTCOM         EXTEND                  # C(MPAC,+1) IS MAGNITUDE OF DELTA SCALER.
                DCA     TIME2SAV        # PRESTANDBY TIME1TIME2
                DAS     MPAC
                TC      TPAGREE         # FORCE SIGN AGREEMENT
                DXCH    MPAC            # UPDATED VALUE FOR T1,T2.
                DAS     TIME2           # LOAD UPDATED VALUE INTO T1,T2, WITH
                TC      DOWNFLAG        # CLEAR NODOFLAG
                ADRES   NODOFLAG

                TC      GOTOPOOH
```

```
                BANK    14
                SETLOC  KEYRUPT
                BANK
                COUNT*  $$/KEYUP

KEYRUPT1        TS      BANKRUPT
                XCH     Q
                TS      QRUPT
                TC      LODSAMPT        # TIME IS SNATCHED IN RUPT FOR NOUN 65.
                CAF     LOW5
                EXTEND
                RAND    MNKEYIN         # CHECK IF KEYS 5M-1M ON
KEYCOM          TS      RUPTREG4
                CS      FLAGWRD5
                MASK    DSKYFBIT
                ADS     FLAGWRD5

ACCEPTUP        CAF     CHRPRIO         # (NOTE: RUPTREG4 = KEYTEMP1)
                TC      NOVAC
                EBANK=  DSPCOUNT
                2CADR   CHARIN

                CA      RUPTREG4
                INDEX   LOCCTR
                TS      MPAC            # LEAVE 5 BIT KEY CDE IN MPAC FOR CHARIN
                TC      RESUME
```

```
# UPRUPT PROGRAM

UPRUPT          TS      BANKRUPT
                XCH     Q
                TS      QRUPT
                TC      LODSAMPT        # TIME IS SNATCHED IN RUPT FOR NOUN 65.
                CAF     ZERO
                XCH     INLINK
                TS      KEYTEMP1
                CAF     BIT3            # TURN ON UPACT LIGHT
                EXTEND                  # (BIT 3 OF CHANNEL 11)
                WOR     DSALMOUT
UPRPT1          CAF     LOW5            # TEST FOR TRIPLE CHAR REDUNDANCY
                MASK    KEYTEMP1        # LOW5 OF WORD
                XCH     KEYTEMP1        # LOW5 INTO KEYTEMP1
                EXTEND
                MP      BIT10           # SHIFT RIGHT 5
                TS      KEYTEMP2
                MASK    LOW5            # MID 5
                AD      HI10
                TC      UPTEST
                CAF     BIT10
                EXTEND
                MP      KEYTEMP2        # SHIFT RIGHT 5
                MASK    LOW5            # HIGH 5
                COM
                TC      UPTEST

UPCK            CS      ELRCODE         # CODE IS GOOD. IF CODE = 'ERROR RESET',
                AD      KEYTEMP1        # CLEAR UPLOCKFL(SET BIT4 OF FLAGWRD7 = 0)
                EXTEND                  # IF CODE DOES NOT = 'ERROR RESET', ACCEPT
                BZF     CLUPLOCK        # CODE ONLY IF UPLOCKFL IS CLEAR (=0).

                CAF     UPLOCBIT        # TEST UPLOCKFL FOR 0 OR 1
                MASK    FLAGWRD7
                CCS     A
                TC      RESUME          # UPLOCKFL = 1
                TC      ACCEPTUP        # UPLOCKFL = 0

CLUPLOCK        CS      UPLOCBIT        # CLEAR UPLOCKFL (I.E.,SET BIT 4 OF )
                MASK    FLAGWRD7        # FLAGWRD7 = 0)
                TS      FLAGWRD7
                TC      ACCEPTUP

                                        # CODE IS BAD
TMFAIL2         CS      FLAGWRD7        # LOCK OUT FURTHER UPLINK ACTIVITY
                MASK    UPLOCBIT        # (BY SETTING UPLOCKFL = 1) UNTIL
                ADS     FLAGWRD7        # 'ERROR RESET' IS SENT VIA UPLINK.
                TC      RESUME
UPTEST          AD      KEYTEMP1
```

```
                 CCS      A
                 TC       TMFAIL2
HI10             OCT      77740
                 TC       TMFAIL2
                 TC       Q

ELRCODE          OCT      22

# 'UPLINK ACTIVITY LIGHT' IS TURNED OFF BY .....
#          1.     VBRELDSP
#          2.     ERROR RESET
#          3.     UPDATE PROGRAM(P27) ENTERED BY V70,V71,V72,AND V73.
#
#
# THE RECEPTION OF A BAD CODE(I.E  CCC FAILURE) LOCKS OUT FURTHER UPLINK ACTIVITY BY SETTING BIT4 OF FLAGWRD7 = 1.
# THIS INDICATION WILL BE TRANSFERRED TO THE GROUND BY THE DOWNLINK WHICH DOWNLINKS ALL FLAGWORDS.
# WHEN UPLINK ACTIVITY IS LOCKED OUT ,IT CAN BE ALLOWED WHEN THE GROUND UPLINKS AND 'ERROR RESET' CODE.
# (IT IS RECOMMENDED THAT THE 'ERROR LIGHT RESET' CODE IS PRECEEDED BY 16 BITS THE FIRST OF WHICH IS 1 FOLLOWED
# BY 15 ZEROES. THIS WILL ELIMINATE EXTRANEOUS BITS FROM INLINK WHICH MAY HAVE BEEN LEFT OVER FROM THE ORIGINAL
# FAILURE)
# UPLINK ACTIVITY IS ALSO ALLOWED(UNLOCKED) DURING FRESH START WHEN FRESH START SETS BIT4 OF FLAGWRD7 = 0.
                 CS       XDSPBIT
```

```
# DISPLAYS CAN BE CLASSIFIED INTO THE FOLLOWING CATEGORIES --
#       1.  PRIORITY DISPLAYS -- DISPLAYS WHICH TAKE PRIORITY OVER ALL OTHER DISPLAYS.  USUALLY THESE DISPLAYS ARE SENT
#           OUT UNDER CRITICAL ALARM CONDITIONS.
#       2.  EXTENDED VERB DISPLAYS -- ALL EXTENDED VERBS AND MARK ROUTINES SHOULD USE EXTENDED VERB (MARK) DISPLAYS.
#       3.  NORMAL DISPLAYS -- ALL MISSION PROGRAM DISPLAYS WHICH INTERFACE WITH THE ASTRONAUT DURING THE NORMAL
#           SEQUENCE OF EVENTS.
#       4.  MISC. DISPLAYS -- ALL DISPLAYS NOT HANDLED BY THE DISPLAY INTERFACE ROUTINES.  THESE INCLUDE SUCH DISPLAYS AS
#           MM DISPLAYS AND SPECIAL PURPOSE DISPLAYS HANDLED BY PINBALL.
#       5.  ASTRONAUT INITIATED DISPLAYS -- ALL DISPLAYS INITIATED EXTERNALLY.
#
# THE FOLLOWING TERMS ARE USED TO DESCRIBE THE STATUS OF DISPLAYS --
#       1.  ACTIVE -- THE DISPLAY WHICH IS (1) BEING DISPLAYED TO THE ASTRONAUT AND WAITING FOR A RESPONSE OR
#           (2) WAITING FIRST IN LINE FOR THE ASTRONAUT TO FINISH USING THE DSKY OR (3) BEING DISPLAYED ON THE DSKY
#           BUT NOT WAITING FOR A RESPONSE.
#       2.  INACTIVE -- A DISPLAY WHICH HAS (1) BEEN ACTIVE BUT WAS INTERRUPTED BY A DISPLAY OF HIGHER PRIORITY,
#           (2) BEEN PUT INTO THE WAITING LIST AT TIME IT WAS REQUESTED DUE TO THE FACT A HIGHER PRIORITY DISPLAY
#           WAS ALREADY DOING, (3) BEEN INTERRUPTED BY THE ASTRONAUT (CALLED A PINBRANCH CONDITION, SINCE THIS TYPE
#           OF INACTIVE DISPLAY IS USUALLY REACTIVATED ONLY BY PINBALL) OR (4) A DISPLAY WHICH HAS FINISHED BUT STILL
#           HAS INFO SAVED FOR RESTART PURPOSES.
#
# DISPLAY PRIORITIES WORK AS FOLLOWS --
#       INTERRUPTS --
#               1.  THE ASTRONAUT CAN INTERRUPT ANY DISPLAY WITH AN EXTERNAL DISPLAY REQUEST.
#               2.  INTERNAL DISPLAYS CAN NOT BE SENT OUT WHEN THE ASTRONAUT IS USING THE DSKY.
#               3.  PRIORITY DISPLAYS INTERRUPT ALL OTHER TYPES OF INTERNAL DISPLAYS.  A PRIORITY DISPLAY INTERRUPTING ANOTHER
#                   PRIORITY DISPLAY WILL CAUSE AN ABORT UNLESS BIT14 IS SET FOR THE LINUS ROUTINE.
#               4.  A MARK DISPLAY INTERRUPTS ANY NORMAL DISPLAY.
#               5.  A MARK THAT INTERRUPTS A MARK COMPLETELY REPLACES IT.
#
#       ORDER OF WAITING DISPLAYS --
#               1.  ASTRONAUT
#               2.  PRIORITY
#               3.  INTERRUPTED MARK
#               4.  INTERRUPTED NORMAL
#               5.  MARK TO BE REQUESTED (SEE DESCRIPTION OF ENDMARK)
#               6.  MARK WAITING
#               7.  NORMAL WAITING
#
```

```
# THE DISPLAY ROUTINES ARE INTENDED TO SERVE AS AN INTERFACE BETWEEN THE USER AND PINBALL.  THE
# FOLLOWING STATEMENTS CAN BE MADE ABOUT NORMAL DISPLAYS AND PRIORITY DISPLAYS (A DESCRIPTION OF MARK ROUTINES
# WILL FOLLOW LATER):
#       1.  ALL ROUTINES THAT END IN R HAVE AN IMMEDIATE RETURN TO THE USER.  FOR ALL FLASHING DISPLAYS THIS RETURN
#           IS TO THE USER'S CALL CADR +4.  FOR THE ONLY NON-FLASHING IMMEDIATE RETURN DISPLAY (GODSPR) THIS RETURN
#           IS TO THE USER'S CALLING LOC +1.
#       2.  ALL ROUTINES NOT ENDING IN R DO NOT DO AN IMMEDIATE RETURN TO THE USER.
#       3.  ALL ROUTINES THAT END IN R START A SEPARATE JOB (MAKEPLAY) WITH USER'S JOB PRIORITY.
#       4.  ALL ROUTINES NOT ENDING IN R BRANCH DIRECTLY TO MAKEPLAY WHICH MAKES THESE DISPLAYS A PART OF THE
#           USER'S JOB.
#       5.  ALL DISPLAY ROUTINES ARE CALLED VIA BANKCALL.
#       6.  TO RESTART A DISPLAY THE USER WILL GENERALLY USE A PHASE OF ONE WITH DESIRED RESTART GROUP (SEE
#           DESCRIPTION OF RESTARTS).
#       7.  ALL FLASHING DISPLAYS HAVE 3 RETURNS TO THE USER FROM ASTRONAUT RESPOSES.  A TERMINATE (V34) BRANCHES
#           TO THE USER'S CALL CADR +1.  A PROCEED (V33) BRANCHES TO THE USER'S CALL CADR +2.  AN ENTER OR RECYCLE
#           (V32) BRANCHES TO THE USER'S CALL CADR +3.
#       8.  ALL ROUTINES MUST BE USED UNDER EXECUTIVE CONTROL
#
# A DESCRIPTION OF EACH ROUTINE WITH AN EXAMPLE FOLLOWS:
#       GODSP IS USED TO DISPLAY A VERB NOUN ARRIVING IN A.  NO RETURN IS MADE TO THE USER.
#               1.  GODSP IS NOT RESTARTABLE
#               2.  A VERB PASTE WITH GODSP ALWAYS TURNS ON THE FLASH.
#                               CAF     VXXNYY
#                               TC      BANKCALL
#                               CADR    GODSP
#                       VXXNYY  OCT     OXXYY
#       GODSPR IS THE SAME AS GODSP ONLY RETURN IS TO THE USER.
#                               CAF     VXXNYY
#                               TC      BANKCALL
#                               CADR    GODSPR
#                               ...     ...             # IMMEDIATE RETURN OF GODSPR
#       GOFLASH DISPLAYS A FLASHING VERB NOUN WITH NO IMMEDIATE RETURN TO THE USER.  3 RETURNS ARE POSSIBLE FORM
#       THE ASTRONAUT (SEE NO. 7 ABOVE).
#                               CAF     VXXNYY          # VXX NYY WILL BE A FLASHING VERB NOUN.
#                               TC      BANKCALL
#                               CADR    GOFLASH
#                               ...     ...             # TERMINATE RETURN
#                               ...     ...             # PROCEED RETURN
#                               ...     ...             # ENTER OR RECYCLE RETURN
#       GOPERF1 IS ENTERED WITH DESIRED CHECKLIST VALUE IN A.  GOPERF1 WILL DISPLAY THIS VALUE IN R1 BY MEANS OF A
```

```
#       V01 N25.  A FLASHING PLEASE PERFORM ON CHECKLIST (V50 N25) IS THEN DISPLAYED.  NO IMMEDIATE RETURN IS MADE TO
#       USER (SEE NO. 7 ABOVE).
#       GOPERF1 BLANKS REGISTERS R2 AND R3
#                               CAF     OCTXX           # CODE FOR CHECKLIST VALUE XX
#                               TC      BANKCALL
#                               CADR    GOPERF1
#                               ...     ...             # TERMINATE RETURN
#                               ...     ...             # PROCEED RETURN
#                               ...     ...             # ENTER RETURN
#       GOPERF2 IS ENTERED WITH A VARIABLE NOUN AND V01 (V00 FOR N10 OR N11) IN A.  GOPERF2 WILL FIRST DISPLAY THE
#       REQUESTED NOUN BY MEANS OF A V01NYY OR A V00NYY.  PLEASE PERFORM ON NOUN (V50 NYY) THEN BECOMES A FLASHING
#       DISPLAY.  NO IMMEDIATE RETURN IS MADE TO THE USER (SEE NO. 7 ABOVE).
#       GOPERF2 DOES NOT BLANK ANY REGISTERS
#                               CAF     VXXNYY          # VARIABLE NOUN YY. XX=0 OR 01.
#                               TC      BANKCALL
#                               CADR    GOPERF2
#                               ...     ...             # TERMINATE RETURN
#                               ...     ...             # PROCEED RETURN
#                               ...     ...             # ENTER RETURN
#       GOPERF3 IS USED FOR A PLEASE PERFORM ON A PROGRAM NUMBER.  THE DESIRED PROGRAM NO. IS ENTERED IN A.  GOPERF3
#       DISPLAYS THE NO. BY MEANS OF A V06 N07 FOLLOWED BY A FLASHING V50 N07 FOR A PLEASE PERFORM.  NO IMMEDIATE RETURN
#       IS MADE TO THE USER (SEE NO. 7 ABOVE).
#       GOPERF3 BLANKS REGISTERS R2 AND R3
#                               CAF     DECXX           # REQUEST PERFORM ON PXX
#                               TC      BANKCALL
#                               CADR    GOPERF3
#                               ...     ...             # TERMINATE RETURN
#                               ...     ...             # PROCEED RETURN
#                               ...     ...             # ENTER RETURN
#       GOPERF4 IS USED FOR A PLEASE PERFORM ON AN OPTION.  THE DESIRED OPTION IS ENTERED IN A AND STORED IN OPTION1.
#       GOPERF4 DISPLAYS R1 AND R2 BY MEANS OF A V04N06 FOLLOWED BY A FLASHING V50N06 FOR A PLEASE PERFORM.  NO
#       IMMEDIATE RETURN IS MADE TO THE USER (SEE NO. 7 ABOVE).
#                               CAF     OCTXX           # REQUEST PERFORM ON OPTION XX
#                               TC      BANKCALL
#                               CADR    GOPERF4
#                               ...     ...             # TERMINATE RETURN
#                               ...     ...             # PROCEED RETURN
#                               ...     ...             # ENTER RETURN
#       GOPERF4 BLANKS REGISTER R3.
```

```
#       GODSPRET IS USED TO DISPLAY A VERB NOUN ARRIVING IN A WITH A RETURN TO THE USER AFTER THE DISPLAY HAS BEEN SENT
#       OUT.
#                               CAF     VXXXNYY
#                               TC      BANKCALL
#                               CADR    GODSPRET
#                               ...     ...             # RETURN TO USER.
#       REGODSP IS USED TO DISPLAY A VERB NOUN ARRIVING IN A.  REGODSP IS THE SAME AS GODSP ONLY REGODSP REPLACES ANY
#       ACTIVE NORMAL DISPLAY IF ONE WAS ACTIVE.
#                               CAF     VXXNYY
#                               TC      BANKCALL
#                               CADR    REGODSP
#       REFLASH IS THE SAME AS GOFLASH ONLY REFLASH REPLACES ANY ACTIVE NORMAL DISPLAY IF ONE WAS ACTIVE.
#                               CAF     VXXNYY          # VXX NYY WILL BE A FLASHING VERB NOUN
#                               TC      BANKCALL
#                               CADR    REFLASH
#                               ...     ...             # TERMINATE RETURN
#                               ...     ...             # PROCEED RETURN
#                               ...     ...             # ENTER RETURN
#       GOFLASHR IF SAME AS GOFLASH ONLY AN IMMEDIATE RETURN IS MADE TO THE USER'S CALL CADR +4.
#                               CAF     VXXNYY
#                               TC      BANKCALL
#                               CADR    GOFLASHR
#                               ...     ...             # TERMINATE RETURN
#                               ...     ...             # PROCEED RETURN
#                               ...     ...             # ENTER OR RECYCLE RETURN
#                               ...     ...             # IMMEDIATE RETURN FROM GOFLASHR
#       GOPERF1R IS THE SAME AS GOPERF1 ONLY GOPERF1R HAS AN IMMEDIATE RETURN TO USER'S CALL CADR +4.
#       GOPERF1R BLANKS REGISTERS R2 AND R3
#                               CAF     OCTXX           # CODE FOR CHECKLIST VALUE XX.
#                               TC      BANKCALL
#                               CADR    GOPERF1R
#                               ...     ...             # TERMINATE RETURN
#                               ...     ...             # PROCEED RETURN
#                               ...     ...             # ENTER RETURN
#                               ...     ...             # IMMEDIATE RETURN FROM GOPERF1R
#       GOPERF2R IS THE SAME AS GOPERF2 ONLY AN IMMEDIATE RETURN IS MADE TO USER'S CALL CADR +4.
```

```
#       GOPERF2R DOES NOT BLANK ANY REGISTERS
#                               CAF     VXXXNYY         # VARIABLE NOUN YY REQUESTED.  XX=00 OR 01
#                               TC      BANKCALL
#                               CADR    GOPERF2R
#                               ...     ...             # TERMINATE RETURN
#                               ...     ...             # PROCEED RETURN
#                               ...     ...             # ENTER RETURN
#                               ...     ...             # IMMEDIATE RETURN HERE FROM GOPERF2R
#       GOPERF3R IS THE SAME AS GOPERF3 ONLY AN IMMEDIATE RETURN IS MADE TO USER'S CALL CADR +4.
#       GOPERF3R BLANKS REGISTERS R2 AND R3
#                               CAF     PROGXX          # PERFORM PROGRAM XX
#                               TC      BANKCALL
#                               CADR    GOPERF3R
#                               ...     ...             # TERMINATE RETURN
#                               ...     ...             # PROCEED RETURN
#                               ...     ...             # ENTER RETURN
#                               ...     ...             # GOPERF3R IMMEDIATELY RETURNS HERE
#       GOPERF4R IS THE SAME AS GOPERF4 ONLY AN IMMEDIATE RETURN IS MADE TO USER'S CALL CADR +4.
#                               CAF     OCTXX           # REQUEST PERFORM ON OPTIONXX
#                               TC      BANKCALL
#                               CADR    GOPERF4R
#                               ...     ...             # TERMINATE RETURN
#                               ...     ...             # PROCEED RETURN
#                               ...     ...             # ENTER RETURN
#                               ...     ...             # IMMEDIATE RETURN TO USER
#       GOPERF4R BLANKS REGISTER R3.
#       REFLASHR IS THE SAME AS REFLASH ONLY AN IMMEDIATE RETURN IS MADE TO THE USER'S CALL CADR +4.
#                               CAF     VXXNYY          # VXX NYY WILL BE A FLASHING VERB NOUN
#                               TC      BANKCALL
#                               CADR    REFLASHR
#                               ...     ...             # TERMINATE RETURN
#                               ...     ...             # PROCEED RETURN
#                               ...     ...             # ENTER RETURN
#                               ...     ...             # IMMEDIATE RETURN TO USER
#       REGODSPR IS THE SAME AS REGODSP ONLY A RETURN (IMMEDIATE) IS MADE TO THE USER.
```

```
#                                CAF      VXXNYY
#                                TC       BANKCALL
#                                CADR     REGODSPR
#                                ...      ...              # IMMEDIATE RETURN TO USER
```

```
#       GOMARK IS USED TO DISPLAY A MARK VERB NOUN ARRIVING IN A.  NO RETURN IS MADE TO THE USER.
#       GOXDSP = GOMARK
#                               CAF     VXXNYY          # VXXNYY CONTAINS VERB AND NOUN
#                               TC      BANKCALL
#                               CADR    GOMARK          # OTHER EXTENDED VERBS USE CADR GOXDSP
#       GOMARKR IS THE SAME AS GOMARK ONLY RETURN IS TO THE USER.
#       GOXDSPR = GOMARKR
#                               CAF     VXXNYY
#                               TC      BANKCALL
#                               CADR    GOMARKR         # OTHER EXTENDED VERBS USE CADR GOXDSPR
#                               ...     ...             # IMMEDIATE RETURN OF GOMARKR
#       GOMARKF DISPLAYS A FLASHING MARK VERB NOUN WITH NO IMMEDIATE RETURN TO THE USER.  3 RETURNS ARE POSSIBLE FROM
#       THE ASTRONAUT (SEE NO. 7 ABOVE).
#       GOXDSPF = GOMARKF
#                               CAF     VXXNYY          # VXXNYY WILL BE A FLASHING MARK VERB NOUN
#                               TC      BANKCALL
#                               CADR    GOMARKFR        # OTHER EXTENDED VERBS USE CADR GOXDSPFR
#                               ...     ...             # TERMINATE RETURN
#                               ...     ...             # PROCEED RETURN
#                               ...     ...             # ENTER OR RECYCLE RETURN
#       GOMARKFR IS THE SAME AS GOMARKF ONLY AN IMMEDIATE RETURN IS MADE TO THE USER CALL CADR +4.
#       GOXDSPFR = GOMARKFR
#                               CAF     VXXNYY          # FLASHING MARK VERB NOUN
#                               TC      BANKCALL
#                               CADR    GOMARKFR        # OTHER EXTENDED VERBS USE CADR GOXDSPFR
#                               ...     ...             # TERMINATE RETURN
#                               ...     ...             # PROCEED RETURN
#                               ...     ...             # ENTER OR RECYCLE RETURN
#                               ...     ...             # IMMEDIATE RETURN TO THE USER
#       GOMARK1 IS USED FOR A PLEASE PERFORM ON A MARK REQUEST WITH ONLY 1 ASTRONAUT RETURN TO THE USER.  NO IMMEDIATE
#       RETURN IS MADE.  THE DESIRED MARK PLEASE PERFORM VERB AND DESIRED NOUN IS ENTERED IN A.  GOMARK1 DISPLAYS R1, R2, R
#       MEANS OF A V05NYY FOLLOWED BY A FLASHING V5XNYY FOR A PLEASE PERFORM.  THE ASTRONAUT WILL RESPOND WITH A MARK
#       OR MARK REJECT OR AN ENTER.  THE ENTER IS THE ONLY ASTRONAUT RESPONSE THAT WILL COME BACK TO THE USER.
#                               CAF     V5XNYY          # X=1,2,3,4     Y=NOUN
#                               TC      BANKCALL
```

```
#                               CADR    GOMARK1
#                               ...     ...             # ENTER RETURN
#       *** IF BLANKING DESIRED ON NON-R ROUTINES, NOTIFY DISPLAYER.
#
#       GOMARK1R IS THE SAME AS A GOMARK1 ONLY AN IMMEDIATE RETURN IS MADE TO THE USER'S CALL CADR +2.
#                               CAF     V5XNYY          # X=1,2,3,4     YY=NOUN
#                               TC      BANKCALL
#                               CADR    GOMARK1R
#                               ...     ...             # ASTRONAUT ENTER RETURN
#                               ...     ...             # IMMEDIATE RETURN TO USER
#       GOMARK2 IS THE SAME AS GOMARK1 ONLY 3 RETURNS ARE MADE TO THE USER FROM THE ASTRONAUT.
#                               CAF     V5XNYY          # X=1,2,3,4     YY=NOUN
#                               TC      BANKCALL
#                               CADR    GOMARK2
#                               ...     ...             # TERMINATE RETURN
#                               ...     ...             # PROCEED RETURN
#                               ...     ...             # ENTER RETURN
#       GOMARK2R IS THE SAME AS GOMARK1R ONLY 3 ASTRONAUT RETURNS ARE MADE TO THE USER.
#                               CAF     V5XNYY          # X=0,1,2,3,4   YY=NOUN
#                               TCF     BANKCALL
#                               CADR    GOMARK2R
#                               ...     ...             # TERMINATE RETURN
#                               ...     ...             # PROCEED RETURN
#                               ...     ...             # ENTER RETURN
#                               ...     ...             # IMMEDIATE RETURN TO THE USER.
#       GOMARK3 IS USED FOR A PLEASE PERFORM ON A MARK REQUEST WITH A 3 COMP. DEC DISPLAY.  THE DESIRED MARK PLEASE
#       PERFORM VERB AND NOUN ARE ENTERED IN A.  GOMARK3 DISPLAYS R1, R2, R3 BY MEANS OF A V06NYY FOLLOWED BY A FLASHING
#       V5XNYY FOR A PLEASE PERFORM.  GOMARK3 HAS 3 ASTRONAUT RETURNS TO THE USER WITH NO IMMEDIATE RETURN.
#                               CAF     V5XNYY          # X=1,2,3,4     YY=NOUN
#                               TC      BANKCALL
#                               CADR    GOMARK3
#                               ...     ...             # TERMINATE RETURN
#                               ...     ...             # PROCEED RETURN
#                               ...     ...             # ENTER RETURN
#       GOMARK4 IS THE SAME AS GOMARK3 ONLY R2 AND R3 ARE BLANKED AND R1 IS DISPLAYED IN OCTAL.
#                               CAF     V5XNYY          # X=1,2,3,4     YY=NOUN
#                               TC      BANKCALL
#                               CADR    GOMARK4
#                               ...     ...             # TERMINATE RETURN
#                               ...     ...             # PROCEED RETURN
```

```
#                        ...     ...                     # ENTER RETURN
#       EXDSPRET IS USED TO DISPLAY A VERB NOUN ARRIVING IN A WITH A RETURN MADE TO THE USER AFTER THE DISPLAY HAS BEEN
#       SEND OUT.
#                                CAF     VXNYY
#                                TC      BANKCALL
#                                CADR    EXDSPRET
#                                ...     ...             # RETURN TO USER
#       KLEENEX CLEANS OUT ALL MARK DISPLAYS (ACTIVE AND INACTIVE). A RETURN IS MADE TO THE USER AFTER THE MARK DISPLAYS
#       HAVE BEEN CLEANED OUT.
#                                TC      BANKCALL
#                                CADR    KLEENEX
#                                ...     ...             # RETURN TO USER
#       MARKBRAN IS A SPECIAL PURPOSE ROUTINE USED FOR SAVING JOB VAC AREAS (SEE DESCRIPTION OF MARKBRAN BELOW).
#                                TC      BANKCALL
#                                CADR    MARKBRAN
#                                ...     ...             # BAD RETURN IF MARK DISPLAY NOT ACTIVE
#                                                        # (GOOD RETURN TO IMMEDIATE RETURN LOC OF
#                                                        # LAST FLASHING MARK R ROUTINE)
#       PINBRNCH REESTABLISHES THE LAST ACTIVE FLASHING DISPLAY.  IF THERE IS NO ACTIVE FLASHING DISPLAY, THE DSKY IS
#       BLANKED AND CONTROL IS SENT TO ENDOFJOB.
#                                TC      POSTJUMP
#                                CADR    PINBRNCH
#       PRIODSP IS USED AS A PRIORITY DISPLAY.  IT WILL DISPLAY A GOFLASH TYPE DISPLAY WITH THREE POSSIBLE RETURNS FROM
#       THE ASTRONAUT (SEE NO. 7 ABOVE).
#       THE MAIN PURPOSE OF PRIODSP IS TO REPLACE THE PRESENT DISPLAY WITH A DISPLAY OF HIGHER PRIORITY AND TO
#       PROVIDE A MEANS FOR RESTORING THE OLD DISPLAY WHEN THE PRIORITY DISPLAY
#       IS RESPONDED TO BY THE ASTRONAUT.
#       THE FORMER DISPLAY IS RESTORED BY AN AUTOMATIC BRANCH TO WAKE UP THE DISPLAY THAT WAS INTERRUPTED BY THE
#       PRIO DISPLAY
#                                CAF     VXXNYY          # VXXNYY WILL BE A FLASHING VERB NOUN
#                                TC      BANKCALL
#                                CADR    PRIODSP
#                                ...     ...             # TERMINATE RETURN
#                                ...     ...             # PROCEED RETURN
```

```
#                               ...     ...             # ENTER OR RECYCLE RETURN
#       PRIODSPR IS THE SAME AS PRIODSP ONLY AN IMMEDIATE RETURN IS MADE TO THE USER'S CALL CADR +4.
#                               CAF     VXXNYY          # VXXNYY WILL BE A FLASHING VERB NOUN
#                               TC      BANKCALL
#                               CADR    PRIODSPR
#                               ...     ...             # TERMINATE ACTION
#                               ...     ...             # PROCEED RETURN
#                               ...     ...             # ENTER OR RECYCLE RETURN
#                               ...     ...             # IMMEDIATE RETURN
#       PRIOLARM DOES A V05N09 PRIODSPR.
#
#       CLEANDSP CLEANS OUT ALL NORMAL DISPLAYS (ACTIVE AND INACTIVE).  A RETURN IS MADE TO THE USER AFTER NORMAL
#       DISPLAYS ARE CLEANED OUT.
#                               TC      BANKCALL
#                               CADR    CLEANDSP
#                               ...     ...             # RETURN TO USER
```

```
#
# GENERAL INFORMATION
# -------------------
#
# ALARM OR ABORT EXIT MODE --
#       PRIOBORT        TC      ABORT
#                       OCT     1502
#
#       PRIOBORT IS BRANCHED TO WHEN (1) A NORMAL DISPLAY IS REQUESTED AND ANOTHER NORMAL DISPLAY IS ALREADY ACTIVE
#       (REFLASH AND REGODSP ARE EXCEPTIONS) OR (2) A PRIORITY DISPLAY IS REQUESTED WHEN ANOTHER PRIORITY DISPLAY IS
#       ALREADY ACTIVE (A PRIORITY WITH LINUS BIT14 IS AN EXCEPTION).
#
# ERASABLE INITIALIZATION REQUIRED --
#       ACCOMPLISHED BY FRESH START --  1.  FLAGWRD4 (USED EXCLUSIVELY BY DISPLAY INTERFACE ROUTINES)
#                                       2.  NVSAVE = NORMAL VERB AND NOUN REGISTER.
#                                       3.  EBANKTEM = NORMAL INACTIVE FLAGWORD (ALSO CONTAINS NORMALS EBANK).
#                                       5.  R1SAVE = MARKBRAN CONTROL WORD
#                                       4.  RESTREG = PRIORITY 30 AND SUPERBANK 3.
#
# OUTPUT --
#       NVWORD = PRIO VERB AND NOUN
#       NVWORD +1 (MARKNV) = MARK VERB AND NOUN
#       NVWORD +2 (NVSAVE) = NORMAL VERB AND NOUN
#       DSPFLG (EBANKSAV) = PRIO FLAGWORD (INCLUDING EBANK)
#       DSPFLG +1 (MARKEBAN) = MARK FLAGWORD (INCLUDING EBANK)
#       DSPFLG +2 (EBANKTEM) = NORMAL FLAGWORD (INCLUDING EBANK)
#       CADRFLSH = PRIO USER'S CALL CADR +1 LOCATION
#       CADRFLSH +1 (MARKFLSH) = MARK USER'S CALL CADR +1 LOCATION
#       CADRFLSH +2 (TEMPFLSH) = NORMAL USER'S CALL CADR +1 LOCATION
#       PRIOTIME = TIME EACH PRIO REQUEST FIRST SENT OUT
#       OPTION1 = DESIRED OPTION FROM GOPERF4
#       FLAGWRD4 = BIT INFO FOR CONTROL OF ALL DISPLAY ROUTINES
#       DSPTEM1 = R1 INFO FOR ASTRONAUT FROM PERFORM DISPLAYS (NORMAL)
#
# SUBROUTINES USED -- NVSUB, FLAGUP, FLAGDOWN, ENDOFJOB, BLANKSUB, ABORT, JOBWAKE, JOBSLEEP, FINDVAC, PRIOCHNG,
#       JAMTERM, NVSUBUSY, FLASHON, ENDIDLE, CHANG1, BANKJUMP, MAKECADR, NOVAC
#
# DEBRIS -- (STORED INTO)
#       TEMPORARY TEMPORARIES -- A, Q, L, MPAC +2, MPAC +3, MPAC +4, MPAC +5, MPAC +6, RUPREG2, RUPTREG3, CYL,
#               EBANK, RUPTREG4, LOC, BANKSET, MODE, MPAC, MPAC +1, FACEREG
#       ERASABLES (SHARED AND USED WITH OTHER PROGRAMS) -- CADRSTOR, DSPLIST, LOC, DSPTEM1, OPTION1
#       ERASABLES (USED ONLY BY DISPLAY ROUTINES) -- NVWORD,+1,+2, DSPFLAG,+1,+2, CADRFLSH,+1,+2, PRIOTIME, FLAGWRD4,
```

```
#                 R1SAVE, MARK2PAC
#
# DEBRIS -- (USED BUT NOT STORED INTO) -- NOUNREG, VERBREG, LOCCTR, MONSAVE1
#
# FLAGWORD DESCRIPTIONS --
#       FLAGWRD4 -- SEE DESCRIPTION UNDER LOG SECTION ERASABLE ASSIGNMENTS
#
#       DSPFLG, DSPFLG+1, DSPFLG+2
#       --------------------------
#       BITS 1  BLANK R1
#            2  BLANK R2
#            3  BLANK R3
#            4  FLASHING DISPLAY REQUESTED
#            5  PERFORM DISPLAY REQUESTED
#            6  -----                   EXDSPRET                GODSPRET
#            7  PRIO DISPLAY            -----                   -----
#            8  -----                   DEC MARK PERFORM        -----
#            9  EBANK
#           10  EBANK
#           11  EBANK
#           12  -----                   -----                   V99PASTE
#           13  2ND PART OF PERFORM
#           14  REFLASH OR REDO         -----                   REFLASH OR REDO
#           15  -----                   MARK REQUEST            -----
#
# RESTARTING DISPLAYS --
#
# RULES FOR THE DSKY OPERATOR --
#       1.  PROCEED AND TERMINATE SERVE AS RESPONSES TO REQUESTS FOR OPERATOR RESPONSE (FLASHING V/N).  AS LONG
#           AS THERE IS ANY REQUEST AWAITING OPERATOR RESPONSE, ANY USE OF PROCEED OR TERMINATE WILL SERVE AS
#           RESPONSES TO THAT REQUEST.  CARE SHOULD BE EXERCISED IN ATTEMPTING TO KILL AN OPERATOR INITIATED MONITOR
#           WITH PROCEED AND TERMINATE FOR THIS REASON.
#       2.  THE ASTRONAUT MUST RESPOND TO A PRIORITY DISPLAY NO SOONER THAN 2 SECONDS FROM THE TIME THE
#           PROGRAM SENT OUT THE REQUEST FOR OPERATOR RESPONSE (THE ASTRONAUT WOULD SEE THIS DISPLAY FOR LESS TIME
#           DUE TO TIME IT TAKES TO GET DISPLAY SENT OUT.)  IF THE ASTRONAUT RESPONDS TOO SOON, THE PRIORITY DISPLAY
#           IS SENT OUT AGAIN -- AND AGAIN UNTIL AN ACCUMULATED 2 SECS FROM THE TIME THE FIRST PRIORITY DISPLAY
#           OUT.  THE SAME 2 SEC. DELAY WILL OCCUR AT 163.84 SECS OR IN ANY MULTIPLE OF THAT TIME DUE TO PROGRAM
#           CONSIDERATION.
#       3.  KEY RELEASE BUTTON --
#           A)  IF THE KEY RELEASE LIGHT IS ON, IT SIMPLY RELEASES THE KEYBOARD AND DISPLAY FOR INTERNAL USE.
#           B)  IF THE KEY RELEASE LIGHT IS OFF, AND IF SOME REQUEST FOR OPERATOR RESPONSE (FLASHING V/N) IS STILL
#               AWAITING RESPONSE THEN IT RE-ESTABLISHES THE DISPLAYS THAT ORIGINALLY REQUESTED RESPONSE.
#           IF AN OPERATOR WANTS THEREFORE TO RE-ESTABLISH BUT CONDITION (A) IS ENCOUNTERED, A SECOND DEPRESSION OF
#           KEY RELEASE BUTTON MAY BE NECESSARY.
#       4.  IT IS IMPORTANT TO ANSWER ALL REQUESTS FOR OPERATOR RESPONSE.
#       5.  IT IS ALWAYS GOOD PRACTICE TO TERMINATE AN EXTENDED VERB BEFORE ASKING FOR ANOTHER ONE OR THE SAME ONE
#           OVER AGAIN.
#
# SPECIAL CONSIDERATONS --
```

```
#       1.   MPAC +2 SAVED ONLY IN MARK DISPLAYS
#       2.   GODSP(R), REGODSP(R), GOMARK(R) ALWAYS TURN ON THE FLASH IF ENTERED WITH A PASTE VERB REQUEST.
#       3.   ALL NORMAL DISPLAYS ARE RESTARTABLE EXCEPT GODSP(R), REGODSP(R)
#       4.   ALL EXTENDED VERBS WITH DISPLAYS SHOULD START WITH A TC TESTXACT AND FINISH WITH A TC ENDEXT.
#       5.   GODSP(R) AND REGODSP(R) MUST BE IN THE SAME EBANK AND SUPERBANK AS THE LAST NORMAL DISPLAY RESTARTED
#            BY A .1 RESTART PHASE CHANGE.
#       6.   IN ORDER TO SET UP A NON DISPLAY .1 RESTART POINT, THE USER MUST MAKE CERTAIN THAT RESTREG CONTAINS THE
#            CORRECT PRIORITY AND SUPERBANK AND THAT EBANKTEM CONTAINS THE CO
#       7.   IF CLEANDSP IS RESTARTED VIA A .1 PHASE CHANGE, CAF ZERO SHOULD BE EXECUTED BEFORE THE TC BANKCALL.
```

```
# CALLING SEQUENCE FOR BLANKING
#               CAF     BITX            # X=1,2,3 BLANK R1,R2,R3 RESPECTIVELY
#               TC      BLANKET
#               ...     ...             # RETURN TO USER HERE
# IN ORDER TO USE BLANKET CORRECTLY, THE USER MUST USE A DISPLAY ROUTINE THAT ENDS IN R FIRST FOLLOWED BY THE CALL
# TO BLANKET AT THE IMMEDIATE RETURN LOC.
                BLOCK   02
                SETLOC  FFTAG4
                BANK

                COUNT*  $$/DSPLA
BLANKET         TS      MPAC    +6
                CS      PLAYTEM4
                MASK    MPAC    +6
                INDEX   MPAC    +5
                ADS     PLAYTEM4

                TC      Q

ENDMARK         TC      POSTJUMP
                CADR    MARKEND

CLEARMRK        CAF     ZERO
                TS      EXTVBACT

 +2             INHINT
                CS      XDSPBIT
                MASK    FLAGWRD4
                TS      FLAGWRD4

                RELINT
                TC      Q

# *** ALL EXTENDED VERB ROUTINES THAT HAVE AT LEAST ONE FLASHING DISPLAY MUST TCF ENDMARK OR TCF ENDEXT WHEN
# FINISHED.

                BANK    10
                SETLOC  DISPLAYS
                BANK

                COUNT*  $$/DSPLA

# NTERONLY IS USED TO DIFFERENTIATE THE MARK ROUTINE WITH ONLY ONE RETURN TO THE USER FROM THE MARKING ROUTINE WITH
# 3 RETURNS TO THE USER.  THIS ROUTINE IS ONLY USED BY GOMARK1 AND GOMARK1R.

MARKEND         TC      CLEARMRK
                TCF     MARKOVER

GOMARK          TS      PLAYTEM1        # ENTRANCE FOR MARK GODSP
```

```
GOMARS          CAF     BIT15           # BIT15 SET FOR ALL MARK REQUESTS
                TCF     GOFLASH2

KLEENEX         CAF     ZERO            # CLEAN OUT EXTENDED VERBS
GOMARKF         TS      PLAYTEM1        # ENTRANCE FOR MARK GOFLASH

                CAF     MARKFMSK        # MARK, FLASH
                TCF     GOFLASH2

GOMARK2         TS      PLAYTEM1        # MARK GOPERFS-3 AST. RETURNS
MARKFORM        CAF     MPERFMSK        # MARK, PERFORM, FLASH
                TCF     GOFLASH2

GOMARK3         TS      PLAYTEM1        # USED FOR 3COMP DECIMAL PERFORM
                CAF     MARK3MSK
                TCF     GOFLASH2

GOMARK4         TS      PLAYTEM1
                CAF     MARK4MSK        # MARK,PERFORM,FLASH,BLANK
                TCF     GOFLASH2

GOMARKR         TS      PLAYTEM1        # ENTRANCE FOR MARK GODSPR

                CAF     BIT15
                TCF     GODSPR2

GOMARKFR        TS      PLAYTEM1        # ENTRANCE FOR MARK GOFLASHR

                CAF     MARKFMSK
                TCF     GODSPRS

GOMARK2R        TS      PLAYTEM1        # MARK GOPERFS-3 AST. RETS+ IMMEDIATE RET.
                CAF     MPERFMSK        # MARK, PERFORM, FLASH
                TCF     GODSPRS

GOMARK3R        TS      PLAYTEM1
                CAF     MARK3MSK
                TCF     GODSPRS

MAKEMARK        CAF     ONE
                TC      COPIES

                CA      FLAGWRD4        # IS NORM OR PRIO BUSY OR WAITING
                MASK    OCT34300
                CCS     A
                TCF     CHKPRIO

                CA      FLAGWRD4        # IS MARK SLEEPING DUE TO ASTRO BUSY
                MASK    MRKNVBIT

                EXTEND
```

```
                BZF     MARKPLAY        # NO
                TCF     ENDOFJOB

MARKPLAY        INHINT
                CS      FIVE            # RESET MARK OVER NORM, SET MARK
                MASK    FLAGWRD4
                AD      ONE
                TS      FLAGWRD4
                RELINT

GOGOMARK        CS      MARKFLAG        # PERFORM
                MASK    BIT5
                CCS     A
                TCF     MARKCOP
                CS      MARKNV
                TS      MARKNV

MARKCOP         CAF     ONE             # MARK INDEX
                TCF     PRIOPLAY

COPYTOGO        CA      MPAC2SAV
                TS      MPAC    +2

COPYPACS        INDEX   COPINDEX
                CAF     PRIOOCT
                TS      GENMASK

                INDEX   COPINDEX
                CAF     EBANKSAV
                TS      TEMPOR2         # ACTIVE EBANK AND FLAG

                TS      EBANK

                TC      Q

# PINCHEK CHECKS TO SEE IF THE CURRENT MARK REQUEST IS MADE BY THE ASTRONAUT WHILE INTERUPTING A GOPLAY DISPLAY
# (A NORMAL OR A PRIO).  IF THE ASTRONAUT TRIES TO MARK DURING A PRIO, THE CHECK FAIL LIGHT GOES ON AND THE MARK
# REQUEST IS ENDED.  IF HE TRIES TO MARK DURING A NORM, THE MARK IS ALLOWED.  IN THIS CASE THE NORM IS PUT TO SLEEP
# UNTIL ALL MARKING IS FINISHED.
#
# IF THE MARK REQUEST COMES FROM THE PROGRAM DURING A TIME THE ASTRONAUT IS NOT INTERRUPTING A NORMAL OR A
# PRIO, THE MARK REQUEST IS PUT TO SLEEP UNTIL THE PRESENT ACTIVE DISPLAY IS RESPONDED TO BY THE ASTRONAUT.

CHKPRIO         CA      FLAGWRD4        # MARK ATTEMPT DURING PRIO
                MASK    OCT24100
                CCS     A
                TCF     MARSLEEP

                CS      FLAGWRD4
```

```
                MASK    MKOVBIT         # SET MARK OVER NORM
                INHINT
                ADS     FLAGWRD4

                TCF     SETNORM

MARKPERF        CA      MARKNV
                MASK    VERBMASK
                TCF     NV50DSP

GODSP           TS      PLAYTEM1

GODSP2          CAF     ZERO
                TCF     GOFLASH2

GODSPRET        TS      PLAYTEM1        # ENTRANCE FOR A GODSP WITH A PASTE

                CAF     BIT6            # SET BIT6 TO GO BACK TO USER AFTER NVSUB
                TCF     GOFLASH2

GODSPR          TS      PLAYTEM1

GODSPR1         CAF     ZERO
GODSPR2         TS      PLAYTEM4

                CAF     ZERO            # * DON'T MOVE
                TCF     GODSPRS1

# CLEANDSP IS USED FOR CLEARING OUT A NORMAL DISPLAY THAT IS PRESENTLY ACTIVE OR A NORMAL DISPLAY THAT IS
# SET UP TO BE STARTED OR RESTARTED.
#
# NORMALLY THE USER WILL NOT NEED TO USE THIS ROUTINE SINCE A NEW NORMAL DISPLAY AUTOMATICALLY CLEARS OUT AN
# OLD DISPLAY.
#
# CALLING SEQUENCE FOR CLEANDSP --
#
#               TC      BANKCALL
#               CADR    CLEANDSP

CLEANDSP        CAF     ZERO
REFLASH         TS      PLAYTEM1

                CAF     REDOMASK        # FLASH AND PERMIT
                TCF     GOFLASH2

REFLASHR        TS      PLAYTEM1

                CAF     REDOMASK        # FLASH AND PERMIT
                TCF     GODSPRS
```

```
REGODSP         TS      PLAYTEM1

                CAF     BIT14
                TCF     GOFLASH2

REGODSPR        TS      PLAYTEM1

                CAF     BIT14
                TCF     GODSPR2

CLOCPLAY        TS      PLAYTEM1
                CAF     CLOCKCON
                TCF     GOFLASH2

GOFLASH         TS      PLAYTEM1

                CAF     BIT4            # LEAVE ONLY FLASH BIT SET
GOFLASH2        TS      PLAYTEM4

                TC      SAVELOCS

                RELINT

                TCF     MAKEPLAY        # BRANCH DIRECT WITH NO SEPARATE JOB CALL

PRIODSPR        TS      PLAYTEM1

                CAF     BITS7+4
                TCF     GODSPRS

PRIODSP         TS      PLAYTEM1

SETPRIO         CAF     BITS7+4
                TCF     GOFLASH2

MAKEPRIO        CAF     ZERO
                TS      COPINDEX

                TC      LINUSCHR
                TCF     HIPRIO          # LINUS RETURN
                CA      FLAGWRD4
                MASK    OCT20100        # IS PRIO IN ENDIDLE OR BUSY
                CCS     A
                TCF     PRIOBORT        # YES, ABORT

HIPRIO          CA      FLAGWRD4        # MARK ACTIVE
                MASK    OCT40400
                EXTEND
                BZF     ASKIFNRM        # NO
```

```
SETMARK         CAF     ZERO
                TCF     JOBXCHS

ASKIFNRM        CA      FLAGWRD4        # NORMAL ACTIVE
                MASK    OCT10200        # BITS 13+8
                EXTEND
                BZF     OKTOCOPY        # NO

SETNORM         CAF     ONE
                TCF     JOBXCHS

OKTOCOPY        TC      COPYNORM
                TC      WITCHONE

                TC      JOBWAKE

                TC      XCHTOEND

REDOPRIO        CA      TIME1           # SAVE TIME PRIODSP SENT OUT
                TS      PRIOTIME

KEEPPRIO        CAF     ZERO            # START UP PRIO DISPLAY
                TCF     PRIOPLAY

MAKEPLAY        CA      PRIORITY        # SAVE USER'S PRIORITY
                MASK    PRIO37
                TS      USERPRIO

                CAF     PRIO33          # RAISE PRIORITY FOR FAST JOBS AFTER WAKE
                TC      PRIOCHNG

                CA      PLAYTEM4        # IS IT MARK OR PRIO OR NORM
                MASK    BITS15+7
                CCS     A
                TCF     MAKEPRIO        # ITS PRIO
                TCF     IFLEGAL
                TCF     MAKEMARK        # ITS MARK

IFLEGAL         CAF     TWO
                TS      COPINDEX

                TC      LINUSCHR

                TCF     OKTOPLAY        # LINUS RETURN
                CS      EBANKTEM
                MASK    BIT4
                CCS     A
                TCF     OKTOPLAY        # NO

                CA      FLAGWRD4        # WAS NORM ASLEEP
```

```
                MASK    NBUSMASK        # ARE ANY NORMS ASLEEP
                EXTEND
                BZF     OKTOPLAY        # NO

PRIOBORT        TC      POODOO
                OCT     1502

OKTOPLAY        TC      COPIES2

                CA      USERPRIO
                EXTEND
                ROR     SUPERBNK
                TS      RESTREG

                CA      FLAGWRD4        # PRIO OR MARK GOING
                MASK    PMMASK
                CCS     A
                TCF     GOSLEEPS        # MARK GOING

                TCF     +2
                TCF     GOSLEEPS

# COULD PUT NORM BUSY CHECK HERE TO SAVE TIME

                TC      WITCHONE        # IS IT NVSUB BUSY, ENDIDLE OR NOONE
                TC      JOBWAKE

                TC      XCHTOEND

PLAYJUM1        CAF     TWO
PRIOPLAY        TS      COPINDEX

                TCF     GOPLAY

EXDSPRET        TS      PLAYTEM1

                CAF     BIT15+6
                TCF     GOFLASH2

GOPERF1         TS      NORMTEM1        # STORE DESIRED CHECKLIST VALUE
                CAF     V01N25          # USED TO DISPLAY CHECKLIST VALUE IN R1

GOPERFS         TS      PLAYTEM1

                CAF     PERFMASK        # LEAVE ONLY FLASH, PERFORM, BLANKING
                TCF     GOFLASH2

GOPERF2         TS      PLAYTEM1        # DESIRED VERB-NOUN TO DISPLAY R1,R2,R3

                CAF     PERF2MSK
                TCF     GOFLASH2
```

```
GOPERF4         TC      PURRS4

                TCF     GOFLASH2

GOFLASHR        TS      PLAYTEM1

                CAF     BIT4            # LEAVE ONLY FLASH BIT SET
GODSPRS         TS      PLAYTEM4

                CAF     THREE

GODSPRS1        INHINT                  # IMMEDIATE RETURN IS CALL CADR +4
                TS      RUPTREG3

                CA      PRIORITY        # MAKE DISPLAY ONE HIGHER THAN USER
                MASK    PRIO37
                TS      NEWPRIO

                CA      PLAYTEM4        # IS THIS A FLASHING R DISPLAY
                MASK    BIT4
                CCS     A
                TCF     VACDSP          # YES, MAKE DSPLAY JOB A VAC
                CA      NEWPRIO         # NO, MAKE DSPLAY JOB A NOVAC
                TC      NOVAC
                EBANK=  WHOCARES
                2CADR   MAKEPLAY

                TCF     BOTHJOBS

VACDSP          CA      BBANK
                EXTEND
                ROR     SUPERBNK
                TS      L
                CAF     MAKEGEN
                TC      SPVAC

BOTHJOBS        TC      SAVELOCS        # COPY TEMPS INTO PERMANENT REGISTERS

                EXTEND                  # SAVE NVWORD AND USER'S MPAC +2
                DCA     MPAC    +1
                INDEX   LOCCTR
                DXCH    MPAC    +1

                EXTEND                  # SAVE USER'S CADR, FLAGS AND EBANK
                DCA     MPAC    +3
                INDEX   LOCCTR
                DXCH    MPAC    +3

                CA      LOCCTR
                TS      MPAC    +5
```

```
                TC      SAVELOCR
                RELINT

                TCF     BANKJUMP        # CALL CADR +4

GOPERF1R        TS      NORMTEM1        # DESIRED CHECKLIST VALUE

                CAF     V01N25          # DISPLAYS CHECKLIST VALUE IN R1

GOPERFRS        TS      PLAYTEM1

                CAF     PERFMASK        # LEAVE ONLY FLASH, PERFORM, BLANKING
                TCF     GODSPRS

GOPERF2R        TS      PLAYTEM1        # DESIRED VERB-NOUN TO DISPLAY R1,R2,R3

                CAF     PERF2MSK
                TCF     GODSPRS

GOPERF4R        TC      PURRS4

                TCF     GODSPRS

PURRS4          TS      OPTION1         # DESIRED OPTION CODE

                CAF     V04N06
                TS      PLAYTEM1

                CAF     PERF4MSK        # FLASH, PERFORM AND EBANK R3
                TC      Q

SAVELOCS        INHINT

                CS      OCT3400         # EBANK BITS
                MASK    PLAYTEM4
                AD      EBANK
                TS      PLAYTEM4

SAVELOCR        LXCH    Q

                TC      MAKECADR
                TS      PLAYTEM3

                AD      RUPTREG3        # NOT USED FOR NON R ROUTINES
                TC      L

COPYNORM        CAF     ZERO
COPIES          TS      COPINDEX
COPIES2         INHINT
                CA      PLAYTEM4        # FLAGWORD
```

```
                INDEX   COPINDEX
                TS      EBANKSAV        # EQUIV TO DSPFLG

                MASK    CADRMASK        # FLASH AND GODSPRET
                EXTEND
                BZF     SKIPADD

                CA      PLAYTEM3
                INDEX   COPINDEX
                TS      CADRFLSH

SKIPADD         CA      PLAYTEM1        # VERB NOUN
                INDEX   COPINDEX
                TS      NVWORD

                TCF     RELINTQ

GOSLEEPS        INDEX   COPINDEX
                CA      PRIOOCT
                MASK    WAITMASK
                TC      UPENT2
WAITMASK        OCT     3004
                CS      ONE
                AD      COPINDEX
                TS      FACEREG

XCHSLEEP        INDEX   FACEREG
                CAF     WAKECADR
                INHINT
                TC      JOBWAKE         # FIND CADR IN JOB AREA

                TC      XCHTOEND        # CAUSES AWAKENED JOB TO GO TO ENDOFJOB

                INDEX   FACEREG         # REPLACE SAME CADR BUT NEW JOB AREA
                CAF     WAKECADR
                TCF     JOBSLEEP

JOBXCHS         TS      FACEREG         # CONTROLS TYPE OF DISPLAY PUT TO SLEEP
                TC      WITCHONE
                TC      JOBWAKE
                CA      FACEREG
                INDEX   LOCCTR
                TS      FACEREG

                CAF     XCHQADD
                TC      XCHNYLOC

                INDEX   FACEREG
                CA      MARKOCT
                MASK    IDLESLEP
```

```
                TC      DOWNENT2
IDLEMASK        OCT     74004           # * DON'T MOVE

                INDEX   FACEREG         # BIT SHOWS PRIO INTERRUPTED NORM OR MARK
                CA      BIT5            # BIT5 FOR MARK, BIT4 FOR NORMAL
                AD      FOUR
                TC      UPENT2          # FLAG ROUTINE DOES RELINT
XCHQADD         GENADR  XCHSLEEP        # * DON'T MOVE
                CA      FLAGWRD4
                MASK    MKOVBIT         # MARK OVER NORM?
                CCS     A
GENMARK         TC      MARKPLAY        # USED AS GENADR FOR JOBWAKE
                TCF     OKTOCOPY

MARKWAKE        CAF     ZERO
WAKEPLAY        TS      TEMPOR2

                INDEX   TEMPOR2
                CA      BITS5+11
                AD      FOUR
                TC      DOWNENT2
MARKFMSK        OCT     40010           # *** DON'T MOVE

                INDEX   TEMPOR2
                CAF     WAKECADR
                INHINT
                TC      JOBWAKE

                TCF     ENDRET

# ALL .1 RESTARTS BRANCH DIRECTLY TO INITDSP.  NORMAL DISPLAYS ARE THE ONLY DISPLAYS ALLOWED TO USE .1 RESTARTS
# INITDSP FIRST RESTORES THE EBANK AND THE SUPERBANK TO THE MOST RECENT NORMAL EBANK AND SUPERBANK.
#
# IF THE MOST RECENT NORMAL DISPLAY REQUEST WAS NOT FINISHED, CONTROL IS SENT BACK TO THE LAST NORMAL USER.
# OTHERWISE THE NORMAL DISPLAY SET UP IN THE NORMAL DISPLAY REGS IS STARTED UP IMMEDIATELY.

INITDSP         CA      EBANKTEM        # RESTORE MOST RECENT NORMAL EBANK
                TS      EBANK

                CA      RESTREG         # SUPERBANK AND JOB PRIORITY
                TC      SUPERSW         # RESTORE SUPERBANK

                MASK    PRIO37
                TC      PRIOCHNG

                CS      THREE
                AD      TEMPFLSH
                TCF     BANKJUMP

PINBRNCH        RELINT                  # FOR GOPIN USERS
                CA      MARK2PAC        # NEEDED TO SAVE MPAC +2 FOR MARK USERS
```

```
                TS      MPAC    +2      # ONLY

                CA      FLAGWRD4        # PINBRANCH CONDITION
                MASK    PINMASK
                CCS     A
                TCF     +3
                TCF     ERASER          # ** NOTHING IN ENDIDLE
                TCF     MARKPLAY

NORMBNCH        TC      UPFLAG          # SET PINBRANCH BIT
                ADRES   PINBRFLG

                CAF     PRIODBIT        # PRIO INTERRUPTED
                MASK    FLAGWRD4
                CCS     A
                TCF     KEEPPRIO

                TCF     PLAYJUM1

NVDSP           TC      COPYPACS

                CA      TEMPOR2         # SET UP BLANK BITS FOR NVMONOPT IN CASE
                MASK    SEVEN           # USER REQUESTS BLANKING MONITOR
                TS      L

                CS      BIT13
                INDEX   COPINDEX
                MASK    DSPFLG
                INDEX   COPINDEX
                TS      DSPFLG

                MASK    BIT8            # BIT8 SET IF DEC MARK PERFORM DISPLAY
                TS      TEM1

                CA      MPAC    +2
                TS      MPAC2SAV

                TS      MARK2PAC        # * FOR DISK ONLY *
                INDEX   COPINDEX
                CCS     NVWORD
                TCF     NVDSP1
                TCF     CLEANEND
                CS      MARKNV
                TS      MARKNV          # IN CASE MARKPLAY AWAKENED AFTER SLEEPING
                MASK    LOW7
                AD      VO5NOOM1
                AD      TEM1
NVDSP1          AD      ONE
NV50DSP         TC      NVMONOPT
                TCF     REST            # IF BUSY
```

```
                TC      FLASHOFF        # IN CASE OF EXTENDED VERB NON-FLASH

                TC      COPYTOGO        # MPACS DESTROYED BY NVSUB
                TC      DOWNFLAG        # UNSET SLEEPING BITS
                ADRES   MRKNVFLG
                TC      DOWNFLAG
                ADRES   NRMNVFLG
                TC      DOWNFLAG
                ADRES   PRONVFLG
BLANKCHK        CA      TEMPOR2         # BLANK BITS 1,2,3 IF SET
                TC      BLANKSUB
                TCF     NVDSP
PERFCHEK        CAF     BIT5            # BIT5 FOR PERFORM
                MASK    TEMPOR2
                CCS     A               # IS THIS A GOPERF DISPLAY
                TCF     1STOR2ND        # YES

GOANIDLE        CAF     BIT4
                MASK    TEMPOR2
                CCS     A
                TCF     FLASHSUB        # IT IS

                CS      TEMPOR2         # IS THIS A GODSPRET
                MASK    BIT6
                CCS     A
                TCF     ISITNOO

                INDEX   COPINDEX
                CA      CADRFLSH
                TS      MPAC    +3
                TCF     ENDIT

ISITNOO         INDEX   COPINDEX        # IS THIS A PASTE
                CA      NVWORD
                MASK    LOW7            # CHECK MADE FOR PINBRNCH AND PRIO ON MARK
                EXTEND
                BZF     FLASHSUB        # YES, ASSUME PASTE ALWAYS ON FLASH

                TCF     ENDOFJOB        # NOT FLASH, NOT GOPERF, THEREFORE EXIT

1STOR2ND        CA      TEMPOR2
                MASK    BIT13
                CCS     A
                TCF     GOANIDLE        # SECOND

                CA      BIT13
                INDEX   COPINDEX
                ADS     DSPFLG

                ZL
```

```
                EXTEND                          # IS IT MARK
                BZMF    MARKPERF                # YES

                MASK    BIT12
                EXTEND
                BZF     V50PASTE
                CS      NVWORD1                 # NVOWRD1= -0 IS V97.  NVWORD1= -400 IS V99
                AD      V97N00
                TCF     NV50DSP
V50PASTE        CAF     V50N00
                TCF     NV50DSP                 # DISPLAY SECOND PART OF GOPERF

WITCHONE        CS      BIT5                    # TURN OFF KEY RELEASE LIGHT
                EXTEND
                WAND    DSALMOUT

                CA      FLAGWRD4
                MASK    NVBUSMSK                # IS IT NVSUB ALEEP
                CCS     A
                CAF     ONE
                TS      L
                CAF     ZERO
                INDEX   L
                XCH     CADRSTOR

                INHINT
                TC      Q

XCHTOEND        CAF     ENDINST                 # TC ENDOFJOB REPLACES GENADR IN LOC FOR
XCHNYLOC        XCH     LOCCTR                  # WAS THIS ADDRESS SLEEPING
                EXTEND
                BZMF    RELINTQ                 # NO
                XCH     LOCCTR                  # YES
                INDEX   LOCCTR
                TS      LOC

RELINTQ         RELINT
                TC      Q                       # BACK TO USER

CLEANEND        CAF     PRIO32                  # ONE LOWER THAN DISPLAYS SLEEPING
                TC      FINDVAC
                EBANK=  NVSAVE
                2CADR   JAMTERM

                TCF     FLASHSUB +1

ISITPRIO        CA      FLAGWRD4
                MASK    ITISMASK                # IS PINBRFLG, MARKIDFLG SET
                EXTEND
```

```
                BZF     PRIOBORT
                TCF     ENDOFJOB

REST            CCS     CADRSTOR        # IS SOMEONE IN ENDIDLE
                TCF     ENDOFJOB        # YES
                TCF     RESTSLEP

                TCF     ENDOFJOB

RESTSLEP        CA      GENMASK         # SET NVSLEEP BITS
                MASK    ASTROMSK
                TC      UPENT2
OCT24100        OCT     24100           # *** DON'T MOVE

                INDEX   COPINDEX
                CAF     NVCADR
                TC      NVSUBUSY        # BUSY OR ABORT IF ILLEGAL

FLASHSUB        TC      FLASHON

                CA      COPINDEX        # COPINDEX DESTROYED BY ENDIDLE
                TS      COPMPAC

                CA      GENMASK
                MASK    IDLEMASK
                TC      UPENT2
ITISMASK        OCT     40040           # *** ENDIDLE ALLOW *** DON'T MOVE

                CA      R1SAVE          # IS THIS A REPEAT AND RETURN DISPLAY
                INDEX   COPINDEX
                MASK    BIT3
                CCS     A
                TCF     UNSETR1         # YES

                CCS     CADRSTOR        # SEE IF SOMEONE ALREADY IN ENDIDLE
                TCF     ISITPRIO
                TCF     +2
                TCF     ISITPRIO

                TC      ENDIDLE
IDLERET1        TCF     TERMATE

                TCF     PROCEED         # ENDIDLE RETURNS HERE ON PROCEED

                CS      LOWLOAD
                AD      MPAC            # VERBREG
                EXTEND
                DIM     A
                EXTEND
                BZF     LOADITIS        # V21 OR V22 OR V23 ON DSKY
```

```
OKTOENT         CAF     TWO
ENDOUT          TS      OUTHERE

                CA      FLAGWRD4        # CHECK NATURE OF ENDIDLE RETURN
                MASK    OCT60000
                CCS     A
                TCF     TIMECHEK        # PRIO ENDIDLE RETURN
                TCF     NORMRET         # NORMAL ENDIDLE RETURN
                TCF     MARKRET         # MARK ENDIDLE RETURN

TIMECHEK        CS      TIME1
                AD      PRIOTIME
                CCS     A
                COM
                AD      OCT37776
                AD      ONE
                AD      -2SEC
                EXTEND
                BZMF    KEEPPRIO

                TCF     NORMRET

NORMWAKE        CAF     ONE
                TCF     WAKEPLAY

ENDRET          CCS     OUTHERE
                AD      ONE
                TCF     +2              # NORMAL ENDIDLE EXIT
                TCF     ENDOFJOB
                INDEX   COPMPAC
                AD      CADRFLSH
                TS      MPAC    +3

                CA      GENMASK         # REMOVE ENDIDLE AND PINBRANCH BITS
                MASK    PINIDMSK
                TC      DOWNENT2
PINIDMSK        OCT     74044           # *** DON'T MOVE

                CS      THREE           # BLANK EVERYTHING EXCEPT MM
                TC      NVSUB
                TCF     +1

ENDIT           CA      USERPRIO        # RETURN TO USER'S PRIORITY
                MASK    PRIO37
                TC      PRIOCHNG
                CA      MPAC    +3
                TCF     BANKJUMP

UNSETR1         INDEX   COPINDEX        # RESET REPEAT AND RETURN REQUEST
                CS      BIT3
```

```
                MASK    R1SAVE
                TS      R1SAVE

                CAF     ZERO            # *** 205 ONLY MARKBRAN USERS IN
                TC      SUPERSW         # SUPERBANK 0

 -1             CAF     THREE           # RETURN TO USER'S IMMEDIATE RETURN LOC
IMMEDRET        INDEX   COPINDEX
                AD      CADRFLSH
                TCF     BANKJUMP

TERMATE         CAF     ZERO            # ASTRONAUT TERMINATE (V34) RETURNS TO
                TCF     ENDOUT

LINUSCHR        CS      PLAYTEM4        # IS THIS A LINUS
                MASK    BIT14
                CCS     A
                TCF     Q+1             # NO
                CS      PLAYTEM3        # YES, IS IT ALREADY IN ENDIDLE
                INDEX   COPINDEX
                AD      CADRFLSH
                EXTEND
                BZF     +2              # YES

                TC      Q               # NO
                CCS     DSPLOCK         # IS THE ASTRONAUT BUSY
                TC      ENDOFJOB        # END THE NEW DISPLAY, IT'S ALREADY ACTIVE
                TC      Q

# MORE LOGIC COULD BE INCORPORATED HERE TO MAKE SURE A RECYCLE IS A RECYCLE AND CONVERSELY THAT A LOAD IS A LOAD

PROCEED         CAF     ONE             # ASTRONAUT PROCEED (V33) RETURNS
                TCF     ENDOUT

# LASTPLAY CHECKS TO SEE IF (1) THE LAST NORMAL DISPLAY WAS EITHER INTERRUPTED BY A PRIO OR A MARK (MARK
# COULD ONLY HAPPEN DURING PINBRANCH) OR IF (2) THE LAST NORMAL DISPLAY WAS REQUESTED WHILE A HIGHER PRIORITY
# DISPLAY WAS GOING, RESULTING IN THE NORMAL BEING PUT TO SLEEP.
#
# IF EITHER OF THE ABOVE 2 CONDITIONS EXISTS, THE NORMAL DISPLAY IS AWAKENED TO GO TO PLAYJUM1 WHICH STARTS
# UP THE MOST RECENT VALID NORMAL DISPLAY.  IF THESE 2 CONDITIONS DO NOT EXIST, CONTROL GOES TO PLAYJUM1 WHICH IS
# STARTED IMMEDIATELY WITH THE ASSUMPTION THAT THE MOST RECENT NORMAL DISPLAY IS ALREADY IN ENDIDLE (DURING A
# PINBRNCH) OR THAT A RESTART HAS OCCURRED AND THE DISPLAY CAN BE STARTED AS A .1 RESTART.

MARKRET         CS      SIX
                MASK    FLAGWRD4
                INHINT                  # *** MAY MOVE DISPLAY FLAGWORD OUT OF
                TS      FLAGWRD4

                RELINT                  # INHINT REALM
                TCF     ENDRET
```

```
MARKOVER        CAF     MINUS1          # RUPTREG2 IS - MEANS ENDOFJOB TO ENDRET
                TS      OUTHERE

                CA      FLAGWRD4        # IS ENDIDFLG SET
                MASK    PRIO30          # IS NORMAL OR PRIO IN ENDIDLE
                CCS     A
                TCF     NORMBNCH

NORMRET         CA      FLAGWRD4        # IS MARK SLEEPING
                MASK    BITS5+11        # OR WAITING
                CCS     A
                TCF     MARKWAKE

                CA      FLAGWRD4        # NO
                MASK    BITS4+10        # IS NORMAL INTERRUPTED OR WAITING
                CCS     A
                TCF     NORMWAKE        # YES

                CA      EBANKTEM        # NO, WAS IT A FLASH REQUEST
                MASK    OCT50           # OR A GODSPRET
                CCS     A
                TCF     ENDRET          # YES
                CA      NVSAVE
                EXTEND
                BZF     ENDRET

                CAF     PRIO15
                INHINT
                TC      NOVAC
                EBANK=  NVWORD
                2CADR   PLAYJUM1

                TCF     ENDRET

MARSLEEP        CA      FLAGWRD4        # IS MARK ALREADY IN
                MASK    BITS5+11
                CCS     A
                TCF     ENDOFJOB        # YES
                TCF     GOSLEEPS

LOADITIS        INDEX   COPMPAC
                CA      NVWORD
                MASK    LOW7
                COM
                AD      MPAC    +1      # NOUNREG
                EXTEND
                BZF     OKTOENT         # NO, THEN LOAD IS VALID
                TCF     PINBRNCH        # YES, ACCEPT LOAD BUT ASK FOR LAST AGAIN
```

```
ERASER          CS      THREE           # BLANK EVERYTHING EXCEPT MM
                TC      NVSUB
                TCF     ENDOFJOB
                TCF     ENDOFJOB

PERFMASK        OCT     0036            # FLASH, PERFORM, BLANK R2 AND R3
V01N25          VN      00125
V06N07          VN      00607           # GOPERF3 VN DISPLAY BEFORE V50
V50N00          VN      5000
PERF2MSK        OCT     00030           # FLASH, PERFORM
V04N06          VN      00406
PERF4MSK        OCT     14              # FLASH, BLANK R3
GOAGIN          EQUALS  PINBRNCH
REDOMASK        OCT     20010           # BITS 4 AND 14
MARK3MSK        OCT     40230           # MARK, DECIMAL NOUN, PERFORM, FLASH
MARK4MSK        OCT     40036           # MARK, PERFORM, FLASH, BLANK 2 AND 3
NVCADR          CADR    REDOPRIO
WAKECADR        CADR    MARKPLAY
                CADR    PLAYJUM1

OCT3400         OCT     3400            # EBANK MASK
NBUSMASK        OCT     11210
PMMASK          OCT     66521
VERBMASK        =       MID7            # (OCT 37600)
V05N00M1        OCT     1177            # V05 MINUS ONE
GOXDSP          EQUALS  GOMARK
GOXDSPR         EQUALS  GOMARKR
GOXDSPF         EQUALS  GOMARKF
GOXDSPFR        EQUALS  GOMARKFR
ENDEXT          EQUALS  ENDMARK
MPAC2SAV        EQUALS  BANKSET
NVBUSMSK        OCT     700
ASTROMSK        OCT     704
MPERFMSK        OCT     40030           # BIT 15,5,4 FOR MARK,PERFORM,FLASH
OCT34300        OCT     34300
BITS15+7        OCT     40100
BITS7+4         OCT     110
DSPFLG          EQUALS  EBANKSAV
MARKFLAG        EQUALS  MARKEBAN
SAVEFLAG        EQUALS  EBANKTEM
BITS5+11        OCT     2020            # * DON'T MOVE
BITS4+10        OCT     1010            # * DON'T MOVE
LOWLOAD         DEC     22
BUSYMASK        OCT     77730
CADRMASK        OCT     50
PINMASK         EQUALS  13,14,15
GOPLAY          EQUALS  NVDSP
PRIOSAVE        EQUALS  R1SAVE
COPMPAC         EQUALS  MPAC    +3
TEMPOR2         EQUALS  MPAC    +4
```

```
OUTHERE         EQUALS  MPAC    +5
COPINDEX        EQUALS  LOC
USERPRIO        EQUALS  MODE
GENMASK         EQUALS  MPAC    +6
PRIOOCT         OCT     20144           # PRIO
MARKOCT         OCT     42424           # MARK
                OCT     11254           # NORM

IDLESLEP        OCT     74704
OCT67777        OCT     67777
LINUS           EQUALS  BLANKET
FACEREG         EQUALS  MPAC
PLAYTEM1        EQUALS  MPAC    +1
PLAYTEM3        EQUALS  MPAC    +3
PLAYTEM4        EQUALS  MPAC    +4
OCT40420        OCT     40420
MAKEGEN         GENADR  MAKEPLAY
OCT10200        OCT     10200
V97N00          VN      09700           # PASTE FOR V97 OR V99
OCT20100        OCT     20100
CLOCKCON        OCT     24030           # FLASH, PERFORM, V99 OR V97 PASTE, REFLASH
```

```
                BANK    10
                SETLOC  DISPLAYS
                BANK
                COUNT*  $$/DSPLA

UPENT2          INHINT
                MASK    OCT77770
                TS      L
                CS      FLAGWRD4
                MASK    L
                ADS     FLAGWRD4
JOIN            RELINT
                TCF     Q+1

DOWNENT2        INHINT
                MASK    OCT77770
                COM
                MASK    FLAGWRD4
                TS      FLAGWRD4
                TCF     JOIN

OCT7            EQUALS  SEVEN
```

```
#     UPFLAG AND DOWNFLAG ARE ENTIRELY GENERAL FLAG SETTING AND CLEARING SUBROUTINES.   USING THEM, WHETHER OR
# NOT IN INTERRUPT, ONE MAY SET OR CLEAR ANY SINGLE, NAMED BIT IN ANY ERASABLE REGISTER, SUBJECT OF COURSE TO
# EBANK SETTING.   A "NAMED" BIT, AS THE WORD IS USED HERE, IS ANY BIT WITH A NAME FORMALLY ASSIGNED BY THE YUL
# ASSEMBLER.
#
#     AT PRESENT THE ONLY NAMED BITS ARE THOSE IN THE FLAGWORDS.   ASSEMBLER CHANGES WILL MAKE IT POSSIBLE TO
# NAME ANY BIT IN ERASABLE MEMORY.
#
#     CALLING SEQUENCES ARE AS FOLLOWS :-
#               TC      UPFLAG                   TC      DOWNFLAG
#               ADRES   NAME OF FLAG             ADRES   NAME OF FLAG
#
#     RETURN IS TO THE LOCATION FOLLOWING THE "ADRES" ABOUT .58 MS AFTER THE "TC".
#     UPON RETURN A CONTAINS THE CURRENT FLAGWRD SETTING.

                BLOCK   02
                SETLOC  FFTAG1
                BANK
                COUNT*  $$/FLAG

UPFLAG          CA      Q
                TC      DEBIT
                COM                     # +(15 - BIT)
                EXTEND
                ROR     LCHAN           # SET BIT
COMFLAG         INDEX   ITEMP1
                TS      FLAGWRD0
                LXCH    ITEMP3
                RELINT
                TC      L

DOWNFLAG        CA      Q
                TC      DEBIT
                MASK    L               # RESET BIT
                TCF     COMFLAG

DEBIT           AD      ONE             # CET DE BITS
                INHINT
                TS      ITEMP3
                CA      LOW4            # DEC15
                TS      ITEMP1
                INDEX   ITEMP3
                CA      0 -1            # ADRES
                TS      L
                CA      ZERO
```

```
                EXTEND
                DV      ITEMP1          # A = FLAGWRD, L = (15 - BIT)
                DXCH    ITEMP1
                INDEX   ITEMP1
                CA      FLAGWRD0
                TS      L               # CURRENT STATE
                INDEX   ITEMP2
                CS      BIT15           # -(15 - BIT)
                TC      Q
```

```
# DELAYJOB- A GENERAL ROUTINE TO DELAY A JOB A SPECIFIC AMOUNT OF TIME BEFORE PICKING UP AGAIN.
#
# ENTRANCE REQUIREMENTS...
#               CAF     DT              # DELAY JOB FOR DT CENTISECS
#               TC      BANKCALL
#               CADR    DELAYJOB

                BANK    06
                SETLOC  DLAYJOB
                BANK

# THIS MUST REMAIN IN BANK 0 *****************************************

                COUNT*  $$/DELAY
DELAYJOB        INHINT
                TS      Q               # STORE DELAY DT IN Q FOR DLY -1 IN
                CAF     DELAYNUM        # WAITLIST
DELLOOP         TS      RUPTREG1
                INDEX   A
                CA      DELAYLOC        # IS THIS DELAYLOC AVAILABLE
                EXTEND
                BZF     OK2DELAY        # YES

                CCS     RUPTREG1        # NO, TRY NEXT DELAYLOC
                TCF     DELLOOP

                DXCH    BUF2
                TC      BAILOUT1        # NO AVAILABLE LOCS.
                OCT     1104

OK2DELAY        CA      TCSLEEP         # SET WAITLIST IMMEDIATE RETURN
                TS      WAITEXIT

                CA      FBANK
                AD      RUPTREG1        # STORE BBANK FOR TASK CALL
                TS      L

                CAF     WAKECAD         # STORE CADR FOR TASK CALL
                TCF     DLY2 -1         # DLY IS IN WAITLIST ROUTINE

TCGETCAD        TC      MAKECADR        # GET CALLERS FCADR

                INDEX   RUPTREG1
                TS      DELAYLOC        # SAVE DELAY CADRS

                TC      JOBSLEEP

WAKER           CAF     ZERO
                INDEX   BBANK
                XCH     DELAYLOC        # MAKE DELAYLOC AVAILABLE
```

```
                TC      JOBWAKE

                TC      TASKOVER

TCSLEEP         GENADR  TCGETCAD -2
WAKECAD         GENADR  WAKER
```

```
# GENTRAN, A BLOCK TRANSFER ROUTINE.
# WRITTEN BY D. EYLES
# MOD 1 BY KERNAN                                      UTILITYM REV 17 11/18/67
# MOD 2 BY SCHULENBERG (REMOVE RELINT)   SKIPPER REV 4 2/28/68
#
#          THIS ROUTINE IS USEFULL FOR TRANSFERING N CONSECUTIVE ERASABLE OR FIXED QUANTITIES TO SOME OTHER N
# CONSECUTIVE ERASABLE LOCATIONS.  IF BOTH BLOCKS OF DATA ARE IN SWITCHABLE EBANKS, THEY MUST BE IN THE SAME ONE.
#
#          GENTRAN IS CALLABLE IN A JOB AS WELL AS A RUPT.  THE CALLING SEQUENCE IS:
#       I       CA      N-1             # # OF QUANTITIES MINUS ONE.
#       I +1    TC      GENTRAN         # IN FIXED-FIXED.
#       I +2    ADRES   L               # STARTING ADRES OF DATA TO BE MOVED.
#       I +3    ADRES   M               # STARTING ADRES OF DUPLICATION BLOCK.
#       I +4                            # RETURNS HERE.
#
#          GENTRAN TAKES 25 MCT'S (300 MICROSECONDS) PER ITEM + 5 MCT'S (60 MICS) FOR ENTERING AND EXITING.
#          A, L, AND ITEMP1 ARE NOT PRESERVED.

                BLOCK   02
                SETLOC  FFTAG4
                BANK

                EBANK=  ITEMP1

                COUNT*  $$/TRAN

GENTRAN         INHINT
                TS      ITEMP1          # SAVE N-1.
                INDEX   Q               # C(Q) = ADRES L.
                AD      0               # ADRES (L + N - 1).
                INDEX   A
                CA      0               # C(ABOVE).
                TS      L               # SAVE DATA.
                CA      ITEMP1
                INDEX   Q
                AD      1               # ADRES (M + N - 1).
                INDEX   A
                LXCH    0               # STUFF IT.
                CCS     ITEMP1          # LOOP UNTIL N-1 = 0.
                TCF     GENTRAN +1
                TCF     Q+2             # RETURN TO CALLER.
```

```
# B5OFF   ZERO BIT 5 OF EXTVBACT, WHICH IS SET BY TESTXACT.
#          MAY BE USED AS NEEDED BY ANY EXTENDED VERB WHICH HAS DONE TESTXACT

                COUNT*  $$/EXTVB

B5OFF           CS      BIT5
                MASK    EXTVBACT
                TS      EXTVBACT
                TC      ENDOFJOB
```

```
# THE FOLLOWING SUBROUTINE MAY BE CALLED TO DISPLAY A NON-ABORTIVE ALARM CONDITION.  IT MAY BE CALLED
# EITHER IN INTERRUPT OR UNDER EXECUTIVE CONTROL.
#
# CALLING SEQUENCE IS AS FOLLOWS:
#               TC      ALARM
#               OCT     AAANN           # ALARM NO. NN IN GENERAL AREA AAA.
#                                       # (RETURNS HERE)

                BLOCK   02
                SETLOC  FFTAG7
                BANK

                EBANK=  FAILREG

                COUNT*  $$/ALARM

# ALARM TURNS ON THE PROGRAM ALARM LIGHT, BUT DOES NOT DISPLAY.

ALARM           INHINT

                CA      Q
ALARM2          TS      ALMCADR
                INDEX   Q
                CA      0
BORTENT         TS      L

PRIOENT         CA      BBANK
 +1             EXTEND
                ROR     SUPERBNK        # ADD SUPER BITS.
                TS      ALMCADR +1

LARMENT         CA      Q               # STORE RETURN FOR ALARM
                TS      ITEMP1

CHKFAIL1        CCS     FAILREG         # IS ANYTHING IN FAILREG
                TCF     CHKFAIL2        # YES TRY NEXT REG
                LXCH    FAILREG
                TCF     PROGLARM        # TURN ALARM LIGHT ON FOR FIRST ALARM

CHKFAIL2        CCS     FAILREG +1
                TCF     FAIL3
                LXCH    FAILREG +1
                TCF     MULTEXIT

FAIL3           CA      FAILREG +2
                MASK    POSMAX
                CCS     A
                TCF     MULTFAIL
                LXCH    FAILREG +2
                TCF     MULTEXIT
```

```
PROGLARM        CS      DSPTAB +11D
                MASK    OCT40400
                ADS     DSPTAB +11D

MULTEXIT        XCH     ITEMP1          # OBTAIN RETURN ADDRESS IN A
                RELINT
                INDEX   A
                TC      1

MULTFAIL        CA      L
                AD      BIT15
                TS      FAILREG +2

                TCF     MULTEXIT

# PRIOLARM DISPLAYS V05N09 VIA PRIODSPR WITH 3 RETURNS TO THE USER FROM THE ASTRONAUT AT CALL LOC +1,+2,+3 AND
# AN IMMEDIATE RETURN TO THE USER AT CALL LOC +4.  EXAMPLE FOLLOWS,
#               CAF     OCTXX           # ALARM CODE
#               TC      BANKCALL
#               CADR    PRIOLARM
#               ...     ...
#               ...     ...
#               ...     ...             # ASTRONAUT RETURN
#               TC      PHASCHNG        # IMMEDIATE RETURN TO USER.  RESTART
#               OCT     X.1             # PHASE CHANGE FOR PRIO DISPLAY

                BANK    10
                SETLOC  DISPLAYS
                BANK

                COUNT*  $$/DSPLA
PRIOLARM        INHINT                  # * * * KEEP IN DISPLAY ROUTINES BANK
                TS      L               # SAVE ALARM CODE

                CA      BUF2            # 2 CADR OF PRIOLARM USER
                TS      ALMCADR
                CA      BUF2 +1
                TC      PRIOENT +1      # * LEAVE L ALONE
-2SEC           DEC     -200            # *** DONT MOVE
                CAF     V05N09
                TCF     PRIODSPR

                BLOCK   02
                SETLOC  FFTAG7
                BANK

                COUNT*  $$/ALARM
BAILOUT         INHINT
                CA      Q
```

```
                TS      ALMCADR

                INDEX   Q
                CAF     0
                TC      BORTENT
OCT40400        OCT     40400

                INHINT
WHIMPER         CA      TWO
                AD      Z
                TS      BRUPT
                RESUME
                TC      POSTJUMP        # RESUME SENDS CONTROL HERE
                CADR    ENEMA
POODOO          INHINT
                CA      Q
ABORT2          TS      ALMCADR
                INDEX   Q
                CAF     0
                TC      BORTENT
OCT77770        OCT     77770           # DON'T MOVE

                CAF     OCT35           # 4.35SPOT FOR GOPOODOO
                TS      L
                COM
                DXCH    -PHASE4
GOPOODOO        INHINT
                TC      BANKCALL        # RESET STATEFLG, REINTFLG, AND NODOFLAG.
                CADR    FLAGS
                CA      FLAGWRD7        # IS SERVICER CURRENTLY IN OPERATION?
                MASK    V37FLBIT
                CCS     A
                TCF     STRTIDLE
                TC      BANKCALL        # TERMINATE GRPS 1, 3, 5, AND 6
                CADR    V37KLEAN
                TC      BANKCALL        # TERMINATE GRPS 2, 4, 1, 3, 5, AND 6
                CADR    MR.KLEAN        #        (I.E., GRP 4 LAST)
                TCF     WHIMPER
STRTIDLE        CAF     BBSERVDL
                TC      SUPERSW
                TC      BANKCALL        # PUT SERVICER INTO ITS "GROUND" STATE
                CADR    SERVIDLE        # AND PROCED TO GOTOPOOH.
CCSHOLE         INHINT
                CA      Q
                TC      ABORT2
OCT1103         OCT     1103
CURTAINS        INHINT
                CA      Q
                TC      ALARM2
OCT217          OCT     00217
```

```
                TC      ALMCADR         # RETURN TO USER

BAILOUT1        INHINT
                DXCH    ALMCADR
                CAF     ADR40400
BOTHABRT        TS      ITEMP1
                INDEX   Q
                CAF     0
                TS      L
                TCF     CHKFAIL1
POODOO1         INHINT
                DXCH    ALMCADR
                CAF     ADR77770
                TCF     BOTHABRT

ALARM1          INHINT
                DXCH    ALMCADR
ALMNCADR        INHINT
                INDEX   Q
                CA      0
                TS      L
                TCF     LARMENT

ADR77770        TCF     OCT77770
ADR40400        TCF     OCT40400
DOALARM         EQUALS  ENDOFJOB
                EBANK=  DVCNTR
BBSERVDL        BBCON   SERVIDLE

# CALLING SEQUENCE FOR VARALARM
#               CAF     (ALARM)
#               TC      VARALARM
#
# VARALARM TURNS ON PROGRAM ALARM LIGHT BUT DOES NOT DISPLAY

VARALARM        INHINT

                TS      L               # SAVE USERS ALARM CODE

                CA      Q               # SAVE USERS Q
                TS      ALMCADR

                TC      PRIOENT
OCT14           OCT     14              # DONT MOVE

                TC      ALMCADR         # RETURN TO USER

ABORT           EQUALS  WHIMPER
                BANK    13
                SETLOC  ABTFLGS
                BANK
```

```
                    COUNT*  $$/ALARM

FLAGS               CS      STATEBIT
                    MASK    FLAGWRD3
                    TS      FLAGWRD3
                    CS      REINTBIT
                    MASK    FLGWRD10
                    TS      FLGWRD10
                    CS      NODOBIT
                    MASK    FLAGWRD2
                    TS      FLAGWRD2
                    TC      Q
```

```
# PROGRAM NAME:         P27
# WRITTEN BY:           KILROY/ DE WOLF
#
# MOD NO:               6
# MOD BY:               KILROY
# DATE:                 01DEC67
#
# LOG SECTION:          UPDATE PROGRAM.
#
# FUNCT. DESCR.:        P27 (THE UPDATE PROGRAM) PROCESSES COMMANDS AND DATA
#                       INSERTIONS REQUESTED BY THE GROUND VIA UPLINK.
#                       THE P27 PROGRAM WILL ACCEPT UPDATES
#                       ONLY DURING P00 FOR THE LM, AND ONLY DURING P00,
#                       P02, AND FRESH START FOR THE CSM
#
# CALLING SEQ:          PROGRAM IS INITIATED BY UPLINK ENTRY OF VERBS 70, 71, 72, AND 73.
#
# SUBROUTINES:          TESTXACT, NEWMODEX, NEWMODEX +3, GOXDSPF, BANKCALL, FINDVAC, INTPRET, INTSTALL, TPAGREE,
#                       INTWAKEU, ENDEXT, POSTJUMP, FALTON, NEWPHASE, PHASCHNG
#
# NORMAL EXIT:          TC ENDEXT
#
# ALARM/ABORT:          TC FALTON FOLLOWED BY TC ENDEXT
#
# RESTARTS:             P27 IS RESTART PROTECTED IN TWO WAYS ...
#                       1.      PRIOR TO VERIFLAG INVERSION (WHICH IS CAUSED BY THE GROUND/ASTRONAUT'S VERIFICATION OF UPDATE
#                               DATA BY SENDING A V33E WHEN V21N02 IS FLASHING)---
#                               NO PROTECTION EXCEPT PRE-P27 MODE IS RESTORED, COAST + ALIGN DOWNLIST IS SELECTED AND UPLINK
#                               ACTIVITY LIGHT IS TURNED OFF.  (JUST AS IF A V34E WAS SENT DURING P27 DATA LOADS).
#                               V70,V71,V72, OR V73 WILL HAVE TO BE COMPLETELY RESENT BY USER.
#                       2.      AFTER VERIFLAG INVERSION (WHEN UPDATE OF THE SPECIFIED ERASABLES IS BEING PERFORMED)---
#                               PROTECTED AGAINST RESTARTS.
#
# DEBRIS:               UPBUFF  (20D)   TEMP STORAGE FOR ADDRESSES AND CONTENTS.
#                       UPVERB  (1)     VERB NUMBER MINUS 70D (E.G., FOR V72, UPVERB = 72D - 70D = 2)
#                       UPOLDMOD(1)     FOR MAJOR MODE INTERRUPTED BY P27.
#                       COMPNUMB(1)     TOTAL NUMBER OF COMPONENTS TO BE TRANSMITTED.
#                       UPCOUNT (1)     ACTUAL NUMBER OF COMPONENTS RECEIVED.
#                       UPTEMP  (1)     SCRATCH, BUT USUALLY CONTAINS COMPONENT NUMBER TO BE CHANGED DURING VERIFY CYCLE
#
# INPUT:
#
#       ENTRY                   DESCRIPTION
#       V70EXXXXXEXXXXXE        (LIFTOFF TIME INCREMENT) DOUBLE PRECISION OCTAL TIME INCREMENT, XXXXX XXXXX,
#                               IS ADDED TO TEPHEM, SUBTRACTED FROM AGC CLOCK(TIME2,TIME1), SUBTRACTED FROM CSM STATE
#                               VECTOR TIME(TETCSM) AND SUBTRACTED FROM LEM STATE VECTOR TIME(TETLEM).
#                               THE DP OCTAL TIME INCREMENT IS SCALED AT 2(28).
```

```
#       V71EIIEAAAAE            (CONTIGUOUS BLOCK UPDATE) II-2 OCTAL COMPONENTS, XXXXX,
#       XXXXXE                  ARE LOADED INTO ERASABLE STARTING AT ECADR, AAAA.
#       XXXXXE                  IT IS .GE. 3 .AND. .LE. 200.,
#       ...                     AND (AAAA + II -3) DOES NOT PRODUCE AN ADDRESS IN THE
#                               NEXT BANK.
#                               SCALING IS SAME AS INTERNAL REGISTERS.
#       V72EIIE                 (SCATTER UPDATE) (II-1)/2 OCTAL COMPONENTS, XXXXX, ARE
#       AAAAEXXXXXE             LOADED INTO ERASABLE LOCATIONS, AAAA.
#       AAAAEXXXXXE             II IS .GE. 3 .AND. .LE. 19D, AND MUST BE ODD.
#       ...                     SCALING IS SAME AS INTERNAL REGISTERS.
#
#       V73EXXXXXEXXXXXE        (OCTAL CLOCK INCREMENT) DOUBLE PRECISION OCTAL TIME
#                               INCREMENT XXXXX XXXXX, IS ADDED TO THE AGC CLOCK, IN
#                               CENTISECONDS SCALED AT (2)28).
#                               THIS LOAD IS THE OCTAL EQUIVALENT OF V55.
#
# OUTPUT:               IN ADDITION TO THE ABOVE REGISTER LOADS, ALL UPDATES
#                       COMPLEMENT BIT3 OF FLAGWORD7.
#
# ADDITIONAL NOTES:     VERB 71, JUST DEFINED ABOVE WILL BE USED TO PERFORM BUT NOT LIMITED TO THE FOLLOWING UPDATES --
#                       1.  CSM/LM STATE VECTOR UPDATE
#                       2.  REFSMMAT UPDATE
#
#       THE FOLLOWING COMMENTS DELINEATE EACH SPECIAL UPDATE ---
#
#       1.  CSM/LM STATE VECTOR UPDATE (ALL DATA ENTRIES IN OCTAL)
#           ENTRIES:            DATA DEFINITION:                                SCALE FACTORS:
#           V71E                CONTIGUOUS BLOCK UPDATE VERB
#              21E              NUMBER OF COMPONENTS FOR STATE VECTOR UPDATE
#            AAAAE              ECADR OF `UPSVFLAG'
#           XXXXXE              STATE VECTOR IDENTIFIER: 00001 FOR CSM, 77776 FOR LEM -- EARTH SPHERE OF INFLUENCE SCALING
#                                                        00002 FOR CSM, 77775 FOR LEM -- LUNAR SPHERE OF INFLUENCE SCALING
#           XXXXXEXXXXXE        X POSITION
#           XXXXXEXXXXXE        Y POSITION
#           XXXXXEXXXXXE        Z POSITION
#           XXXXXEXXXXXE        X VELOCITY
#           XXXXXEXXXXXE        Y VELOCITY
#           XXXXXEXXXXXE        Z VELOCITY
#           XXXXXEXXXXXE        TIME FROM AGC CLOCK ZERO
#           V33E                VERB 33 TO SIGNAL THAT THE STATE VECTOR IS READY TO BE STORED.
#
#       2.  REFSMMAT (ALL DATA ENTRIES IN OCTAL)
#           ENTRIES:            DATA DEFINITITIONS:                             SCALE FACTORS:
```

```
#               V71E            CONTIGUOUS BLOCK UPDATE VERB
#                  24E          NUMBER OF COMPONENTS FOR REFSMMAT UPDATE
#                AAAAE          ECADR OF `REFSMMAT'
#               XXXXXEXXXXXE    ROW 1 COLUMN 1                                  2(-1)
#               XXXXXEXXXXXE    ROW 1 COLUMN 2                                  2(-1)
#               XXXXXEXXXXXE    ROW 1 COLUMN 3                                  2(-1)
#               XXXXXEXXXXXE    ROW 2 COLUMN 1                                  2(-1)
#               XXXXXEXXXXXE    ROW 2 COLUMN 2                                  2(-1)
#               XXXXXEXXXXXE    ROW 2 COLUMN 3                                  2(-1)
#               XXXXXEXXXXXE    ROW 3 COLUMN 1                                  2(-1)
#               XXXXXEXXXXXE    ROW 3 COLUMN 2                                  2(-1)
#               XXXXXEXXXXXE    ROW 3 COLUMN 3                                  2(-1)
#               V33E            VERB 33 TO SIGNAL THAT REFSMMAT IS READY TO BE STORED


                BANK    07
                SETLOC  EXTVERBS
                BANK

                EBANK=  TEPHEM

                COUNT*  $$/P27
V70UPDAT        CAF     UP70            # COMES HERE ON V70E
                TCF     V73UPDAT +1

V71UPDAT        CAF     UP71            # COMES HERE ON V71E
                TCF     V73UPDAT +1

V72UPDAT        CAF     UP72            # COMES HERE ON V72E
                TCF     V73UPDAT +1

V73UPDAT        CAF     UP73            # COMES HERE ON V73E

 +1             TS      UPVERBSV        # SAVE UPVERB UNTIL IT'S OK TO ENTER P27

                TC      TESTXACT        # GRAB DISPLAY IF AVAILABLE, OTHERWISE
                                        # TURN*OPERATOR ERROR* ON AND TERMINATEJOB

                CA      MODREG          # CHECK IF UPDATE ALLOWED
                EXTEND                  # FIRST CHECK FOR MODREG = +0, -0
                BZF     +3              # (+0 = P00, -0 = FRESHSTART)
UPERROR         TC      POSTJUMP        # TURN ON 'OPERATOR ERROR' LIGHT
                CADR    UPERROUT +2     # GO TO COMMON UPDATE PROGRAM EXIT

                CAE     MODREG          # UPDATE ALLOWED.
CKMDMORE        =       UPERROR
                TS      UPOLDMOD        # SAVE CURRENT MAJOR MODE
```

```
                CAE     UPVERBSV        # SET UPVERB TO INDICDATE TO P27
                TS      UPVERB          # WHICH EXTENDED VERB CALLED IT.

                CAF     ONE
                TS      UPCOUNT         # INITIALIZE UPCOUNT TO 1

                TC      POSTJUMP        # LEAVE EXTENDED VERB BANK AND
                CADR    UPPART2         # GO TO UPDATE PROGRAM (P27) BANK.

UP70            EQUALS  ZERO
UP71            EQUALS  ONE
UP72            EQUALS  TWO
UP73            EQUALS  THREE

                BANK    04
                SETLOC  UPDATE2
                BANK

                COUNT*  $$/P27

UPPART2         EQUALS                  # UPDATE PROGRAM -- PART 2

                TC      PHASCHNG        # SET RESTART GROUP 6 TO RESTORE OLD MODE
                OCT     07026           # AND DOWNLIST AND EXIT IF RESTART OCCURS.
                OCT     30000           # PRIORITY SAME AS CHRPRIO
                SBANK=  PINSUPER
                EBANK=  UPBUFF
                2CADR   UPOUT +1

                CAF     ONE
                TS      DNLSTCOD        # DOWNLIST

                TC      NEWMODEX        # SET MAJOR MODE = 27
                DEC     27

                INDEX   UPVERB          # BRANCH DEPENDING ON WHETHER THE UPDATE
                TCF     +1              # VERB REQUIRES A FIXED OR VARIABLE NUMBER
                TCF     +3              # V70 FIXED                (OF COMPONENTS)
                TCF     OHWELL1         # V71 VARIABLE -- GO GET NO. OF COMPONENTS
                TCF     OHWELL1         # V72 VARIABLE -- GO GET NO. OF COMPONENTS
                CA      TWO             # V73 (AND V70) FIXED
                TS      COMPNUMB        # SET NUMBER OF COMPONENTS TO 2.
                TCF     OHWELL2         # GO GET THE TWO UPDATE COMPONENTS

OHWELL1         CAF     ADUPBUFF        # * REQUEST USER TO SEND NUMBER *
                TS      MPAC +2         # * OF COMPONENTS PARAMETER(II).*
 +2             CAF     UPLOADNV        # (CKV432 RETURNS HERE IF V32 ENCOUNTERED)
                TC      BANKCALL        # DISPLAY A FLASHING V21N01
```

```
                CADR    GOXDSPF         # TO REQUEST II.

                TCF     UPOUT4          # V34 TERMINATE UPDATE (P27) RETURN
                TCF     OHWELL1 +2
                TC      CK4V32          # DATA OR V32 RETURN
                CS      BIT2
                AD      UPBUFF          # IS II (NUMBER OF COMPONENTS PARAMETER)
                EXTEND                  # .GE. 3 AND .LE. 20D.
                BZMF    OHWELL1 +2
                CS      UPBUFF
                AD      UP21
                EXTEND
                BZMF    OHWELL1 +2
                CAE     UPBUFF
                TS      COMPNUMB        # SAVE II IN COMPNUMB

# UPBUFF LOADING SEQUENCE

                INCR    UPCOUNT         # INCREMENT COUNT OF COMPONENTS RECEIVED.
OHWELL2         CAF     ADUPBFM1        # CALCULATE LOCATION (ECADR) IN UPBUFF
                AD      UPCOUNT         # WHERE NEXT COMPONENT SHOULD BE STORED
 +2             TS      MPAC +2         # PLACE ECADR INTO R3.
 +3             CAF     UPLOADNV        # (CK4V32 RETURNS HERE IF V32 ENCOUNTERED)
                TC      BANKCALL        # DISPLAY A FLASHING V21N01
                CADR    GOXDSPF         # TO REQUEST DATA.
                TCF     UPOUT4          # V34 TERMINATE UPDATE (P27) RETURN.
                TCF     OHWELL2 +3      # V33 PROCEED RETURN
                TC      CK4V32          # DATA OR V32 RETURN
                CS      UPCOUNT         # HAVE WE FINISHED RECEIVING ALL
                AD      COMPNUMB        # THE DATA WE EXPECTED.
                EXTEND
                BZMF    UPVERIFY        # YES -- GO TO VERIFICATION SEQUENCE
                TCF     OHWELL2 -1      # NO -- REQUEST ADDITIONAL DATA.

# VERIFY SEQUENCE

UPVERIFY        CAF     ADUPTEMP        # PLACE ECADR WHERE COMPONENT NO. INDEX
                TS      MPAC +2         # IS TO BE STORED INTO R3.
                CAF     UPVRFYNV        # (CK4V32 RETURNS HERE IF V32 ENCOUNTERED)
                TC      BANKCALL        # DISPLAY A FLASHING V21N02 TO REQUEST
                CADR    GOXDSPF         # DATA CORRECTION OR VERIFICATION.
                TCF     UPOUT4          # V34 TERMINATE UPDATE (P27) RETURN
                TCF     UPSTORE         # V33 DATA SENT IS GOOD.  GO STORE IT.
                TC      CK4V32          # COMPONENT NO. INDEX OR V32 RETURN
                CA      UPTEMP          # DOES THE COMPONENT NO. INDEX JUST SENT
                EXTEND                  # SPECIFY A LEGAL COMPONENT NUMBER?
                BZMF    UPVERIFY        # NO, IT IS NOT POSITIVE NONZERO
                CS      UPTEMP
                AD      COMPNUMB
```

```
                AD      BIT1
                EXTEND
                BZMF    UPVERIFY        # NO
                CAF     ADUPBFM1        # YES -- BASED ON THE COMPONENT NO. INDEX
                AD      UPTEMP          # CALCULATE THE ECADR OF LOCATION IN
                TCF     OHWELL2 +2      # UPBUFF WHICH USER WANTS TO CHANGE.

UPOUT4          EQUALS  UPOUT +1        # COMES HERE ON V34 TO TERMINATE UPDATE

# CHECK FOR VERB 32 SEQUENCE

CK4V32          CS      MPAC            # ON DATA RETURN FROM `GOXDSPF'
                MASK    BIT6            # ON DATA RETURN FROM "GOXDSP" & THE CON-
                CCS     A               # TENTS OF MPAC = VERB.  SO TEST FOR V32.
                TC      Q               # IT'S NOT A V32, IT'S DATA.  PROCEED.
                INDEX   Q
                TC      0 -6            # V32 ENCOUNTERED -- GO BACK AND GET DATA

ADUPTEMP        ADRES   UPTEMP          # ADDRESS OF TEMP STORAGE FOR CORRECTIONS
ADUPBUFF        ADRES   UPBUFF          # ADDRESS OF UPDATE DATA STORAGE BUFFER
UPLOADNV        VN      2101            # VERB 21 NOUN 01
UPVRFYNV        VN      2102            # VERB 21 NOUN 02
UP21            =       MD1             # DEC 21 = MAX NO OF COMPONENTS +1
UPDTPHAS        EQUALS  FIVE

# PRE-STORE AND FAN TO APPROPRIATE BRANCH SEQUENCE

UPSTORE         EQUALS                  # GROUND HAS VERIFIED UPDATE.  STORE DATA.

                INHINT

                CAE     FLAGWRD7        # INVERT VERIFLAG (BIT 3 OF FLAGWRD7) TO
                XCH     L               # INDICATE TO THE GROUND (VIA DOWNLINK)
                CAF     VERIFBIT        # THAT THE V33 (WHICH THE GROUND SENT TO
                EXTEND                  # VERIFY THE UPDATE) HAS BEEN SUCCESSFULLY
                RXOR    LCHAN           # RECEIVED BY THE UPDATE PROGRAM
                TS      FLAGWRD7

                TC      PHASCHNG        # SET RESTART GROUP 6 TO REDO THE UPDATE
                OCT     04026           # DATA STORE IF A RESTART OCCURS.
                INHINT                  # (BECAUSE PHASCHNG DID A RELINT)

                CS      TWO             # GO TO UPFNDVAC IF INSTALL IS REQUIRED.
                AD      UPVERB          # THAT IS, IF IT'S A V70 - V72.
                EXTEND                  # GO TO UPEND73 IF IT'S A V73.
                BZMF    UPFNDVAC

# VERB 73 BRANCH
```

```
UPEND73         EXTEND                          # V73 -- PERFORM DP OCTAL AGC CLOCK INCREMENT

                DCA     UPBUFF
                DXCH    UPBUFF +8D
                TC      TIMEDIDL
                TC      FALTON                  # ERROR -- TURN ON *OPERATOR ERROR* LIGHT
                TC      UPOUT +1                # GO TO COMMON UPDATE PROGRAM EXIT

UPFNDVAC        CAF     CHRPRIO                 # (USE EXTENDED VERB PRIORITY)
                TC      FINDVAC                 # GET VAC AREA FOR 'CALL INTSTALL'
                EBANK=  TEPHEM
                2CADR   UPJOB                   # (NOTE:  THIS WILL ALSO SET EBANK FOR
                TC      ENDOFJOB                # 'TEPHEM' UPDATE BY V70)

UPJOB           TC      INTPRET                 # THIS COULD BE A STATE VECTOR UPDATE -- SO
                CALL                            # WAIT (PUT JOB TO SLEEP) IF ORBIT INT(OI)
                        INTSTALL                # IS IN PROGRESS -- OR -- GRAB OI AND RETURN
                                                # TO UPWAKE IF OI IS NOT IN PROGRESS.

UPWAKE          EXIT

                TC      PHASCHNG                # RESTART PROTECT (GROUP 6)
                OCT     04026

                TC      UPFLAG                  # SET INTEGRATION RESTART BIT
                ADRES   REINTFLG
                INHINT
UPPART3         EQUALS

                INDEX   UPVERB                  # BRANCH TO THE APPROPRIATE UPDATE VERB
                TCF     +1                      # ROUTINE TO ACTUALLY PERFORM THE UPDATE
                TCF     UPEND70                 # V70
                TCF     UPEND71                 # V71
                TCF     UPEND72                 # V72

# ROUTINE TO INCREMENT CLOCK (TIME2,TIME1) WITH CONTENTS OF DP WORD AT UPBUFF.

TIMEDIDL        EXTEND
                QXCH    UPTEMP                  # SAVE Q FOR RETURN
                CAF     ZERO                    # ZERO AND SAVE TIME2,TIME1
                ZL
                DXCH    TIME2
                DXCH    UPBUFF +18D             # STORE IN CASE OF OVERFLOW

                CAF     UPDTPHAS                # SO
                TS      L                       # A
                COM                             # QUICK
                DXCH    -PHASE6                 # PHASCHNG
```

```
TIMEDIDR        INHINT

                CAF     ZERO
                ZL                      # PICK UP INCRMENTER (AND ZERO
                TS      MPAC +2         # IT IN CASE OF RESTARTS) AND
                DXCH    UPBUFF +8D      # STORE IT
                DXCH    MPAC            # INTO MPAC FOR TPAGREE.

                EXTEND
                DCA     UPBUFF +18D
                DAS     MPAC            # FORM SUM IN MPAC
                EXTEND
                BZF     DELTAOK         # TEST FOR OVERFLOW
                CAF     ZERO
                DXCH    UPBUFF +18D     # OVERFLOW, RESTORE OLD VALUE OF CLOCK
                DAS     TIME2           # AND TURN ON OPERATOR ERROR

                TC      PHASCHNG        # RESTART PROTECT (GROUP 6)
                OCT     04026

                TC      UPTEMP          # GO TO ERROR EXIT

DELTAOK         TC      TPAGREE         # FORCE SIGN AGREEMENT
                DXCH    MPAC
                DAS     TIME2           # INCREMENT TIME2,TIME1

                TC      PHASCHNG        # RESTART PROTECT (GROUP 6)
                OCT     04026

                INHINT
                INDEX   UPTEMP          # (CODED THIS WAY FOR RESTART PROTECTION)
                TC      1               # NORMAL RETURN

# VERB 71 BRANCH

UPEND71         CAE     UPBUFF +1       # SET EBANK
                TS      EBANK           #       AND
                MASK    LOW8            # CALCULATE
                TS      UPTEMP          # S-REG VALUE OF RECEIVING AREA

                AD      NEG3            # IN THE PROCESS OF
                AD      COMPNUMB        # PERFORMING
                EXTEND                  # THIS UPDATE
                BZF     STORLP71        # WILL WE
                MASK    BIT9            # OVERFLOW
                CCS     A               # INTO THE NEXT EBANK....
                TCF     UPERROUT        # YES

                CA      NEG3            # NO -- CALCULATE NUMBER OF
                AD      COMPNUMB        # WORDS TO BE STORED MINUS ONE
STORLP71        TS      MPAC            # SAVE NO. OF WORDS REMAINING MINUS ONE
```

```
                INDEX   A               # TAKE NEXT UPDATE WORD FROM
                CA      UPBUFF +2       # UPBUFF AND

                TS      L               # SAVE IT IN L
                CA      MPAC            # CALCULATE NEXT
                AD      UPTEMP          # RECEIVING ADDRESS
                INDEX   A
                EBANK=  1400
                LXCH    1400            # UPDATE THE REGISTER BY CONTENTS OF L
                EBANK=  TEPHEM
                CCS     MPAC            # ARE THERE ANY WORDS LEFT TO BE STORED
                TCF     STORLP71        # YES
                TCF     UPOUT           # NO -- THEN EXIT UPDATE PROGRAM
ADUPBFM1        ADRES   UPBUFF -1       # SAME AS ADUPBUFF BUT LESS 1 (DON'T MOVE)
                TCF     UPOUT           # NO -- EXIT UPDATE (HERE WHEN COMPNUMB = 3)

# VERB 72 BRANCH

UPEND72         CAF     BIT1            # HAVE AN ODD NO. OF COMPONENTS
                MASK    COMPNUMB        # BEEN SENT FOR A V72 UPDATE ...
                CCS     A
                TCF     +2              # YES
                TCF     UPERROUT        # ERROR -- SHOULD BE ODD NO. OF COMPONENTS
                CS      BIT2
                AD      COMPNUMB
LDLOOP72        TS      MPAC            # NOW PERFORM THE UPDATE
                INDEX   A
                CAE     UPBUFF +1       # PICK UP NEXT UPDATE WORD
                LXCH    A
                CCS     MPAC            # SET POINTER TO ECADR (MUST BE CCS)
                TS      MPAC
                INDEX   A
                CAE     UPBUFF +1       # PICK UP NEXT ECADR OF REG TO BE UPDATED
                TS      EBANK           # SET EBANK
                MASK    LOW8            # ISOLATE RELATIVE ADDRESS
                INDEX   A
                EBANK=  1400
                LXCH    1400            # UPDATE THE REGISTER BY CONTENTS OF L
                EBANK=  TEPHEM
                CCS     MPAC            # ARE WE THORUGH THE V72 UPDATE...
                TCF     LDLOOP72        # NO

# NORMAL FINISH OF P27

UPOUT           EQUALS
                TC      INTWAKEU        # RELEASE GRAB OF ORBITAL INTEGRATION
 +1             CAE     UPOLDMOD        # RESTORE PRIOR P27 MODE
                TC      NEWMODEX +3
                CAF     ZERO
```

```
                TS      DNLSTCOD
                TC      UPACTOFF        # TURN OFF 'UPLINK ACTIVITY' LIGHT

                EXTEND                  # KILL GROUP 6.
                DCA     NEGO
                DXCH    -PHASE6

                TC      ENDEXT          # EXTENDED VERB EXIT

# VERB TO BRANCH

UPEND70         EXTEND                  # V70 DOES THE FOLLOWING WITH DP DELTA
                DCS     UPBUFF          # TIME IN UPBUFF
                DXCH    UPBUFF +8D
                TC      TIMEDIDL        # DECREMENT AGC CLOCK

                TC      UPERROUT        # ERROR WHILE DECREMENTING CLOCK -- EXIT

                EBANK=  TEPHEM
                EXTEND
                DCS     UPBUFF          # COPY DECREMENTERS FOR
                DXCH    UPBUFF +10D     # RESTART PROTECTION
                EXTEND
                DCS     UPBUFF
                DXCH    UPBUFF +12D

                TC      PHASCHNG        # RESTART PROTECT (GROUP 6)
                OCT     04026

                CAF     ZERO
                ZL
                DXCH    UPBUFF +10D     # DECREMENT CSM STATE VECTOR TIME
                DAS     TETCSM

                CAF     ZERO
                ZL
                DXCH    UPBUFF +12D     # DECREMENT LEM STATE VECTOR TIME
                DAS     TETLEM

                CAF     ZERO
                ZL
                DXCH    UPBUFF
                DAS     TEPHEM +1       # INCREMENT TP TEPHEM
                ADS     TEPHEM

                TC      PHASCHNG        # RESTART PROTECT (GROUP 6)
                OCT     04026

                EBANK=  UPBUFF
```

```
                TC      UPOUT           # GO TO STANDARD UPDATE PROGRAM EXIT

# ERROR SEQUENCE

UPERROUT        TC      FALTON          # TURN ON *OPERATOR ERROR* LIGHT
                TCF     UPOUT           # GO TO COMMON UPDATE PROGRAM EXIT

 +2             TC      FALTON          # TURN ON 'OPERATOR ERROR' LIGHT
                TC      UPACTOFF        # TURN OFF 'UPLINK ACTIVITY' LIGHT
                TC      ENDEXT          # EXTENDED VERB EXIT
                                        # (THE PURPOS OF UPERROUT +2 EXIT IS
                                        # TO PROVIDE AN ERROR EXIT WHICH DOES NOT
                                        # RESET ANY RESTART GROUPS)

# 'UPACTOFF' IS A ROUTINE TO TURN OFF UPLINK ACTIVITY LIGHT ON ALL EXITS FROM UPDATE PROGRAM (P27).

UPACTOFF        CS      BIT3
                EXTEND                  # TURN OFF UPLINK ACTIVITY LIGHT
                WAND    DSALMOUT        # (BIT 3 OF CHANNEL 11)
                TC      Q
```

```
                BANK    22
                SETLOC  RTBCODES
                BANK

                EBANK=  XNB
                COUNT*  $$/RTB

# LOAD TIME2, TIME1 INTO MPAC:

LOADTIME        EXTEND
                DCA     TIME2
                TCF     SLOAD2

#           CONVERT THE SINGLE PRECISION 2'S COMPLEMENT NUMBER ARRIVING IN MPAC (SCALED IN HALF-REVOLUTIONS) TO A
# DP 1'S COMPLEMENT NUMBER SCALED IN REVOLUTIONS.

CDULOGIC        CCS     MPAC
                CAF     ZERO
                TCF     +3
                NOOP
                CS      HALF

                TS      MPAC +1
                CAF     ZERO
                XCH     MPAC
                EXTEND
                MP      HALF
                DAS     MPAC
                TCF     DANZIG          # MODE IS ALREADY AT DOUBLE-PRECISION

#           FORCE TP SIGN AGREEMENT IN MPAC:

SGNAGREE        TC      TPAGREE
                TCF     DANZIG

#           CONVERT THE DP 1'S COMPLEMENT ANGLE SCALED IN REVOLUTIONS TO A SINGLE PRECISION 2'S COMPLEMENT ANGLE
# SCALED IN HALF-REVOLUTIONS.

1STO2S          TC      1TO2SUB
                CAF     ZERO
                TS      MPAC +1
                TCF     NEWMODE

#           DO 1STO2S ON A VECTOR OF ANGLES:

V1STO2S         TC      1TO2SUB         # ANSWER ARRIVES IN A AND MPAC.

                DXCH    MPAC +5
                DXCH    MPAC
                TC      1TO2SUB
```

```
                TS      MPAC +2

                DXCH    MPAC +3
                DXCH    MPAC
                TC      1TO2SUB
                TS      MPAC +1

                CA      MPAC +5
                TS      MPAC

TPMODE          CAF     ONE             # MODE IS TP.
                TCF     NEWMODE

#           V1STO2S FOR 2 COMPONENT VECTOR. USED BY RR.

2V1STO2S        TC      1TO2SUB
                DXCH    MPAC +3
                DXCH    MPAC
                TC      1TO2SUB
                TS      L
                CA      MPAC +3
                TCF     SLOAD2

#           SUBROUTINE TO DO DOUBLING AND 1'S TO 2'S CONVERSION:

1TO2SUB         DXCH    MPAC            # FINAL MPAC +1 UNSPECIFIED.
                DDOUBL
                CCS     A
                AD      ONE
                TCF     +2
                COM                     # THIS WAS REVERSE OF MSU.

                TS      MPAC            # AND SKIP ON OVERFLOW.
                TC      Q

                INDEX   A               # OVERFLOW UNCORRECT AND IN MSU.
                CAF     LIMITS
                ADS     MPAC
                TC      Q

#           THE FOLLOWING ROUTINE INCREMENTS IN 2S COMPLEMENT THE REGISTER WHOSE ADDRESS IS IN BUF BY THE 1S COMPL.
# QUANTITY FOUND IN TEM2. THIS MAY BE USED TO INCREMENT DESIRED IMU AND OPTICS CDU ANGLES OR ANY OTHER 2S COMPL.
# (+0 UNEQUAL TO -0) QUANTITY. MAY BE CALLED BY BANKCALL/SWCALL.

CDUINC          TS      TEM2            # 1S COMPL.QUANT. ARRIVES IN ACC. STORE IT
                INDEX   BUF
                CCS     0               # CHANGE 2S COMPL. ANGLE(IN BUF)INTO 1S
                AD      ONE
                TCF     +4
                AD      ONE
```

```
		AD	ONE		# OVERFLOW HERE IF 2S COMPL. IS 180 DEG.
		COM

		AD	TEM2		# SULT MOVES FROM 2ND TO 3D QUAD.(OR BACK)
		CCS	A		# BACK TO 2S COMPL.
		AD	ONE
		TCF	+2
		COM
		TS	TEM2		# STORE 14BIT QUANTITY WITH PRESENT SIGN
		TCF	+4
		INDEX	A		#  SIGN.
		CAF	LIMITS		# FIX IT,BY ADDING IN 37777 OR 40000
		AD	TEM2

		INDEX	BUF
		TS	0		# STORE NEW ANGLE IN 2S COMPLEMENT.
		TC	Q
```

```
#          RTB TO TORQUE GYROS, EXCEPT FOR THE CALL TO IMUSTALL. ECADR OF COMMANDS ARRIVES IN X1.

PULSEIMU        INDEX   FIXLOC          # ADDRESS OF GYRO COMMANDS SHOULD BE IN X1
                CA      X1
                TC      BANKCALL
                CADR    IMUPULSE
                TCF     DANZIG
```

```
#          THE SUBROUTINE SIGNMPAC  SETS C(MPAC, MPAC +1) TO SIGN(MPAC).
# FOR THIS, ONLY THE CONTENTS OF MPAC ARE EXAMINED.   ALSO +0 YIELDS POSMAX AND -0 YIELDS NEGMAX.
#
# ENTRY MAY BE BY EITHER OF THE FOLLOWING:
# 1.       LIMIT THE SIZE OF MPAC ON INTERPRETIVE OVERFLOW:
# ENTRY:          BOVB
#                        SIGNMPAC
# 2.       GENERATE IN MPAC THE SIGNUM FUNCTION OF MPAC:
# ENTRY:          RTB
#                        SIGNMPAC
# IN EITHER CASE, RETURN IS TO  THE NEXT INTERPRETIVE INSTRUCTION IN THE CALLING SEQUENCE.

SIGNMPAC        EXTEND
                DCA     DPOSMAX
                DXCH    MPAC
                CCS     A
DPMODE          CAF     ZERO            # SETS MPAC +2 TO ZERO IN THE PROCESS
                TCF     SLOAD2 +2
                TCF     +1
                EXTEND
                DCS     DPOSMAX
                TCF     SLOAD2

#     RTB OP CODE NORMUNIT IS LIKE INTERPRETIVE INSTRUCTION UNIT, EXCEPT THAT IT CAN BE DEPENDED ON NOT TO BLOW
# UP WHEN THE VECTOR BEING UNITIZED IS VERY SMALL -- IT WILL BLOW UP WHEN ALL COMPONENTS ARE ZERO.   IF NORMUNIT
# IS USED AND THE UPPER ORDER HALVES OF ALL COMPONENTS ARE ZERO, THE MAGNITUDE RETURNED IN 36D WILL BE TOO LARGE
# BY A FACTOR OF 2(13) AND THE SQUARED MAGNITUDE RETURNED AT 34D WILL BE TOO BIG BY A FACTOR OF 2(26).

NORMUNX1        CAF     ONE
                TCF     NORMUNIT +1
NORMUNIT        CAF     ZERO
                AD      FIXLOC
                TS      MPAC +2
                TC      BANKCALL        # GET SIGN AGREEMENT IN ALL COMPONENTS
                CADR    VECAGREE
                CCS     MPAC
                TCF     NOSHIFT
                TCF     +2
                TCF     NOSHIFT
                CCS     MPAC +3
                TCF     NOSHIFT
                TCF     +2
                TCF     NOSHIFT
                CCS     MPAC +5
                TCF     NOSHIFT
                TCF     +2
                TCF     NOSHIFT
```

```
                CA      MPAC +1         # SHIFT ALL COMPONENTS LEFT 13
                EXTEND
                MP      BIT14
                DAS     MPAC            # DAS GAINS A LITTLE ACCURACY
                CA      MPAC +4
                EXTEND
                MP      BIT14
                DAS     MPAC +3
                CA      MPAC +6
                EXTEND
                MP      BIT14
                DAS     MPAC +5
                CAF     THIRTEEN
                INDEX   MPAC +2
                TS      37D
OFFTUNIT        TC      POSTJUMP
                CADR    UNIT +1         # SKIP THE "TC VECAGREE" DONE AT UNIT

NOSHIFT         CAF     ZERO
                TCF     OFFTUNIT -2

# RTB VECSGNAG   ...FORCES SIGN AGREEMENT OF VECTOR IN MPAC.

VECSGNAG        TC      BANKCALL
                CADR    VECAGREE
                TC      DANZIG

# *** END OF SKIPPER .087 ***
```

```
# PROGRAM NAMES:         (1) T6JOBCHK    MOD. NO. 5      OCTOBER 2, 1967
#                        (2) DOT6RUPT
# MODIFICATION BY:       LOWELL G. HULL (A.C.ELECTRONICS)
#
# THESE PROGRAMS ENABLE THE LM DAP TO CONTROL THE THRUST TIMES OF THE REACTION CONTROL SYSTEM JETS BY USING TIME6.
# SINCE THE LM DAP MAINTAINS EXCLUSIVE CONTROL OVER TIME6 AND ITS INTERRUPTS, THE FOLLOWING CONVENTIONS HAVE BEEN
# ESTABLISHED AND MUST NOT BE TAMPERED WITH:
#       1.      NO NUMBER IS EVER PLACED INTO TIME6 EXCEPT BY LM DAP.
#       2.      NO PROGRAM OTHER THAN LM DAP ENABLES THE TIME6 COUNTER.
#       3.      TO USE TIME6, THE FOLLOWING SEQUENCE IS ALWAYS EMPLOYED:
#               A.      A POSITIVE (NON-ZERO) NUMBER IS STORED IN TIME6.
#               B.      THE TIME6 CLOCK IS ENABLED.
#               C.      TIME6 IS INTERROGATED AND IS:
#                       I.      NEVER FOUND NEGATIVE (NON-ZERO) OR +0.
#                       II.     SOMETIMES FOUND POSITIVE (BETWEEN 1 AND 240D) INDICATING THAT IT IS ACTIVE.
#                       III.    SOMETIMES FOUND POSMAX INDICATING THAT IT IS INACTIVE AND NOT ENABLED.
#                       IV.     SOMETIMES FOUND NEGATIVE ZERO INDICATING THAT:
#                               A.      A T6RUPT IS ABOUT TO OCCUR AT THE NEXT DINC, OR
#                               B.      A T6RUPT IS WAITING IN THE PRIORITY CHAIN, OR
#                               C.      A T6RUPT IS IN PROCESS NOW.
#       4.      ALL PROGRAMS WHICH OPERATE IN EITHER INTERRUPT MODE OR WITH INTERRUPT INHIBITED MUST CALL T6JOBCHK
#               EVERY 5 MILLISECONDS TO PROCESS A POSSIBLE WAITING T6RUPT BEFORE IT CAN BE HONORED BY THE HARDWARE.
#      (5.      PROGRAM JTLST, IN Q,R-AXES, HANDLES THE INPUT LIST.)
#
# T6JOBCHK CALLING SEQUENCE:
#               L       TC      T6JOBCHK
#               L+1     (RETURN)
#
# DOT6RUPT CALLING SEQUENCE:
#                       DXCH    ARUPT           # T6RUPT LEAD IN AT LOCATION 4004.
#                       EXTEND
#                       DCA     T6ADR
#                       DTCB
#
# SUBROUTINES CALLED:   DOT6RUPT CALLS T6JOBCHK.
#
# NORMAL EXIT MODES:    T6JOBCHK RETURNS TO L +1.
#                       DOT6RUPT TRANSFERS CONTROL TO RESUME.
#
# ALARM/ABORT MODES:    NONE.
#
# INPUT:        TIME6           NXT6ADR         OUTPUT:         TIME6           NXT6ADR         CHANNEL 5
#               T6NEXT          T6NEXT +1                       T6NEXT          T6NEXT +1       CHANNEL 6
#               T6FURTHA        T6FURTHA +1                     T6FURTHA        T6FURTHA +1     BIT15/CH13
#
# DEBRIS:       T6JOBCHK CLOBBERS A.  DOT6RUPT CLOBBERS NOTHING.

                BLOCK   02
```

```
                BANK    17
                SETLOC  DAPS2
                BANK
                EBANK=  T6NEXT
                COUNT*  $$/DAPT6

T6JOBCHK        CCS     TIME6           # CHECK TIME6 FOR WAITING T6RUPT:
                TC      Q               # NONE: CLOCK COUNTING DOWN.
                TC      CCSHOLE
                TC      T6JOBCHK +3

# CONTROL PASSES TO T6JOB ONLY WHEN C(TIME6) = -0 (I.E., WHEN A T6RUPT MUST BE PROCESSED).

T6JOB           CAF     POSMAX          # DISABLE CLOCK: NEEDED SINCE RUPT OCCURS
                EXTEND                  # 1 DINC AFTER T6 = 77777. FOR 625 MUSECS
                WAND    CHAN13          # MUST NOT HAVE T6 = +0 WITH ENABLE SET

                CA      POSMAX
                ZL
                DXCH    T6FURTHA
                DXCH    T6NEXT
                LXCH    NXT6ADR
                TS      TIME6

                AD      PRIO37
                TS      A
                TCF     ENABLET6
                CA      POSMAX
                TS      TIME6
                TCF     GOCH56
ENABLET6        CA      BIT15
                EXTEND
                WOR     CHAN13
                CA      T6NEXT
                AD      PRIO37
                TS      A
                TCF     GOCH56
                CA      POSMAX
                TS      T6NEXT
GOCH56          INDEX   L
                TCF     WRITEP -1

                BLOCK   02
                SETLOC  FFTAG9
                BANK
                EBANK=  CDUXD
                COUNT*  $$/DAPT6

                CA      NEXTP
WRITEP          EXTEND
                WRITE   CHAN6
```

```
                TC      Q

                CA      NEXTU
WRITEU          TS      L
                CS      00314OCT
                EXTEND
                RAND    CHAN5
                AD      L
                EXTEND
                WRITE   CHAN5
                TC      Q

                CA      NEXTV
WRITEV          TS      L
                CA      00314OCT
                TCF     -9D
00314OCT        OCT     00314

                BANK    17
                SETLOC  DAPS2
                BANK

                EBANK=  T6NEXT
                COUNT*  $$/DAPT6

DOT6RUPT        LXCH    BANKRUPT        # (INTERRUPT LEAD INS CONTINUED)
                EXTEND
                QXCH    QRUPT

                TC      T6JOBCHK        # CALL T6JOBCHK.

                TCF     RESUME          # END TIME6 INTERRUPT PROCESSOR.
```

```
                BANK    20
                SETLOC  DAPS3
                BANK

                EBANK=  CDUXD
                COUNT*  $$/DAPIF

# MOD 0         DATE    11/15/66        BY GEORGE W. CHERRY
# MOD 1                  1/23/67        MODIFICATION BY PETER ADLER
#
# FUNCTIONAL DESCRIPTION
#       HEREIN ARE A COLLECTION OF SUBROUTINES WHICH ALLOW MISSION CONTROL PROGRAMS TO CONTROL THE MODE
#       AND INTERFACE WITH THE DAP.
#
# CALLING SEQUENCES
#       IN INTERRUPT OR WITH INTERRUPT INHIBITED
#               TC      IBNKCALL
#               FCADR   ROUTINE
#       IN A JOB WITHOUT INTERRUPT INHIBITED
#               INHINT
#               TC      IBNKCALL
#               FCADR   ROUTINE
#               RELINT
#
# OUTPUT
#       SEE INDIVIDUAL ROUTINES BELOW
#
# DEBRIS
#       A, L, AND SOMETIMES MDUETEMP                    ODE     NOT IN PULSES MODE
```

```
# SUBROUTINE NAMES:
#       SETMAXDB, SETMINDB, RESTORDB, PFLITEDB
# MODIFIED:     30 JANUARY 1968 BY P S WEISSMAN TO CREATE RESTORDB.
# MODIFIED:     1 MARCH 1968 BY P S WEISSMAN TO SAVE EBANK AND CREATE PFLITEDB
#
# FUNCTIONAL DESCRIPTION:
#       SETMAXDB -- SET DEADBAND TO 5.0 DEGREES
#       SETMINDB -- SET DEADBAND TO 0.3 DEGREE
#       RESTORDB -- SET DEADBAND TO MAX OR MIN ACCORDING TO SETTINGS OF DBSELECT BIT OF DAPBOOLS
#       PFLITEDB -- SET DEADBAND TO 1.0 DEGREE AND ZERO THE COMMANDED ATTITUDE CHANGE AND COMMANDED RATE
#
#       ALL ENTRIES SET UP A NOVAC JOB TO DO 1/ACCS SO THAT THE TJETLAW SWITCH CURVES ARE POSITIONED TO
#       REFLECT THE NEW DEADBAND.  IT SHOULD BE NOTED THAT THE DEADBAND REFERS TO THE ATTITUDE IN THE P-, U-, AND V-AXES.
#
# SUBROUTINE CALLED:    NOVAC
#
# CALLING SEQUENCE:     SAME AS ABOVE
#                       OR      TC RESTORDB +1    FROM ALLCOAST
#
# DEBRIS:               A, L, Q, RUPTREG1, (ITEMPS IN NOVAC)

RESTORDB        CAE     DAPBOOLS        # DETERMINE CREW-SELECTED DEADBAND.
                MASK    DBSELECT
                EXTEND
                BZF     SETMINDB

SETMAXDB        CAF     WIDEDB          # SET 5 DEGREE DEADBAND.
 +1             TS      DB

                EXTEND                  # SET UP JOB TO RE-POSITION SWITCH CURVES.
                QXCH    RUPTREG1
CALLACCS        CAF     PRIO27
                TC      NOVAC
                EBANK=  AOSQ
                2CADR   1/ACCJOB

                TC      RUPTREG1        # RETURN TO CALLER.

SETMINDB        CAF     NARROWDB        # SET 0.3 DEGREE DEADBAND.
                TCF     SETMAXDB +1

PFLITEDB        EXTEND                  # THE RETURN FROM CALLACCS IS TO RUPTREG1.
                QXCH    RUPTREG1
                TC      ZATTEROR        # ZERO THE ERRORS AND COMMANDED RATES.
                CAF     POWERDB         # SET DB TO 1.0 DEG.
                TS      DB
                TCF     CALLACCS        # SET UP 1/ACCS AND RETURN TO CALLER.
NARROWDB        OCTAL   00155           # 0.3 DEGREE SCALED AT 45.
```

```
WIDEDB          OCTAL   03434           # 5.0 DEGREES SCALED AT 45.
POWERDB         DEC     .02222          # 1.0 DEGREE SCALED AT 45.

ZATTEROR        CAF     EBANK6
                XCH     EBANK
                TS      L               # SAVE CALLERS EBANK IN L.
                CAE     CDUX
                TS      CDUXD
                CAE     CDUY
                TS      CDUYD
                CAE     CDUZ
                TS      CDUZD
                TCF     STOPRATE +3

STOPRATE        CAF     EBANK6
                XCH     EBANK
                TS      L               # SAVE CALLERS EBANK IN L.
 +3             CAF     ZERO
                TS      OMEGAPD
                TS      OMEGAQD
                TS      OMEGARD
                TS      DELCDUX
                TS      DELCDUY
                TS      DELCDUZ
                TS      DELPEROR
                TS      DELQEROR
                TS      DELREROR
                LXCH    EBANK           # RESTORE CALLERS EBANK.
                TC      Q

# SUBROUTINE NAME:      ALLCOAST
# WILL BE CALLED BY FRESH STARTS AND ENGINE OFF ROUTINES.
#
# CALLING SEQUENCE:     (SAME AS ABOVE)
#
# EXIT:                 RETURN TO Q.
#
# SUBROUTINES CALLED:   STOPRATE, RESTORDB, NOVAC
#
# ZERO:                 (FOR ALL AXES) AOS, ALPHA, AOSTERM, OMEGAD, DELCDU, DELEROR
#
# OUTPUT:               DRIFTBIT/DAPBOOLS, OE, JOB TO DO 1/ACCS
#
# DEBRIS:               A, L, Q, RUPTREG1, RUPTREG2, (ITEMPS IN NOVAC)

ALLCOAST        EXTEND                  # SAVE Q FOR RETURN
                QXCH    RUPTREG2
```

```
                TC      STOPRATE        # CLEAR RATE INTERFACE.  RETURN WITH A=0
                LXCH    EBANK           #   AND L=EBANK6.  SAVE CALLER'S EBANK.
                TS      AOSQ
                TS      AOSQ +1
                TS      AOSR
                TS      AOSR +1
                TS      ALPHAQ          # FOR DOWNLIST.
                TS      ALPHAR
                TS      AOSQTERM
                TS      AOSRTERM
                LXCH    EBANK           # RESTORE EBANK (EBANK6 NO LONGER NEEDED)

                CS      DAPBOOLS        # SET UP DRIFTBIT
                MASK    DRIFTBIT
                ADS     DAPBOOLS
                TC      RESTORDB +1     # RESTORE DEADBANK TO CREW-SELECTED VALUE.

                TC      RUPTREG2        # RETURN.
```

```
# THE DAPIDLER PROGRAM IS STARTED BY FRESH START AND RESTART.          THE DAPIDLER PROGRAM IS DONE 10 TIMES
# PER SECOND UNTIL THE ASTRONAUT DESIRES THE DAP TO WAKE UP, AND THE IMU AND CDUS ARE READY FOR USE BY THE DAP.
# THE NECESSARY INITIALIZATION OF THE DAP IS DONE BY THE DAPIDLER PROGRAM.
                BANK    16
                SETLOC  DAPS1
                BANK

                EBANK=  AOSQ

                COUNT*  $$/DAPID

CHEKBITS        EXTEND
                READ    CHAN31          # IF BOTH BIT13 AND BIT14 ARE ONE, THEN
                COM                     # THE MODE SELECT SWITCH IS IN THE OFF
                MASK    BIT13-14        # POSITION, AND SO THE DAP SHOULD BE OFF,
                EXTEND                  # WITH NO ATTITUDE ERROR DISPLAY.
                BZF     MOREIDLE

                CS      IMODES33
                MASK    BIT6
                CCS     A
                TCF     JUMPDSP
                CS      RCSFLAGS        # IMU NOT USABLE.  SET UP INITIALIZATION
                MASK    BIT3            # FLAG FOR ATT ERROR DISPLAY ROUTINE.
                ADS     RCSFLAGS
                TCF     SHUTDOWN

CHEKMORE        CAF     BIT10           # BIT 10 OF 30 IS PGNCS CONTROL OF S/C
                EXTEND
                RAND    CHAN30          # BITS IN 30 ARE INVERTED
                CCS     A
                TCF     MOREIDLE

                RETURN
```

```
# DAPIDLER ENTRY.
DAPIDLER        LXCH    BANKRUPT        # INTERRUPT LEAD INS (CONTINUED)
                EXTEND
                QXCH    QRUPT

                CA      RCSFLAGS
                MASK    BIT13
                CCS     A               # CHECK IF 1/ACCJOB HAS BEEN SET UP SINCE
                TCF     CHECKUP         # THE LAST FRESH START OR RESTART.
                CA      BIT13
                ADS     RCSFLAGS        # BIT 13 IS 1.
                CAF     PRIO27
                TC      NOVAC           # SET UP JOB TO DO A LITTLE INITIALIZATION
                EBANK=  AOSQ            #   AND EXECUTE 1/ACCS.
                2CADR   1/ACCSET        # (WILL BRANCH TO MOREIDLE ON ACCSOKAY)

CHECKUP         TC      CHEKBITS        # CHECK TO SEE IF LM DAP IS TO GO ON AND
                                        #   DO ERROR DISPLAY.

                CAE     DAPBOOLS        # IF 1/ACCS HAS NOT BEEN COMPLETED, IDLE.
                MASK    ACCSOKAY        #   NOTE: ONLY FRESH START AND RESTART
                EXTEND                  #         KNOCK THIS BIT DOWN.
                BZF     MOREIDLE

STARTDAP        TC      IBNKCALL        # ZERO ATTITUDE ERROR AND DESIRED RATES.
                FCADR   ZATTEROR
                CAF     ZERO            # ********** INITIALIZE: **********
                TS      TJP
                TS      TJU
                TS      TJV
                TS      OMEGAP          # RATES IN BODY (PILOT) COORDINATES.
                TS      OMEGAQ
                TS      OMEGAR
                TS      TRAPEDP
                TS      TRAPEDQ
                TS      TRAPEDR
                TS      AOSQ            # OFFSET ACCELERATION ESTIMATES.
                TS      AOSQ +1
                TS      AOSR
                TS      AOSR +1
                TS      ALPHAQ          # COPIES OF OFFSET ESTIMATES FOR DOWNLIST.
                TS      ALPHAR
                TS      NEGUQ
                TS      NEGUR
                TS      AOSQTERM        # QRAXIS RATE DERIVATION TERMS AND KALMAN
                TS      AOSRTERM        # FILTER INITIALIZATION TERMS.
                TS      QACCDOT         # DESCENT ACCELERATION DERIVATIVE EST.
                TS      RACCDOT
```

```
                TS      ALLOWGTS        # AOSTASK FLAG FOR QRAXIS RCS CONTROL USE.
                TS      COTROLER        # DO TRYGTS ON FIRST PASS (WILL GO TO RCS)
                TS      INGTS           # RECOGNIZE FIRST GTS PASS AS SUCH.
                TS      QGIMTIMR        # STOP GIMBAL DRIVES.  (PROBABLY WOULD BE
                TS      RGIMTIMR        #   GOOD ENOUGH JUST TO INACTIVATE TIMERS)
                TS      OLDPMIN         # MINIMUM IMPULSE MODE ERASABLES
                TS      OLDQRMIN
                TS      PJETCTR         # INITIALIZE DOCKED JET INHIBITION
                TS      UJETCTR         # COUNTERS
                TS      VJETCTR
CALLGMBL        EQUALS  BIT5            # RCSFLAGS INITIALIZATION.
                CS      MANFLAG
                MASK    RCSFLAGS        # NEGUQ(R) HAVE BEEN GENERATED.
                TS      RCSFLAGS

# SET UP "OLD" MEASURED CDU ANGLES:

                EXTEND
                DCA     CDUX            # OLDXFORP AND OLDYFORP
                DXCH    OLDXFORP
                CA      CDUZ
                TS      OLDZFORQ
                CS      RCSFLAGS
                MASK    BIT12
                ADS     RCSFLAGS        # BIT 12 SET TO 1.
                CA      FOUR
                TS      SKIPU
                TS      SKIPV
                CA      POSMAX
                TS      TIME6
                TS      T6NEXT
                TS      T6FURTHA
                CA      ZERO
                TS      T6NEXT +1
                TS      T6FURTHA +1
                TS      NXT6ADR
                TS      NEXTP
                TS      NEXTU
                TS      NEXTV
                CS      TEN
                TS      DAPZRUPT        # JASK NOT IN PROGRESS, INITIALIZE NEG.
                CA      TWO
                TS      NPTRAPS
                TS      NQTRAPS
                TS      NRTRAPS
                EXTEND
                DCA     PAXADIDL
                DXCH    T5ADR
SETTIME5        CAF     MS100
                TS      TIME5
```

```
                TCF     RESUME
                EBANK=  AOSQ
IDLERADR        2CADR   DAPIDLER

MOREIDLE        TC      IBNKCALL        # CALCULATE Q,R-AXES ATTITUDE ERRORS.
                CADR    QERRCALC

                TC      IBNKCALL
                CADR    CALCPERR        # CALCULATE P AXIS ATTITUDE ERRORS.

SHUTDOWN        EXTEND
                DCA     IDLERADR
                DXCH    T5ADR

                CAF     ZERO            # KILL ANY POSSIBLE JET REQUESTS
                TS      NEXTP
                TS      NEXTU
                TS      NEXTV
                EXTEND                  # COMMAND JETS OFF.
                WRITE   CHAN5
                EXTEND
                WRITE   CHAN6
                CS      BGIM23          # TURN TRIM GIMBAL OFF
                EXTEND
                WAND    CHAN12
                TCF     SETTIME5        # RETURN IN 100 MSEC.

MANFLAG         OCT     03021
BGIM23          OCTAL   07400
                EBANK=  OMEGAP
PAXADIDL        2CADR   PAXIS

MS100           =       OCT37766
COSMG           =       ITEMP1
JUMPDSP         EXTEND                  # TRANSFER TO BANK 20
                DCA     DSPCADR         # FOR ATTITUDE ERROR DISPLAYS
                DTCB

                EBANK=  AK
DSPCADR         2CADR   ALTDSPLY
```

```
                BANK    20
                SETLOC  DAPS3
                BANK
                COUNT*  $$/NEEDL

# PROGRAM: ALTDSPLY
#
# MOD 0.  6 DEC 1967
#
# AUTHOR:  CRAIG WORK, DON KEENE, MIT IL
#
# MOD 3 BY DON KEENE AUG 1, 1968 MOVED PROGRAM TO BANK 20
#
# PROGRAM DESCRIPTION:
#
# ALTDSPLY REVERSES THE DSPLYALT BIT OF RCSFLAGS EACH TIME IT IS CALLED, WHICH IS PRESUMABLY EVERY 100 MS.
# IF THE REVERSED BIT IS ONE, NEEDLER IS CALLED TO DISPLAY ATTITUDE ERRORS.  IF THE BIT IS ZERO, THE ATTITUDE ERR-
# ORS ARE CALCULATED AS 1) DAP FOLLOWING ERRORS, IF NEEDLFLG = 0, AND 2) TOTAL ATTITUDE ERRORS FOR NEEDLFLG = 1.
#
# WARNING:  ALTDSPLY MAY ONLY BE CALLED WITH INTERRUPT INHIBITED.
#
# WARNING:  EBANK MUST BE SET TO 6 WHEN USING THIS ROUTINE.
#
# INPUT:  RCSFLAGS AND 1) IF NEEDLFLG=0, INPUT PERROR,QERROR,RERROR.
#                      2) IF NEEDLFLG=1, INPUT CPHI,CTHETA,CPSI,CDUX,CDUY,CDUZ,M11,M21,M32,M22,M32. (GPMATRIX)
#
# OUTPUTS:  RCSFLAGS WITH DSPLYALT REVERSED,AK,AK1,AK2,+ NEEDLER OUTPUTS.
#
# ENTRY:   TCF    ALTDSPLY
#
# EXIT:    TCF    CHEKMORE
#
# ALARM OR ABORT EXITS:  NONE
#
# SUBPROGRAMS CALLED: NEEDLER, OVERSUB2
#
# DEBRIS:  A,L,AND NEEDLER DEBRIS.

ALTDSPLY        CA      RCSFLAGS        # INVERT THE DISPLAY ALTERNATION BIT.
                TS      L
                CA      DSPLYALT
                EXTEND
                RXOR    LCHAN
                TS      RCSFLAGS

                MASK    DSPLYALT
                CCS     A               # IS ALTERNATION FLAG ZERO?
                TCF     NEEDLER

                CAE     FLAGWRD0        # NEEDLFLG WILL INDICATE TOTAL OR DAP AT-
```

```
                MASK    NEEDLBIT        # TITUDE ERROR DISPLAY REQUEST.
                CCS     A
                TCF     DSPLYTOT        # TOTAL ERROR IS NEEDED IN AK,AK +1,AK +2

                CS      QERROR          # YES. DISPLAY ATT ERRORS ON THE ,-BALL.
                TS      AK +1           # ERROR COMPLEMENTS ARE INPUT TO NEEDLER.
                CS      RERROR
                TS      AK +2
                CS      PERROR
                XCH     AK

                TCF     RETNMORE        # DISPLAY THESE THE NEXT TIME THROUGH

# CALCULATE GIMBAL ANGLE TOTAL ERRORS, RESOLVE INTO PILOT AXES, STORE TOTAL ERRORS FOR NEEDLER.  Q-AXIS FIRST.

DSPLYTOT        EXTEND
                QXCH    ITEMP1          # SAVE Q FOR CHEKBITS RETURN.

                CA      CTHETA          # DESIRED ATTITUDE, Y-AXIS, 2'S COMP.
                EXTEND                  # SUBTRACT CURRENT ATTITUDE.
                MSU     CDUY            # DIFFERENCE SCALED AT PI, 1'S COMP.
                TS      AK              # SAVE FOR R-ERROR CALCULATION.
                EXTEND
                MP      M21             # (CHTETA-CDUY)*M21 SCALED AT PI RADIANS.
                XCH     AK +1           # STORE FIRST TERM OF Q ERROR.
                CA      CPSI            # DESIRED ATTITUDE,Z-AXIS, 2'S COMP.
                EXTEND                  # SUBTRACT CURRENT ATTITUDE.
                MSU     CDUZ            # DIFFERENCE SCALED AT PI, 1'S COMP.
                TS      AK +2           # SAVE Z-AXIS TERM FOR R ERROR CALCULATION
                EXTEND
                MP      M22             # (CPSI-CDUZ)*M22, SCALED AT PI RADIANS.
                AD      AK +1           # Q ERROR COMPLETE         , AT PI RAD.
                TC      OVERSUB2        # PIN NEEDLES IN CASE OF OVERFLOW
                TS      AK +1

# R ERROR CALCULATION NEXT.

                CA      AK              # Y-AXIS DIFFERENCE STORED BY Q-AXIS CALC.
                EXTEND
                MP      M31             # (CTHETA-CDUY)*M31, SCALED AT PI RADIANS.
                XCH     AK +2           # FIRST TERM OF R ERROR.
                                        # Z-AXIS DIFFERENCE, STORED BY A CALC. IS
                EXTEND                  # RECOVERED BY THE EXCHANGE.
                MP      M32             # (CPSI-CDUZ)*M32, SCALED AT PI RADIANS.
                AD      AK +2           # R ERROR COMPLETE         , AT PI RAD.
                TC      OVERSUB2        # PIN NEEDLES IN CASE OF OVERFLOW.
                TS      AK +2

# NOW CALCULATE P ERROR.  (NOTE THAT M13 = 1, SCALED AT 1, SO THE MULTIPLICATION IS BY-PASSED.)
```

```
                CA      AK              # Y-AXIS DIFFERENCE STORED BY Q AXIS CALC.
                EXTEND
                MP      M11             # (CTHETA-CDUY)*M11 SCALED AT PI RADIANS.
                XCH     AK              # FIRST TERM OF P ERROR IN AK, AT PI RAD.
                CAE     CPHI            # DESIRED ATTITUDE, X-AXIS, 2'S COMP.
                EXTEND                  # SUBTRACT CURRENT X ATTITUDE.
                MSU     CDUX            # X-AXIS DIFFERENCE, 1'S COMP, AT PI RAD.

# M13 = 1, SO BYPASS THE MULTIPLICATION.
#               EXTEND
#               MP      M13               (CPHI-CDUX)*M13 SCALED AT PI RADIANS.

                AD      AK              # P ERROR COMPLETE      , SCALED AT PI RAD
                TC      OVERSUB2        # PIN NEEDLES IN CASE OF OVERFLOW.
                TS      AK

                EXTEND
                QXCH    ITEMP1          # RESTORE Q FOR CHEKBITS RETURN.

                TCF     RETNMORE        # DISPLAY THESE THE NEXT TIME THROUGH
```

```
# FDAI ATTITUDE ERROR DISPLAY SUBROUTINE
#
# PROGRAM DESCRIPTION:    D. KEENE   5/24/67
#
# MOD 1 BY CRAIG WORK, 12 DEC 67
#
# MOD 2 BY CRAIG WORK, 6 APRIL 68 CONVERTS ATTITUDE ERROR DISPLAY SCALING FROM 16 7/8 DEG. TO 42 3/16 DEGREES.
#
#     THIS SUBROUTINE IS USED TO DISPLAY ATTITUDE ERRORS ON THE FDAI VIA THE DIGITAL TO ANALOG CONVERTERS (DACS)
# IN THE CDUS.  CARE IS TAKEN TO METER OUT THE APPROPRIATE NUMBER OF PULSES TO THE IMU ERROR COUNTERS AND PREVENT
# OVERFLOW, TO CONTROL THE RELAY SEQUENCING, AND TO AVOID INTERFERENCE WITH THE COARSE ALIGN LOOP WHICH ALSO USES
# THE DACS.
#
#
# CALLING SEQUENCE:
#
#     DURING THE INITIALIZATION SECTION OF THE USER'S PROGRAM, BIT3 OF RCSFLAGS SHOULD BE SET TO INITIATE THE
# TURN-ON SEQUENCE WITHIN THE NEEDLES PROGRAM:
#
#          CS     RCSFLAGS        IN EBANK6
#          MASK   BIT3
#          ADS    RCSFLAGS
#
# THEREAFTER, THE ATTITUDE ERRORS GENERATED BY THE USER SHOULD BE TRANSFERRED TO THE FOLLOWING LOCATIONS IN EBANK6:
#
#          AK     SCALED 180 DEGREES  NOTE: THESE LOCATIONS ARE SUBJECT
#          AK1    SCALED 180 DEGREES        TO CHANGE
#          AK2    SCALED 180 DEGREES
#
# FULL SCALED DEFLECTION OF THE NEEDLES CORRESPONDS TO 5 1/16 DEGREES, WHILE 384 BITS IN THE IMU ERROR COUNTER
# CORRESPONDS TO 42 3/16 DEGREES.  (DAC MAXIMUM CAPACITY IS 384 BITS.) 46 BITS EFFECTIVELY PIN THE NEEDLES.
#
# A CALL TO NEEDLER WILL THEN UPDATE THE DISPLAY:
#
#          INHINT
#          TC     IBNKCALL       NOTE: EBANK SHOULD BE SET TO E6
#          CADR   NEEDLER
#          RELINT
#
#     THIS PROCESS SHOULD BE REPEATED EACH TIME THE ERRORS ARE UPDATED.  AT LEAST 3 PASSES THRU THE PROGRAM ARE
# REQUIRED BEFORE ANYTHING IS ACTUALLY DISPLAYED ON THE ERROR METERS.
# NOTE: EACH CALL TO NEEDLER MUST BE SEPARATED BY AT LEAST 50MS TO ASSURE PROPER RELAY SEQUENCING.
#
# ERASABLES USED:
#                       AK          CDUXCMD
#                       AK1         CDUYCMD
#                       AK2         CDUZCMD
#                       EDRIVEX     A,L,Q
#                       EDRIVEY     T5TEMP
#                       EDRIVEZ     DINDX
```

```
#
# SWITCHES:              RCSFLAGS   BITS 3,2
#
# I/O CHANNELS:          CHAN12     BIT 4 (COARSE ALIGN - READ ONLY)
#                        CHAN12     BIT 6 (IMU ERROR COUNTER ENABLE)
#                        CHAN14     BIT 13,14,15 (DAC ACTIVITY)
#
#
# SIGN CONVENTION<   AK = THETAC - THETA
#           WHERE    THETAC = COMMAND ANGLE
#                    THETA  = PRESENT ANGLE


NEEDLER         CA      RCSFLAGS
                MASK    SIX
                EXTEND
                BZF     NEEDLES3
                MASK    BIT3
                EXTEND
                BZF     NEEDLER2        # BIT3 = 0, BIT2 = 1

                CS      BIT6            # FIRST PASS BIT3 = 1
                EXTEND                  # DISABLE IMU ERROR COUNTER TO ZERO DACS
                WAND    CHAN12          # MUST WAIT AT LEAST 60 MS BEFORE
NEEDLE11        CS      ZERO            # ENABLING COUNTERS.
                TS      AK              # ZERO THE INPUTS ON FIRST PASS
                TS      AK1
                TS      AK2
                TS      EDRIVEX         # ZERO THE DISPLAY REGISTERS
                TS      EDRIVEY
                TS      EDRIVEZ
                TS      CDUXCMD         # ZERO THE OUT COUNTERS
                TS      CDUYCMD
                TS      CDUZCMD
                CS      SIX             # RESET RCSFLAGS FOR PASS2
                MASK    RCSFLAGS
                AD      BIT2
                TS      RCSFLAGS
                TCF     RETNMORE

NEEDLER2        CAF     BIT6            # ENABLE IMU ERROR COUNTERS
                EXTEND
                WOR     CHAN12
                CS      SIX             # RESET RCSFLAGS TO DISPLAY ATTITUDE
                MASK    RCSFLAGS        # ERRORS    WAIT ATLEAST 4 MS FOR
                TS      RCSFLAGS        # RELAY CLOSURE
                TCF     RETNMORE

NEEDLES3        CAF     BIT6            # CHECK TO SEE IF IMU ERROR COUNTER
                EXTEND                  # IS ENABLED
                RAND    CHAN12
```

```
                CCS     A               # IF NOT, RE-INITIALIZE NEEDLER.
                TCF     NEEDLES

                CS      RCSFLAGS        # SET UP INITIALIZATION FLAG IN RCSFLAGS.
                MASK    BIT3
                ADS     RCSFLAGS
                TCF     RETNMORE

NEEDLES         CAF     TWO
DACLOOP         TS      DINDX
                CS      ONETENTH        # RESCALE INPUTS TO + OR - 1800 DEGREES.
                EXTEND
                INDEX   DINDX
                MP      AK
                TS      L
                CCS     A
                CA      DACLIMIT
                TCF     +2
                CS      DACLIMIT
                AD      L
                TS      T5TEMP          # OVFLO CHK
                TCF     +4
                INDEX   A               # ON OVERFLOW LIMIT OUTPUT TO +-384
                CAF     DACLIMIT
                TS      L
                INDEX   DINDX
                CS      EDRIVEX         # CURRENT VALUE OF DAC
                AD      L
                INDEX   DINDX
                ADS     CDUXCMD
                INDEX   DINDX
                LXCH    EDRIVEX
                CCS     DINDX
                TCF     DACLOOP
                CAF     13,14,15
                EXTEND
                WOR     CHAN14          # SET DAC ACTIVITY BITS
                TCF     RETNMORE

                DEC     -384
DACLIMIT        DEC     16000
                DEC     384

ONETENTH        OCT     03146           # DECIMAL +0.1, SCALED AT 1.
DSPLYALT        EQUALS  BIT4            # 100 MS ALTERNATION BIT IN RCSFLAGS

OVERSUB2        TS      7               # RETURNS  A  UNCHANGED  OR LIMITED TO
                TC      Q               # POSMAX OR NEGMAX IF A HAS OVERFLOW
                INDEX   A
```

```
                CS      LIMITS          # DUPLICATE CODING IN  BANK 16
                TC      Q

RETNMORE        EXTEND                  # RETURN TO CHEKMORE
                DCA     MORECADR
                DTCB

                EBANK=  AOSQ
MORECADR        2CADR   CHEKMORE
```

```
                BANK    16
                SETLOC  DAPS1
                BANK

                EBANK=  PERROR
                COUNT*  $$/DAPP

# THE FOLLOWING T5RUPT ENTRY BEGINS THE PROGRAM WHICH CONTROLS THE P-AXIS ACTION OF THE LEM USING THE RCS JETS.
# THE NOMINAL TIME BETWEEN THE P-AXIS RUPTS IS 100 MS IN ALL NON-IDLING MODES OF THE DAP.

PAXIS           CA      MS100
                ADS     TIME5           # *** NECESSARY IN ORDER TO ALLOW
                                        # SYNCHRONIZATION WITH OTHER INTERRUPTS ***

                LXCH    BANKRUPT        # INTERRUPT LEAD IN (CONTINUED)
                EXTEND
                QXCH    QRUPT

# CHECK IF DAP PASS IS PERMISSIBLE

                CCS     DAPZRUPT        # IF DAPZRUPT POSITIVE, DAP (JASK) IS
                TC      BAILOUT         # STILL IN PROGRESS AND A RESTART IS
                OCT     02000           # CALLED FOR.  IT IS NEVER ZERO

                TC      CHEKBITS        # RETURN IS TC I+1 IF DAP SHOULD STAY ON.

                CA      CDUX            # READ AND STORE CDU'S
                TS      DAPTREG4
                CA      CDUY
                TS      DAPTREG5
                CA      CDUZ
                TS      DAPTREG6

# ***** KALCMANU-DAP AND "RATE-HOLD"-DAP INTERFACE *****
#
# THE FOLLOWING SECTION IS EXECUTED EVERY 100 MS (10 TIMES A SECOND) WITHIN THE P-AXIS REACTION CONTROL SYSTEM
# AUTOPILOT (WHENEVER THE DAP IS IN OPERATION).

                CA      CDUXD
                EXTEND
                MSU     DELCDUX
                TC      1STOTWOS
                TS      CDUXD
                CA      CDUYD
                EXTEND
                MSU     DELCDUY
                TC      1STOTWOS
                TS      CDUYD
                CA      CDUZD
                EXTEND
                MSU     DELCDUZ
```

```
                TC      1STOTWOS
                TS      CDUZD
                EXTEND                  # DIMINISH MANUAL CONTROL DIRECT RATE
                DIM     TCP             # TIME COUNTERS.
                EXTEND
                DIM     TCQR

# RATFLOOP COMPUTES JETRATEQ, JRATER, AND 1JACC*NO. PJEETS IN ITEMP1.
# RETURNS TO BACKP.
#
# JETRATE = 1JACC*NO.PJETS*TJP          (NOTE TJ IS THE TIME FIRED DURING CSP)
# JETRATEQ = 1JACCQ(TJU*NO.UJETS - TJV*NO.VJETS)
# JETRATER = 1JACCR(TJU*NO.UJETS + TJV*NO.VJETS)

                TCF     PAXFILT         # PROCEEDS TO RATELOOP AFTER SUPERJOB
1STOTWOS        CCS     A
                AD      ONE
                TC      Q
                CS      A
                TC      Q
SUBDIVDE        EXTEND                  # OVERFLOW PROTECTION ROUTINE TO GIVE
                MP      DAPTEMP3        # POSMAX OR NEGMAX IF THE DIVIDE WOULD
                DAS     OMEGAU          # OVERFLOW

 +3             EXTEND
                DCA     OMEGAU
                DXCH    DAPTEMP5
                CCS     OMEGAU
                TCF     +2
                TCF     DIVIDER
                AD      -OCT630
                EXTEND
                BZMF    DIVIDER

                CCS     OMEGAU
                CA      POSMAX          # 45 DEG/SEC
                TC      Q
                CS      POSMAX
                TC      Q

DIVIDER         DXCH    OMEGAU
                EXTEND
                DV      DAPTREG4
                TC      Q

OVERSUB         TS      7               # RETURNS A UNCHANGED OR LIMITED TO
                TC      Q               # POSMAX OR NEGMAX IF A HAS OVERFLOW
                INDEX   A
                CS      BIT15   -1
```

```
                TC      Q

-OCT630         OCT     77147

BACKP           CA      DAPTEMP1
                EXTEND
                MP      1JACC
                TS      JETRATE

# BEGINNING OF THE RATE DERIVATION
#       OMEGAP,Q,R      BODY RATES SCALED AT PI/4
#       TRAPEDP,Q,R     BODY ANGLE ERRORS FROM PREDICTED ANGLE (PI/40)
#       NP(QR)TRAPS     NUMBER OF TIMES ANGLE ERROR HAS BEEN ACCUMULATED
#       AOSQ(R)TERM     CHANGE IN RATE DUE TO OFFSET ACCELERATION.  (PI/4)
#       JETRATE,Q,R     CHANGE IN RATE DUE TO  JET   ACCELERATION.  (PI/4)
#       TRAPSIZE        NEGATIVE LIMIT OF MAGNITUDE OF TRAPEDP, ETC.
#       OMEGAU          DP-TEMPORARY STORAGE
# OMEGA = OMEGA + JETRATE + AOSTERM (+TRAPED/NTRAPS IF TRAPED BIG)

                CAE     DAPTREG4        # CDUX IS STORED HERE
                TS      L
                EXTEND
                MSU     OLDXFORP        # SCALED AT PI
                LXCH    OLDXFORP
                TS      DAPTEMP1
                CA      1/40
                TS      DAPTREG4
                CS      JETRATE
                EXTEND
                MP      BIT14
                ADS     TRAPEDP
                CA      JETRATEQ
                AD      AOSQTERM
                EXTEND
                MP      -BIT14
                ADS     TRAPEDQ
                CA      JETRATER
                AD      AOSRTERM
                EXTEND
                MP      -BIT14
                ADS     TRAPEDR

                CA      DAPTREG5        # CDUY IS STORED HERE
                TS      L
                EXTEND
                MSU     OLDYFORP        # SCALED AT PI
                LXCH    OLDYFORP
                TS      DAPTEMP2
                EXTEND
                MP      M11             # M11 SCALED AT 1
```

```
                AD      DAPTEMP1
                DXCH    OMEGAU

                TC      SUBDIVDE +3     # RETURNS WITH CDU-RATE AT PI/4

                EXTEND
                SU      OMEGAP
                ADS     TRAPEDP
                TC      OVERSUB
                TS      TRAPEDP
                EXTEND
                DCA     DAPTEMP5
                DAS     DXERROR
                CS      PLAST
                EXTEND
                MP      1/40
                DAS     DXERROR         # MANUAL MODE X-ATTITUDE ERROR (DP)
                CA      DAPTREG6        # CDUZ IS STORED HERE
                TS      L
                EXTEND
                MSU     OLDZFORQ
                TS      DAPTEMP3
                LXCH    OLDZFORQ
                CA      M21
                EXTEND
                MP      DAPTEMP2
                DXCH    OMEGAU
                CA      M22
                TC      SUBDIVDE

                EXTEND
                SU      OMEGAQ
                ADS     TRAPEDQ
                TC      OVERSUB
                TS      TRAPEDQ
                EXTEND
                DCA     DAPTEMP5
                DAS     DYERROR
                CS      QLAST
                EXTEND
                MP      1/40
                DAS     DYERROR         # MANUAL MODE Y-ATTITUDE ERROR (DP)
                CA      M31
                EXTEND
                MP      DAPTEMP2
                DXCH    OMEGAU
                CA      M32

                TC      SUBDIVDE
```

```
                EXTEND
                SU      OMEGAR
                ADS     TRAPEDR
                TC      OVERSUB
                TS      TRAPEDR         # TRAPEDS HAVE ALL BEEN COMPUTED

                EXTEND
                DCA     DAPTEMP5
                DAS     DZERROR
                CS      RLAST
                EXTEND
                MP      1/40
                DAS     DZERROR         # MANUAL MODE Z-ATTITUDE ERROR (DP)
                CA      DAPBOOLS        # PICK UP PAD LOADED STATE ESTIMATOR GAINS
                MASK    CSMDOCKD
                EXTEND
                BZF     LMONLY
                EXTEND                  # DOCKED
                DCA     DKOMEGAN
                DXCH    DAPTREG4
                CA      DKTRAP
                TCF     +5
LMONLY          EXTEND                  # UNDOCKED
                DCA     LMOMEGAN
                DXCH    DAPTREG4
                CA      LMTRAP
 +5             TS      DAPTREG6
                CCS     TRAPEDP
                TCF     +2
                TCF     SMALPDIF
                AD      DAPTREG6        # TRAPSIZE > ABOUT 77001 %-1.4DEG/SEC"
                EXTEND
                BZMF    SMALPDIF
                ZL
                LXCH    TRAPEDP
                CA      ZERO
                EXTEND
                DV      NPTRAPS
                ADS     OMEGAP
                TC      OVERSUB
                TS      OMEGAP
                CA      DAPTREG4        ABOUT 10 OR 0 FOR DOCKED OR UNDOCKED
                TS      NPTRAPS
SMALPDIF        INCR    NPTRAPS
P-RATE          CA      JETRATE
                ADS     OMEGAP
                TC      OVERSUB
                TS      OMEGAP

                CCS     TRAPEDQ
```

```
                TCF     +2
                TCF     Q-RATE
                AD      DAPTREG6        # TRAPSIZE > ABOUT 77001 %-1.4DEG/SEC"
                EXTEND
                BZMF    Q-RATE
                ZL
                LXCH    TRAPEDQ
                CA      ZERO
                EXTEND
                DV      NQTRAPS
                TS      DAPTEMP1        # SAVE FOR OFFSET ESTIMATE
                ADS     OMEGAQ
                TC      OVERSUB
                TS      OMEGAQ
                CA      DAPTREG4        # ABOUT 10 OR 0 FOR DOCKED OR UNDOCKED
                XCH     NQTRAPS
                AD      DAPTREG5        # KAOS > ABOUT 60D %N/N_60"
                XCH     DAPTEMP1
                EXTEND
                MP      FIVE
                EXTEND
                DV      DAPTEMP1
                ADS     AOSQ
Q-RATE          INCR    NQTRAPS
                CA      JETRATEQ
                AD      AOSQTERM
                ADS     OMEGAQ
                TC      OVERSUB
                TS      OMEGAQ

                CCS     TRAPEDR
                TCF     +2
                TCF     R-RATE
                AD      DAPTREG6        # TRAPSIZE > ABOUT 77001 %-1.4DEG/SEC"
                EXTEND
                BZMF    R-RATE
                ZL
                LXCH    TRAPEDR
                CA      ZERO
                EXTEND
                DV      NRTRAPS
                TS      DAPTEMP2        # SAVE FOR OFFSET ESTIMATE
                ADS     OMEGAR
                TC      OVERSUB
                TS      OMEGAR
                CA      DAPTREG4        # ABOUT 10 OR 0 FOR DOCKED OR UNDOCKED
                XCH     NRTRAPS
                AD      DAPTREG5        # KAOS > ABOUT 60D %N/N_60"
                XCH     DAPTEMP2
                EXTEND
```

```
                MP      FIVE
                EXTEND
                DV      DAPTEMP2
                ADS     AOSR
R-RATE          INCR    NRTRAPS
                CA      JETRATER
                AD      AOSRTERM
                ADS     OMEGAR
                TC      OVERSUB
                TS      OMEGAR

# END OF RATE DERIVATION
#       BEGIN OFFSET ESTIMATER
#               IN POWERED FLIGHT, AOSTASK WILL BE CALLED EVERY 2 SECONDS.
#                       AOS = AOS + K*SUMRATE

                CS      DAPBOOLS
                MASK    DRIFTBIT
                CCS     A
                TCF     WORKTIME
                TS      ALPHAQ          # ZERO THE OFFSET ACCELERATION VALUES.
                TS      ALPHAR
                TS      AOSQTERM
                TS      AOSRTERM
                TS      AOSQ
                TS      AOSR
                TCF     PRETIMCK
KAOS            DEC     60
WORKTIME        CA      QACCDOT
                EXTEND
                MP      CALLCODE        # OCTAL 00032 IS DECIMAL .1 AT 2(6).
                DAS     AOSQ
                CA      AOSQ
                TS      ALPHAQ
                EXTEND
                MP      200MS           # .2 AT 1
                TS      AOSQTERM
                CA      RACCDOT
                EXTEND
                MP      CALLCODE        # OCTAL 00032 IS DECIMAL .1 AT 2(6).
                DAS     AOSR
                CA      AOSR
                TS      ALPHAR
                EXTEND
                MP      200MS           # .2 AT 1
                TS      AOSRTERM
                TCF     PRETIMCK
```

```
PAXFILT         CA      CALLGMBL        # EXECUTE ACDT+C12, IF NEEDED.
                MASK    RCSFLAGS
                CCS     A               # CALLGMBL IS NOT BIT15, SO THIS TEST IS
                TC      ACDT+C12        # VALID.

                DXCH    ARUPT
                DXCH    DAPARUPT
                CA      SUPERJOB        # SETTING UP THE SUPERJOB
                XCH     BRUPT
                LXCH    QRUPT
                DXCH    DAPBQRPT
                CA      SUPERADR
                DXCH    ZRUPT
                DXCH    DAPZRUPT
                TCF     NOQBRSM +1      # RELINT (JUST IN CASE) AND RESUME, IN THE
                                        #       FORM OF A JASK, AT SUPERJOB.

SUPERADR        GENADR  SUPERJOB +1

# COUNT DOWN GIMBAL DRIVE TIMERS AND TURN OFF DRIVES IF REQUIRED.

SUPERJOB        TCF     RATELOOP
PRETIMCK        CCS     QGIMTIMR
                TCF     DECQTIMR        # POSITIVE -- COUNTING DOWN
                TCF     TURNOFFQ        # NEGATIVE -- DRIVE SHOULD BE ENDED
CHKRTIMR        CCS     RGIMTIMR        # NEGATIVE -- INACTIVE
                TCF     DECRTIMR        # (NEG ZERO -- IMPOSSIBLE)
                TCF     TURNOFFR        # REPEATED (ABOVE) FOR R AXIS.

                EXTEND                  # DECREMENT DOCKED JET INHIBITION COUNTERS
                DIM     PJETCTR
                EXTEND
                DIM     UJETCTR
                EXTEND
                DIM     VJETCTR
                CA      BIT12
                MASK    RCSFLAGS
                EXTEND
                BZF     SKIPPAXS
                TC      CHKVISFZ
DECQTIMR        TS      QGIMTIMR        # COUNT TIMERS DOWN TO POS ZERO.
                TCF     CHKRTIMR
DECRTIMR        TS      RGIMTIMR
                TCF     CHKRTIMR +3

TURNOFFQ        TS      NEGUQ           # HALT DRIVES.
                TS      QACCDOT
                CS      QGIMBITS
                EXTEND
```

```
                WAND    CHAN12
                CAF     NEGMAX
                TS      QGIMTIMR
                TCF     CHKRTIMR
TURNOFFR        TS      NEGUR
                TS      RACCDOT
                CS      RGIMBITS
                EXTEND
                WAND    CHAN12
                CAF     NEGMAX
                TS      RGIMTIMR
                TCF     CHKRTIMR +3
QGIMBITS        EQUALS  OCT1400         # BITS 9 AND 10 (OF CHANNEL 12).
RGIMBITS        EQUALS  PRIO6           # BITS 11 AND 12 (OF CHANNEL 12).

SKIPPAXS        CS      RCSFLAGS
                MASK    BIT12
                ADS     RCSFLAGS        # BIT 12 SET TO 1.
                TCF     QRAXIS          # GO TO QRAXIS OR TO CTS.

# Y-X TRANSLATION
#
# INPUT:        BITS 9-12 OF CH31 (FROM TRANSLATION CONTROLLER)
#
# OUTPUT:       NEXTP
#
#               NEXTP IS THE CHANNEL 6 CODE OF JETS FOR THE DESIRED TRANSLATION.
#               IF THERE ARE FAILURES IN THE DESIRED POLICY, THEN
#               (1) FOR DIAGONAL TRANS:         UNFAILED PAIR
#                                               ALARM (IF NO PAIR)
#               (2) FOR PRINCIPAL TRANS:        TRY TO TACK WITH DIAGONAL PAIRS
#                                               ALARM (IF DIAGONAL PAIRS ARE FAILED)

CHKVISFZ        EXTEND
                READ    CHAN31
                CS      A
                MASK    07400OCT
                EXTEND
                BZF     TSNEXTP
                EXTEND
                MP      BIT7
                INDEX   A
                CA      INDXYZ
                TS      ROTINDEX
TRYUORV         CA      SIX
                TC      SELECTYZ
                CS      SIX
                AD      NUMBERT
                EXTEND
```

```
                BZF     TSNEXTP -1
                CS      FIVE
                AD      ROTINDEX
                EXTEND
                BZMF    ALTERYZ
                CS      NUMBERT
                AD      FOUR
                EXTEND
                BZMF    TSNEXTP -1
ABORTYZ         TC      ALARM
                OCT     02001
                CA      BIT1            # INVERT BIT 1 OF RCSFLAGS.
                LXCH    RCSFLAGS
                EXTEND
                RXOR    1
                TS      RCSFLAGS
                CA      ZERO
                TCF     TSNEXTP
ALTERYZ         CA      BIT1            # INVERT BIT 1 OF RCSFLAGS.
                LXCH    RCSFLAGS
                EXTEND
                RXOR    1
                TS      RCSFLAGS
                MASK    BIT1
                AD      FOUR
                ADS     ROTINDEX
                TCF     TRYUORV
                CA      POLYTEMP
TSNEXTP         TS      NEXTP

# STATE LOGIC
#       CHECK IN ORDER:         IF ON
#       LPDPHASE                GO TO PURGENCY
#       PULSES                  MINIMUM PULSE LOTIC
#       DETENT(BIT15 CH31)      RATE COMMAND
#       GOTO TO PURGENCY

                CA      BIT13           # CHECK STICK IF IN ATT. HOLD.
                EXTEND
                RAND    CHAN31
                EXTEND
                BZF     MANMODE

                CA      DAPBOOLS
                MASK    XOVINHIB
                CCS     A
                TCF     PURGENCY        # ATTITUDE STEER DURING VISIBILITY PHASE

                TCF     DETENTCK
MANMODE         CA      PULSES          # PULSES IS ONE FOR PULSE MODE
                MASK    DAPBOOLS
```

```
                EXTEND
                BZF     DETENTCK        # BRANCH FOR RATE COMMAND

                CA      ZERO
                TS      PERROR

# MINIMUM IMPULSE MODE

                CA      CDUX
                TS      CDUXD

                CCS     OLDPMIN
                TCF     CHECKP

FIREP           CA      BIT3
                EXTEND
                RAND    CHAN31
                EXTEND
                BZF     +XMIN

                CA      BIT4
                EXTEND
                RAND    CHAN31
                EXTEND
                BZF     -XMIN

                TCF     JETSOFF

CHECKP          EXTEND
                READ    CHAN31
                CS      A
                MASK    OCT14
                TS      OLDPMIN
                TCF     JETSOFF

-XMIN           CS      TEN             # ANYTHING LESS THAN 14MS. CORRECTED
                TCF     +2              #       IN JET SELECTION ROUTINE
+XMIN           CA      TEN
                TS      TJP
                CA      ONE
                TS      OLDPMIN
                TCF     PJETSLEC -6

#                                       MANUAL RATE COMMAND MODE
#                                       ========================
#                                         BY ROBERT F. STENGEL
#
# THIS MODE PROVIDES RCAH MANUAL CONTROL THRU 2 CONTROL LAWS:  1) DIRECT RATE AND 2) PSEUDO-AUTO.
# THE DIRECT RATE MODE AFFORDS IMMEDIATE CONTROL WITHOUT OVERSHOOT.  THE PSEUDO-AUTO MODE PROVIDES PRECISE
# RATE CONTROL AND ATTITUDE HOLD.
#
```

```
# IN DIRECT RATE, JETS ARE FIRED WHEN STICK POSITION CHANGES BY A FIXED NUMBER OF INCREMENTS IN ONE DAP CYCLE.
# THE `BREAKOUT LEVEL' IS .6 D/S FOR LM-ONLY AND .3 D/S FOR CSM-DOCKED.  THIS LAW NULLS THE RATE ERROR TO WITHIN
# THE `TARGET DEADBAND', WHICH EQUALS THE BREAKOUT LEVEL.
#
# IN PSEUDO-AUTO, BODY-FIXED RATE AND ATTITUDE ERRORS ARE SUPPLIED TO TJETLAW, WHICH EXERCISES CONTROL.
# CONTROL SWITCHES FROM DIRECT RATE TO PSEUDO-AUTO IF THE TARGET DB IS ACHIEVED OR IF TIME IN (1) EXCEEDS 4 SEC.
# IF THE INITIAL COMMAND DOES NOT EXCEED THE BREAKOUT LEVEL, CONTROL GOES TO PSEUDO-AUTO IMMEDIATELY.
#
# SINCE P-AXIS CONTROL IS SEPARATE FROM Q,R AXES CONTROL, IT IS POSSIBLE TO USE (1) IN P-AXIS AND (2) IN Q,R AXES,
# OR VICE VERSA.  THIS ALLOWS A DEGREE OF ATTITUDE HOLD IN UNCONTROLLED AXES.  DUE TO U,V CONTROL, HOWEVER, Q AND
# R AXES ARE COUPLED AND MUST USE THE SAME CONTROL LAW.
#
# HAND CONTROLLER COMMANDS ARE SCALED BY A LINEAR/QUADRATIC LAW.  FOR THE LM-ALONE, MAXIMUM COMMANDED RATES ARE 20
# AND 4 D/S IN NORMAL AND FINE SCALING; HOWEVER, STICK SENSITIVITY AT ZERO COUNTS (OBTAINED AT A STICK DEFLECTION
# OF 2 DEGREES FROM THE CENTERED POSITION) IS .5 OR .1 D/S PER DEGREE.  NORMAL AND FINE SCALINGS FOR THE CSM-DOCKED
# CASE IS AUTOMATICALLY SET TO 1/10 THE ABOVE VALUES.  SCALING IS DETERMINED IN ROUTINE 3.
#
# ZEROENBL        ENABLES COUNTERS SO THEY CAN BE READ NEXT TIME
# JUSTOUT         FIRST DETECTION OF OUT OF DETENT (BY OURRCBIT)

DETENTCK        EXTEND
                READ    CHAN31
                TS      CH31TEMP
                MASK    BIT15           # CHECK OUT-OF-DETENT BIT.
                EXTEND
                BZF     RHCMOVED        # BRANCH IF OUT OF DETENT.
                CAF     OURRCBIT        # IN DETENT.  CHECK THE RATE COMMAND BIT.
                MASK    DAPBOOLS
                EXTEND
                BZF     PURGENCY        # BRANCH IF NOT IN RATE COMMAND LAST PASS.

# ................................................................................

                CA      BIT9            # JUST IN DETENT??
                MASK    RCSFLAGS
                EXTEND
                BZF     RUTH
                CAF     BIT13           # CHECK FOR ATTITUDE HOLD.
                EXTEND
                RAND    CHAN31
                EXTEND
                BZF     RATEDAMP        # BRANCH IF IN ATTITUDE HOLD.

                CS      BITS9,11        # IN AUTO.
                MASK    RCSFLAGS        # (X-AXIS OVERRIDE)
                TS      RCSFLAGS        # ZERO ORBIT (BIT 11) AND JUST-IN BIT (9).
                TCF     RATEDAMP

RUTH            CA      RCSFLAGS
                MASK    PBIT            # IN ATTITUDE HOLD.
                EXTEND
                BZF     +2              # BRANCH IF P-RATE DAMPING IS FINISHED.
                TCF     RATEDAMP
```

```
                CA      RCSFLAGS
                MASK    QRBIT
                EXTEND
                BZF     RATEDONE        # BRANCH IF Q,R RATE DAMPING IS FINISHED.
                TCF     RATEDAMP

# ============================================

1/10SEC         OCT     1
40CYC           OCT     50
PQRBIT          OCT     74777
BITS9,11        EQUALS  EBANK5
LINRATP         DEC     46

# ============================================

RATEDONE        CS      OURRCBIT        # MANUAL COMMAND AND DAMPING COMPLETED IN
                INHINT                  # ALL AXES.
                MASK    DAPBOOLS
                TS      DAPBOOLS

# READ CDUS INTO CDU DESIRED REGISTERS

                CAF     BIT13
                EXTEND
                RAND    CHAN31
                EXTEND
                BZF     +4
                CA      CDUX            # (X-AXIS OVERRIDE)
                TS      CDUXD
                TC      +3
                TC      IBNKCALL
                FCADR   ZATTEROR
                RELINT
                TCF     PURGENCY

                TS      PERROR
JUSTOUT         CA      OURRCBIT        # INITIALIZATION -- FIRST MANUAL PASS.
                ADS     DAPBOOLS
                CA      ZERO
                TS      DXERROR
                TS      DXERROR +1
                TS      DYERROR
                TS      DYERROR +1
                TS      DZERROR
                TS      DZERROR +1
                TS      PLAST
                TS      QLAST
                TS      RLAST
                TS      Q-RHCCTR
                TS      R-RHCCTR
                CA      PQRBIT
                MASK    RCSFLAGS
                TS      RCSFLAGS        # BITS 10 AND 11 OF RCSFLAGS ARE 0.
```

```
                CS      RCSFLAGS        # SET 'JUST-IN' BIT TO 1.
                MASK    BIT9
                ADS     RCSFLAGS
                TC      ZEROENBL
                TCF     JETSOFF
ZEROENBL        LXCH    R-RHCCTR
                CA      Q-RHCCTR
                DXCH    SAVEHAND
                CA      ZERO
                TS      P-RHCCTR
                TS      Q-RHCCTR
                TS      R-RHCCTR
                CA      BITS8,9
                EXTEND
                WOR     CHAN13          # COUNTERS ZEROED AND ENABLED
                TC      Q
RATEDAMP        CA      ZERO
                TS      P-RHCCTR
                TCF     RATERROR

RHCMOVED        CA      OURRCBIT        # P CONTROL
                MASK    DAPBOOLS
                EXTEND
                BZF     JUSTOUT -1
RATERROR        CA      CDUX            # FINDCDUW REQUIRES THAT CDUXD=CDUX DURING
                TS      CDUXD           # X-AXIS OVERRIDE
                CCS     P-RHCCTR
                TCF     +3
                TCF     +2
                TCF     +1
                DOUBLE                  # LINEAR/QUADRATIC CONTROLLER SCALING
                DOUBLE                  # (SEE EXPLANATION OF Q,R-AXES RCS
                AD      LINRATP         # AUTOPILOT)
                EXTEND
                MP      P-RHCCTR
                CA      L
                EXTEND
                MP      STIKSENS
                XCH     PLAST
                COM
                AD      PLAST
                TS      DAPTEMP1
                TC      ZEROENBL        # INTERVAL.  ZERO AND ENABLE ACA COUNTERS.
                CS      PLAST
                AD      OMEGAP
                TS      EDOTP
                CCS     DAPTEMP1        # IF P COMMAND CHANGE EXCEEDS BREAKOUT
                TCF     +3              # LEVEL, GO TO DIRECT RATE CONTROL.  IF NOT
                TCF     +8D             # CHECK FOR DIRECT RATE CONTROL LAST TIME.
                TCF     +1
```

```
                AD      -RATEDB
                EXTEND
                BZMF    +4
                CA      40CYC
                TS      TCP
                TC      PEGI
                CA      RCSFLAGS        # CHECK FOR DIRECT RATE COMMAND LAST TIME.
                MASK    PBIT
                EXTEND
                BZF     +2
                TC      PEGI            # TO PURE RATE COMMAND
                CA      DXERROR         # PSEUDO-AUTO CONTROL.
                TS      E               # X-ATTITUDE ERROR (SP)
                TS      PERROR          # LOAD P-AXIS ERROR FOR MODE1 FDAI DISPLAY
                TC      PURGENCY +4
PEGI            CA      CDUX            # DIRECT RATE CONTROL.
                TS      CDUXD
                CA      ZERO
                TS      DXERROR
                TS      DXERROR +1
                TS      PERROR          # ZERO P-AXIS ERROR FOR MODE1 FDAI DISPLAY
                CCS     EDOTP
                TC      +3
                TC      +2
                TC      +1
                TS      ABSEDOTP
                AD      TARGETDB
                EXTEND                  # IF RATE ERROR IS LESS THAN DEADBANK,
                BZMF    LAST            # FIRE, AN SWITCH TO PSEUDO-AUTO.
                CA      TCP
                EXTEND                  # IF TIME IN RATE COMMAND EXCEEDS 4 SEC.
                BZMF    LAST
                CS      RCSFLAGS
                MASK    PBIT
                ADS     RCSFLAGS        # BIT 10 IS 1.
                TCF     +4
LAST            CS      PBIT
                MASK    RCSFLAGS
                TS      RCSFLAGS        # BIT 10 IS 0.
                CS      EDOTP
                EXTEND
                MP      1/ANETP         # 1/2JTACC SCALED AT 2EXP(7)/PI
                DAS     A
                TC      OVERSUB
                EXTEND
                MP      25/32           # A CONTAINS TJET SCALED AT 2EXP(4)(16/25)
                TS      TJP             # 4.JET TIME
                CA      ABSEDOTP
                AD      -2JETLIM        # COMPARING DELTA RATE WITH 2 JET LIMIT
                EXTEND
```

```
                BZMF    +3

                CA      SIX
                TCF     +8D
                CA      TJP
                ADS     TJP

# GOES TO PJETSLEC FOR TWO JETS
# P-JET-SELECTION-ROUTINE (ROTATION)
#
# INPUT:        NUMBERT         4,5,6 FOR WHICH PAIR OR 4 JETS
#               TJP             + FOR +P ROTATION
#
# OUTPUT:       CHANNEL 6
#               PJUMPADR        FOR P-AXIS SKIP
#               (JTLST CALL)    (SMALL TJP)
#
# ORDER OF POLICIES TRIED IN CASE OF FAILURE.
#       +P      -P
#       7,15    8,16
#       4,12    3,11
#       4,7     8,11
#       7,12    11,16
#       12,15   3,16
#       4,15    3,8
#       ALARM   ALARM

                CA      AORBSYST
                MASK    DAPBOOLS
                CCS     A
                CA      ONE
                AD      FOUR
                TS      NUMBERT
PJETSLEC        CA      ONE
                TS      L
                CCS     TJP
                TCF     +5
                TCF     JETSOFF
                TCF     +2
                TCF     JETSOFF
                ZL
                AD      ONE
                TS      ABSTJ
                LXCH    ROTINDEX
                TC      SELECTP
                CS      SIX
                AD      NUMBERT
                EXTEND
                BZF     +2

                CS      TWO
```

```
                AD      FOUR
                TS      NO.PJETS
                CA      POLYTEMP
                TC      WRITEP
                CS      ABSTJ
                AD      +150MST6
                EXTEND
                BZMF    QRAXIS          # GO TO QRAXIS OR TO GTS.

                AD      -136MST6
                EXTEND
                BZMF    +5

                ADS     ABSTJ
                INDEX   ROTINDEX
                CA      MINTIMES
                TS      TJP

                CA      ABSTJ
                ZL
                INHINT
                DXCH    T6FURTHA
                TC      IBNKCALL
                CADR    JTLST
                CS      BIT12
                MASK    RCSFLAGS
                TS      RCSFLAGS        # BIT 12 SET TO 0.
                TC      ALTSYST
                TCF     QRAXIS

ALTSYST         CA      DAPBOOLS        # ALTERNATE P-AXIS JETS
                TS      L
                CA      AORBSYST
                EXTEND
                RXOR    LCHAN
                TS      DAPBOOLS
                RELINT
                TC      Q

DKALT           TC      ALTSYST

JETSOFF         TC      WRITEP  -1
                CA      ZERO
                TS      TJP
                TCF     QRAXIS

# (NOTE -- M13 = 1 IDENTICALLY IMPLIES NULL MULTIPLICATION.)

CALCPERR        CA      CDUY            # P-ERROR CALCULATION.
                EXTEND
                MSU     CDUYD           # CDU VALUE -- ANGLE DESIRED (Y-AXIS)
```

```
                EXTEND
                MP      M11             # (CDUY-CDUYD)M11 SCALED AT PI RADIANS
                XCH     E               # SAVE FIRST TERM (OF TWO)
                CA      CDUX            # THIRD COMPONENT
                EXTEND
                MSU     CDUXD           # CDU VALUE -- ANGLE DESIRED (X-AXIS)
#               EXTEND
#               MP      M13
                AD      DELPEROR        # KALCMANU INTERFACE ERROR.
                ADS     E               # SAVE SUM OF TERMS.  COULD BE OVERFLOW.
                XCH     PERROR          # SAVE P-ERROR FOR EIGHT-BALL DISPLAY.
                TC      Q               # RETURN TO CALLER

# P-AXIS URGENCY FUNCTION CALCULATION.

PURGENCY        TC      CALCPERR        # CALCULATE P-AXIS ERRORS.
                CS      OMEGAPD         # THIS CODING IS COMMON TO BOTH LM DAP AND
                AD      OMEGAP          # SPS-BACKUP MODE.
                TS      EDOTP           # EDOTP = OMEGAP - OMEGAPD AT PI/4 RAD/SEC

                CS      ONE
                TS      AXISCTR
                CA      DAPBOOLS
                MASK    CSMDOCKD
                EXTEND
                BZF     HEADTJET
                INHINT                  # IF CSMDOCKD = 1, GOT TO DOCKED RCS LOGIC
                TC      IBNKCALL
                CADR    SPSRCS

                CA      TJP
                EXTEND
                BZF     DKALT           # IF TJP = ZERO, CHANGE AORBSYST.
                RELINT
                TCF     PJETSLEC -6     # SELECT AORBSYST AND USE TWO JETS.
HEADTJET        CA      ZERO
                TS      SENSETYP
                INHINT
                TC      IBNKCALL
                CADR    TJETLAW
                RELINT

                CS      FIREFCT
                AD      -FOURDEG
                EXTEND
                BZMF    PJETSLEC -6
                CCS     TJP
                TCF     +2
                TCF     JETSOFF
```

```
                AD      -16OMST6
                EXTEND
                BZMF    PJETSLEC -6
                CA      SIX
                TCF     PJETSLEC -1
-16OMST6        DEC     -256
-FOURDEG        DEC     -.08888
```

```
# JET POLICY CONSTRUCTION SUBROUTINE
#
# INPUT:          ROTINDEX, NUMBERT
#
# OUTPUT:         POLYTEMP (JET POLICY)
#
# THIS SUBROUTINE SELECT A SUBSET OF THE DESIRED JETS WHICH HAS NO FAILURE

SELECTP         CA      SIX
                TS      TEMPNUM
                INDEX   NUMBERT
                CA      TYPEP
                INDEX   ROTINDEX
                MASK    JETSALL
                TS      POLYTEMP
                MASK    CH6MASK
                CCS     A
                TCF     +2
                TC      Q
                CCS     TEMPNUM
                TCF     +4
                TC      ALARM
                OCT     02003
                TCF     JETSOFF         # ********** TCF ALARMJET **********
SELECTYZ        TS      NUMBERT
                TCF     SELECTP +1
 -1             TCF     ABORTYZ +2
JETSALL         OCT     00252
                OCT     00125           # +P
                OCT     00140           # -Y
                OCT     00006           # -Z
                OCT     00220           # +Y
                OCT     00011           # +Z
                OCT     00151           # +V
TYPEP           OCT     00146           # -U
                OCT     00226           # -V
                OCT     00231           # +U
                OCT     00151           # +V
                OCT     00132           # 1-3
                OCT     00245           # 2-4
                OCT     00377           # ALL
INDXYZ          =       -136MST6
-136MST6        DEC     -218
                DEC     4
                DEC     2
                OCT     07776
                DEC     5
                DEC     9
                DEC     10
                OCT     07776
                DEC     3
```

```
                DEC     8
                DEC     7
                OCT     07776           # THESE INDEXES OF MASK JETSALL WILL
                OCT     07776           # CHANGE THE INSTRUCTION AT SELECTP +4
                OCT     07776           # TO BE         TC JETSALL -1
                OCT     07776           # ONLY USED FOR TRANSLATION FAILURE
                OCT     07776
+150MST6        DEC     240
07400OCT        OCT     07400

# T-JET LAW FIXED CONSTANTS

NORMSCL         OCT     266
-100MS          DEC     -.1
200MS           DEC     .2
25/32           =       PRIO31          # (DEC .78125)
BITS8,9         OCTAL   00600
1/40            DEC     .02500
MINTIMES        DEC     -22
                DEC     22
PSKIPADR        GENADR  SKIPPAXS

# GOES TO Q,R-AXES RCS AUTOPILOT

QRAXIS          CS      OMEGARD
                AD      OMEGAR
                TC      OVERSUB
                TS      EDOTR
                CS      OMEGAQD
                AD      OMEGAQ
                TC      OVERSUB
                TS      EDOTQ
                EXTEND
                DCA     QERRCALL
                DTCB

                EBANK=  AOSQ
QERRCALL        2CADR   CALLQERR
```

```
                BANK    17
                SETLOC  DAPS2
                BANK

                EBANK=  CDUXD

                COUNT*  $$/DAPQR

CALLQERR        CA      BIT13           # CALCULATE Q,R ERRORS UNLESS THESE AXES
                EXTEND                  # ARE IN MANUAL RATE COMMAND.
                RAND    CHAN31
                CCS     A
                TCF     +5              # IN AUTO COMPUTE Q,R ERRORS
                CS      DAPBOOLS        # IN MANUAL RATE COMMAND?
                MASK    OURRCBIT
                EXTEND
                BZF     Q,RORGTS        # IF SO BYPASS CALCULATION OF ERRORS.
                TC      QERRCALC

Q,RORGTS        CCS     COTROLER        # CHOOSE CONTROL SYSTEM FOR THIS DAP PASS:
                TCF     GOTOGTS         #       GTS (ALTERNATES WITH RCS WHEN DOCKED)
                TCF     TRYGTS          #       GTS IF ALLOWED, OTHERWISE RCS
RCS             CAF     ZERO            #       RCS (TRYGTS MAY BRANCH TO HERE)
                TS      COTROLER

                DXCH    EDOTQ
                TC      ROT-TOUV
                DXCH    OMEGAU

# X - TRANSLATION
#
# INPUT:        BITS 7,8 OF CH31 (TRANSLATION CONTROLLER)
#               ULLAGER
#               APSFLAG, DRIFTBIT
#               ACC4OR2X, ACRBTRAN
#
# OUTPUT:       NEXTU, NEXTV    CODES OF TRANSLATION FOR AFTER ROTATION
#               SENSETYP        TELL ROTATION DIRECTION AND DESIRE
#
# X-TRANS POLICIES ARE EITHER 4 JETS OR A DIAGONAL PAIR.  IN 2-JET TRANSLATION THE SYSTEM IS SPECIFIED.  A FAILURE
# WILL OVERRIDE THIS SPECIFICATION.  AN ALARM RESULTS WHEN NO POLICY IS AVAILABLE BECAUSE OF FAILURES.

SENSEGET        CA      BIT7            # INPUT BITS OVERRIDE THE INTERNAL BITS
                EXTEND                  # SENSETYP WILL NOT OPPOSE ANYTRANS
                RAND    CHAN31
                EXTEND
                BZF     +XORULGE
```

```
                CA      BIT8
                EXTEND
                RAND    CHAN31
                EXTEND
                BZF     -XTRANS

                CA      ULLAGER
                MASK    DAPBOOLS
                CCS     A
                TCF     +XORULGE

                TS      NEXTU           # STORE NULL TRANSLATION POLICIES
                TS      NEXTV
                CS      DAPBOOLS        # BURNING OR DRIFTING?
                MASK    DRIFTBIT
                EXTEND
                BZF     TSENSE
                CA      FLGWRD10        # DPS (INCLUDING DOCKED) OR APS?
                MASK    APSFLBIT
                CCS     A
                CAF     TWO             # FAVOR +X JETS DURING AN APS BURN.
TSENSE          TS      SENSETYP
                TCF     QRCONTRL

+XORULGE        CAF     ONE
-XTRANS         AD      FOUR
                TS      ROTINDEX
                AD      NEG3
                TS      SENSETYP        # FAVOR APPROPRIATE JETS DURING TRANS.
                CA      DAPBOOLS
                MASK    ACC4OR2X
                CCS     A
                TCF     TRANS4

                CA      DAPBOOLS
                MASK    AORBTRAN
                CCS     A
                CA      ONE             # THREE FOR B
                AD      TWO             # TWO FOR A SYSTEM 2 JET X TRANS
TSNUMBRT        TS      NUMBERT

                TC      SELCTSUB

                CCS     POLYTEMP
                TCF     +3
                TC      ALARM
                OCT     02002
                CA      00314OCT
                MASK    POLYTEMP
TSNEXTS         TS      NEXTU
```

```
                CS      00314OCT
                MASK    POLYTEMP
                TS      NEXTV

# Q,R-AXES RCS CONTROL MODE SELECTION
#       SWITCHES        INDICATION WHEN SET
#       BIT13/CHAN31    AUTO, GO TO ATTSTEER
#       PULSES          MINIMUM IMPULSE MODE
#       (OTHERWISE)     RATE COMMAND/ATTITUDE HOLD MODE

QRCONTRL        CA      BIT13           # CHECK MODE SELECT SWITCH.
                EXTEND
                RAND    CHAN31          # BITS INVERTED
                CCS     A
                TCF     ATTSTEER
CHKBIT10        CAF     PULSES          # PULSES = 1 FOR MIN IMP USE OF RHC
                MASK    DAPBOOLS
                EXTEND
                BZF     CHEKSTIK        # IN ATT-HOLD/RATE-COMMAND IF BIT10=0

# MINIMUM IMPULSE MODE

                INHINT
                TC      IBNKCALL
                CADR    ZATTEROR
                CA      ZERO
                TS      QERROR
                TS      RERROR          # FOR DISPLAYS
                RELINT

                EXTEND
                READ    CHAN31
                TS      TEMP31          # IS EQUAL TO DAPTEMP1
                CCS     OLDQRMIN
                TCF     CHECKIN

FIREQR          CA      TEMP31
                MASK    BIT1
                EXTEND
                BZF     +QMIN

                CA      TEMP31
                MASK    BIT2
                EXTEND
                BZF     -QMIN

                CA      TEMP31
                MASK    BIT5
```

```
                EXTEND
                BZF     +RMIN

                CA      TEMP31
                MASK    BIT6
                EXTEND
                BZF     -RMIN

                TCF     XTRANS

CHECKIN         CS      TEMP31
                MASK    OCT63
                TS      OLDQRMIN
                TCF     XTRANS

+QMIN           CA      14MS
                TS      TJU
                CS      14MS
                TCF     MINQR
-QMIN           CS      14MS
                TS      TJU
                CA      14MS
                TCF     MINQR
+RMIN           CA      14MS
                TCF     +2
-RMIN           CS      14MS
                TS      TJU
MINQR           TS      TJV
                CA      MINADR
                TS      RETJADR
                CA      ONE
                TS      OLDQRMIN
MINRTN          TS      AXISCTR
                CA      DAPBOOLS
                MASK    CSMDOCKD
                EXTEND
                BZF     MIMRET
                INDEX   AXISCTR         # IF DOCKED, USE 60MS MINIMUM IMPULSE
                CCS     TJU
                CA      60MS
                TCF     +2
                CS      60MS
                INDEX   AXISCTR
                TS      TJU
MIMRET          CA      DAPBOOLS
                MASK    AORBTRAN
                CCS     A
                CA      ONE
                AD      TWO
                TS      NUMBERT
```

```
                TCF     AFTERTJ

60MS            DEC     96              # RSB 2009 -- WAS 96.0.
MINADR          GENADR  MINRTN
OCT63           OCT     63
14MS            =       +TJMINT6

TRANS4          CA      FOUR
                TCF     TSNUMBRT

# RATE COMMAND MODE:
#
# DESCRIPTION (SAME AS P-AXIS)

CHEKSTIK        TS      INGTS           # NOT IN GTS WHEN IN ATT HOLD
                CS      ONE             # 1/ACCS WILL DO THE NULLING DRIVES
                TS      COTROLER        # COME BACK TO RCS NEXT TIME
                CA      BIT15
                MASK    CH31TEMP
                EXTEND
                BZF     RHCACTIV        # BRANCH IF OUT OF DETENT.
                CA      OURRCBIT        # ***********
                MASK    DAPBOOLS        # *IN DETENT*   CHECK FOR MANUAL CONTROL
                EXTEND                  # ***********   LAST TIME.
                BZF     STILLRCS
                CS      BIT9
                MASK    RCSFLAGS
                TS      RCSFLAGS        # BIT 9 IS 0.
                TCF     DAMPING
40CYCL          OCT     50
1/10S           OCT     1
LINRAT          DEC     46

# ============================================================

DAMPING         CA      ZERO
                TS      SAVEHAND
                TS      SAVEHAND +1
RHCACTIV        CCS     SAVEHAND        # ******************
                TCF     +3              # Q,R MANUAL CONTROL    WC = A*(B+|D|)*D
                TCF     +2              # ******************
                TCF     +1
                DOUBLE                  # WHERE
                DOUBLE                  #
                AD      LINRAT          #       WC  = COMMANDED ROTATIONAL RATE
                EXTEND                  #       A   = QUADRATIC SENSITIVITY FACTOR
                MP      SAVEHAND        #       B   = LINEAR/QUADRATIC SENSITIVITY
                CA      L               #       |D| = ABS. VALUE OF DEFLECTION
                EXTEND                  #       D   = HAND CONTROLLER DEFLECTION
                MP      STIKSENS
                XCH     QLAST           # COMMAND Q RATE, SCALED 45 DEG/SEC
                COM
```

```
                AD      QLAST
                TS      DAPTEMP3
                CCS     SAVEHAND +1
                TCF     +3
                TCF     +2
                TCF     +1
                DOUBLE
                DOUBLE
                AD      LINRAT
                EXTEND
                MP      SAVEHAND +1
                CA      L
                EXTEND
                MP      STIKSENS
                XCH     RLAST
                COM
                AD      RLAST
                TS      DAPTEMP4
                CS      QLAST           # INTERVAL.
                AD      OMEGAQ
                TS      QRATEDIF
                CS      RLAST
                AD      OMEGAR
                TS      RRATEDIF
ENTERQR         DXCH    QRATEDIF        # TRANSFORM RATES FROM Q,R TO U,V AXES
                TC      ROT-TOUV
                DXCH    URATEDIF
                CCS     DAPTEMP3        # CHECK IF Q COMMAND CHANGE EXCEEDS
                TC      +3              # BREAKOUT LEVEL.  IF NOT, CHECK R.
                TC      +2
                TC      +1
                AD      -RATEDB
                EXTEND
                BZMF    +2
                TCF     ENTERUV -2      # BREAKOUT LEVEL EXCEEDED.  DIRECT RATE.
                CCS     DAPTEMP4        # R COMMAND BREAKOUT CHECK.
                TC      +3
                TC      +2
                TC      +1
                AD      -RATEDB
                EXTEND
                BZMF    +2
                TCF     ENTERUV -2      # BREAKOUT LEVEL EXCEEDED.  DIRECT RATE.
                CA      RCSFLAGS        # BREAKOUT LEVEL NOT EXCEEDED.  CHECK FOR
                MASK    QRBIT           # DIRECT RATE CONTROL LAST TIME.
                EXTEND
                BZF     +2
                TCF     ENTERUV         # CONTINUE DIRECT RATE CONTROL.
                TCF     STILLRCS        # PSEUDO-AUTO CONTROL.
                CA      40CYCL
```

```
                TS      TCQR
ENTERUV         INHINT                  # DIRECT RATE CONTROL
                TC      IBNKCALL
                FCADR   ZATTEROR
                RELINT
                CA      ZERO
                TS      DYERROR
                TS      DYERROR +1
                TS      DZERROR
                TS      DZERROR +1
                CCS     URATEDIF
                TCF     +3
                TCF     +2
                TCF     +1
                AD      TARGETDB        # IF TARGET DB IS EXCEEDED, CONTINUE
                EXTEND                  # DIRECT RATE CONTROL.
                BZMF    VDB
                CCS     VRATEDIF
                TCF     +3
                TCF     +2
                TCF     +1
                AD      TARGETDB
                EXTEND
                BZMF    +2
                TCF     QRTIME
                CA      ZERO
                TS      VRATEDIF
                TCF     QRTIME
VDB             CCS     VRATEDIF
                TC      +3
                TC      +2
                TC      +1
                AD      TARGETDB        # IF TARGET DB IS EXCEEDED, CONTINUE
                EXTEND                  # DIRECT RATE CONTROL.  IF NOT, FIRE AND
                BZMF    TOPSEUDO        # SWITCH TO PSEUDO-AUTO CONTROL ON NEXT
                CA      ZERO            # PASS.
                TS      URATEDIF
QRTIME          CA      TCQR            # DIRECT RATE TIME CHECK.
                EXTEND
                BZMF    +5              # BRANCH IF TIME EXCEEDS 4 SEC.
                CS      RCSFLAGS
                MASK    QRBIT
                ADS     RCSFLAGS        # BIT 11 IS 1.
                TC      +4
TOPSEUDO        CS      QRBIT
                MASK    RCSFLAGS
                TS      RCSFLAGS        # BIT 11 IS 0.
                CA      HANDADR
                TS      RETJADR
                CA      ONE
```

```
BACKHAND        TS      AXISCTR

                CA      FOUR
                TS      NUMBERT

                INDEX   AXISCTR
                INDEX   SKIPU
                TCF     +1
                CA      FOUR
                INDEX   AXISCTR
                TS      SKIPU
                TCF     LOOPER

                INDEX   AXISCTR
                CCS     URATEDIF        #       INDEX   AXIS    QUANTITY
                CA      ZERO            #       0       -U      1/JETACC-AOSU
                TCF     +2              #       1       +U      1/JETACC+AOSU
                CA      ONE             #       16      -V      1/JETACC-AOSV
                INDEX   AXISCTR         #       17      +V      1/JETACC+AOSV
                AD      AXISDIFF        # JETACC = 2 JET ACCELERATION (1 FOR FAIL)

                INDEX   A
                CS      1/ANET2 +1
                EXTEND
                INDEX   AXISCTR         # UPRATEDIF IS SCALED AT PI/4 RAD/SEC
                MP      URATEDIF        # JET TIME IN A, SCALED 32 SEC
                TS      Q
                DAS     A
                AD      Q
                TS      A               # OVERFLOW SKIP
                TCF     +2
                CA      Q               # RIGHT SIGN AND BIGGER THAN 150MS
SETTIME         INDEX   AXISCTR
                TS      TJU             # SCALED AT 10.67 WHICH IS CLOSE TO 10.24
                TCF     AFTERTJ

ZEROTJ          CA      ZERO
                TCF     SETTIME

HANDADR         GENADR  BACKHAND

# GTS WILL BE TRIED IF
#       1. USEQRJTS = 0,
#       2. ALLOWGTS POS,
#       3. JETS ARE OFF (Q,R-AXES)

TRYGTS          CAF     USEQRJTS        # IS JET USE MANDATORY.         (AS LONG AS
                MASK    DAPBOOLS        # USEQRJTS BIT IS NOT BIT 15, CCS IS SAFE.)
                CCS     A
                TCF     RCS
                CCS     ALLOWGTS        # NO.  DOES AOSTASK OK CONTROL FOR GTS?
```

```
                TCF     +2
                TCF     RCS
                EXTEND
                READ    CHAN5
                CCS     A
                TCF     CHKINGTS
GOTOGTS         EXTEND
                DCA     GTSCADR
                DTCB

CHKINGTS        CCS     INGTS           # WAS THE TRIM GIMBAL CONTROLLING
                TCF     +2              #       YES.  SET UP A DAMPED NULLING DRIVE.
                TCF     RCS             #       NO.  NULLING WAS SET UP BEFORE.  DO RCS.
                INHINT
                TC      IBNKCALL
                CADR    TIMEGMBL
                RELINT
                CAF     ZERO
                TS      INGTS
                TCF     RCS

                EBANK=  CDUXD
GTSCADR         2CADR   GTS
```

```
# SUBROUTINE TO COMPUTE Q,R-AXES ATTITUDE ERRORS FOR USE IN THE RCS AND GTS CONTROL LAWS AND THE DISPLAYS.

QERRCALC        CAE     CDUY            # Q-ERROR CALCULATION
                EXTEND
                MSU     CDUYD           # CDU ANGLE -- ANGLE DESIRED (Y-AXIS)
                TS      DAPTEMP1        # SAVE FOR RERRCALC
                EXTEND
                MP      M21             # (CDUY-CDUYD)*M21 SCALED AT PI RADIANS
                TS      E
                CAE     CDUZ            # SECOND TERM CALCULATION:
                EXTEND
                MSU     CDUZD           # CDU ANGLE -ANGLE DESIRED (Z-AXIS)
                TS      DAPTEMP2        # SAVE FOR RERRCALC
                EXTEND
                MP      M22             # (CDUZ-CDUZD)*M22 SCALED AT PI RADIANS
                AD      DELQEROR        # KALCMANU INERFACE ERROR
                AD      E
                XCH     QERROR          # SAVE Q-ERROR FOR EIGHT-BALL DISPLAY.

RERRCALC        CAE     DAPTEMP1        # R-ERROR CALCULATION:
                EXTEND                  # CDU ANGLE -ANGLE DESIRED (Y-AXIS)
                MP      M31             # (CDUY-CDUYD)*M31 SCALED AT PI RADIANS
                TS      E
                CAE     DAPTEMP2        # SECOND TERM CALCULATION:
                EXTEND                  # CDU ANGLE -ANGLE DESIRED (Z-AXIS)
                MP      M32             # (CDUZ-CDUZD)*M32 SCALED AT PI RADIANS
                AD      DELRER0R        # KALCMANU INERFACE ERROR
                AD      E
                XCH     RERROR          # SAVE R-ERROR FOR EIGHT-BALL DISPLAY.
                TC      Q
```

```
# "ATTSTEER" IS THE ENTRY POINT FOR Q,R-AXES (U,V-AXES) ATTITUDE CONTROL USING THE REACTION CONTROL SYSTEM

ATTSTEER        EQUALS  STILLRCS        # "STILLRCS" IS THE RCS EXIT FROM TRYGTS.

STILLRCS        CA      RERROR
                LXCH    A
                CA      QERROR
                TC      ROT-TOUV
                DXCH    UERROR

# PREPARES CALL TO TJETLAW (OR SPSRCS(DOCKED))
# PREFORMS SKIP LOGIC ON U OR Y AXIS IF NEEDED.

TJLAW           CA      TJLAWADR
                TS      RETJADR
                CA      ONE
                TS      AXISCTR
                INDEX   AXISCTR
                INDEX   SKIPU
                TCF     +1
                CA      FOUR
                INDEX   AXISCTR
                TS      SKIPU
                TCF     LOOPER
                INDEX   AXISCTR
                CA      UERROR
                TS      E
                INDEX   AXISCTR
                CA      OMEGAU
                TS      EDOT
                CA      DAPBOOLS
                MASK    CSMDOCKD
                CCS     A
                TCF     +3
                TC      TJETLAW
                TCF     AFTERTJ
 +3             CS      DAPBOOLS        # DOCKED.  IF GIMBAL USABLE DO GTS CONTROL
                MASK    USEQRJTS        #       ON THE NEXT PASS.
                CCS     A               # USEQRJTS BIT MUST NOT BE BIT 15.
                TS      COTROLER        # GIMBAL USABLE.  STORE POSITIVE VALUE.
                INHINT
                TC      IBNKCALL
                CADR    SPSRCS          # DETERMINE RCS CONTROL
                RELINT
                CAF     FOUR            # ALWAYS CALL FOR 2-JET CONTROL ABOUT U,V.
                TS      NUMBERT         # FALL THROUGH TO JET SELECTION, ETC.

# Q,R-JET-SELECTION-LOGIC
#
# INPUT:        AXISCTR         0,1 FOR U,V
#               SNUFFBIT        ZERO TJETU,V AND TRANS. ONLY IF SET IN A DPS BURN
```

```
#                TJU,TJV          JET TIME SCALED 10.24 SEC.
#                NUMBERT          INDICATES NUMBER OF JETS AND TYPE OF POLICY
#                RETJADR          WHERE TO RETURN TO
#
# OUTPUT:        NO.U(V)JETS      RATE DERIVATION FEEDBACK
#                CHANNEL 5
#                SKIPU,SKIPV      FOR LESS THAN 150MS FIRING
#
# NOTES:         IN CASE OF FAILURE IN DESIRED ROTATION POLICY, "ALL" UNFAILED
#                JETS OF THE DESIRED POLICY ARE SELECTED.  SINCE THERE ARE ONLY
#                TWO JETS, THIS MEANS THE OTHER ONE OR NONE.  THE ALARM IS SENT
#                IF NONE CAN BE FOUND.
#
#                TIMES LESS THAN 14 MSEC ARE TAKEN TO CALL FOR A SINGLE-JET
#                MINIMUM IMPULSE, WITH THE JET CHOSEN SEMI-RANDOMLY.

AFTERTJ          CA      FLAGWRD5        # IF SNUFFBIT SET DURING A DPS BURN GO TO
                 MASK    SNUFFBIT        # XTRANS; THAT IS, INHIBIT CONTROL.
                 EXTEND
                 BZF     DOROTAT
                 CS      FLGWRD10
                 MASK    APSFLBIT
                 EXTEND
                 BZF     DOROTAT
                 CA      DAPBOOLS
                 MASK    DRIFTBIT
                 EXTEND
                 BZF     XTRANS

DOROTAT          CAF     TWO
                 TS      L
                 INDEX   AXISCTR
                 CCS     TJU
                 TCF     +5
                 TCF     NOROTAT
                 TCF     +2
                 TCF     NOROTAT
                 ZL
                 AD      ONE
                 TS      ABSTJ

                 CA      AXISCTR
                 AD      L
                 TS      ROTINDEX        # 0 1 2 3 = -U -V +U +V

                 CA      ABSTJ
                 AD      -150MS
                 EXTEND
                 BZMF    DOSKIP
```

```
                TC      SELCTSUB

                INDEX   AXISCTR
                CA      INDEXES
                TS      L

                CA      POLYTEMP
                INHINT
                INDEX   L
                TC      WRITEP

                RELINT
                TCF     FEEDBACK
NOROTAT         INDEX   AXISCTR
                CA      INDEXES
                INHINT
                INDEX   A
                TC      WRITEP  -1

                RELINT
LOOPER          CCS     AXISCTR
                TC      RETJADR
                TCF     CLOSEOUT
DOSKIP          CS      ABSTJ
                AD      +TJMINT6        # 14MS
                EXTEND
                BZMF    NOTMIN

                ADS     ABSTJ
                INDEX   AXISCTR
                CCS     TJU
                CA      +TJMINT6
                TCF     +2
                CS      +TJMINT6
                INDEX   AXISCTR
                TS      TJU

                CCS     SENSETYP        # ENSURE MIN-IMPULSE NOT AGAINST TRANS
                TCF     NOTMIN  -1
                EXTEND
                READ    LOSCALAR
                MASK    ONE
                TS      NUMBERT

NOTMIN          TC      SELCTSUB

                INDEX   AXISCTR
                CA      INDEXES
                INHINT
```

```
                TS      T6FURTHA +1
                CA      POLYTEMP
                INDEX   T6FURTHA +1
                TC      WRITEP

                CA      ABSTJ
                TS      T6FURTHA
                TC      JTLST           # IN QR BANK BY NOW

                RELINT

                CA      ZERO
                INDEX   AXISCTR
                TS      SKIPU

FEEDBACK        CS      THREE
                AD      NUMBERT
                EXTEND
                BZMF    +3

                CA      TWO
                TCF     +2
                CA      ONE
                INDEX   AXISCTR
                TS      NO.UJETS
                TCF     LOOPER

XTRANS          CA      ZERO
                TS      TJU
                TS      TJV
                CA      FOUR
                INHINT
                XCH     SKIPU
                EXTEND
                BZF     +2
                TC      WRITEU  -1
                CA      FOUR
                XCH     SKIPV
                RELINT

                EXTEND
                BZF     CLOSEOUT
                INHINT
                TC      WRITEV  -1
                RELINT

                TCF     CLOSEOUT
INDEXES         DEC     4
                DEC     13
+TJMINT6        DEC     22
```

```
-150MS          DEC     -240
BIT8,9          OCT     00600
SCLNORM         OCT     266
TJLAWADR        GENADR  TJLAW   +3      # RETURN ADDRESS FOR RCS ATTITUDE CONTROL

# THE JET LIST:
# THIS IS A WAITLIST FOR T6RUPTS.
#
# CALLED BY:
#               CA      TJ              # TIME WHEN NEXT JETS WILL BE WRITTEN
#               TS      T6FURTHA
#               CA      INDEX           # AXIS TO BE WRITTEN AT TJ (FROM NOW)
#               TS      T6FURTHA +1
#               TC      JTLST
#
# EXAMPLE -- U-AXIS AUTOPILOT WILL WRITE ITS ROTATION CODE OF
# JETS INTO CHANNEL 5.  IF IT DESIRES TO TURN OFF THIS POLICY WITHIN
# 150MS AND THEN FIRE NEXTU, A CALL TO JTLST IS MADE WITH T6FURTHA
# CONTAINING THE TIME TO TURN OFF THE POLICY, T6FURTHA +1 THE INDEX
# OF THE U-AXIS(4), AND NEXTU WILL CONTAIN THE "U-TRANS" POLICY OR ZERO.
#
# THE LIST IS EXACTLY 3 LONG.  (THIS LEADS UP TO SKIP LOGIC AND 150MS LIMIT)
# THE INPUT IS THE LAST MEMBER OF THE LIST.
#
# RETURNS BY:
#       +       TC      Q
#
# DEFINITIONS:  (OUTPUT)
#       TIME6           TIME OF NEXT RUPT
#       T6NEXT          DELTA TIME TO NEXT RUPT
#       T6FURTHA        DELTA TIME FROM 2ND TO LAST RUPT
#       NXT6ADR         AXIS INDEX      0 -- P-AXIS
#       T6NEXT +1       AXIS INDEX      4 -- U-AXIS
#       T6FURTHA +1     AXIS INDEX      13 -- V-AXIS

JTLST           CS      T6FURTHA
                AD      TIME6
                EXTEND
                BZMF    MIDORLST        # TIME6 -- TI IS IN A

                LXCH    NXT6ADR
                DXCH    T6NEXT
                DXCH    T6FURTHA
                TS      TIME6
                LXCH    NXT6ADR

TURNON          CA      BIT15
                EXTEND
                WOR     CHAN13
                TC      Q
```

```
MIDORLST        AD      T6NEXT
                EXTEND
                BZMF    LASTCHG         # TIME6 + T6NEXT - T IS IN A

                LXCH    T6NEXT  +1
                DXCH    T6FURTHA
                EXTEND
                SU      TIME6
                DXCH    T6NEXT

                TC      Q

LASTCHG         CS      A
                AD      NEGO
                TS      T6FURTHA

                TC      Q

# ROT-TOUV IS ENTERED WITH THE Q-COMPONENT OF THE QUANTITY TO BE TRANSFORMED IN A AND THE R-COMPONENT IN L.
# ROT-TOUV TRANSFORMS THE QUANTITY INTO THE NON-ORTHOGONAL U-V AXIS SYSTEM.  IN THE U-V SYSTEM NO CROSS-COUPLING IS
# PRODUCED FROM RCS JET FIRINGS.  AT THE COMPLETION OF ROT-TOUV, THE U-COMPONENT OF THE TRANSFORMED QUANTITY IS IN
# A AND THE V-COMPONENT IS IN L.

ROT-TOUV        LXCH    ROTEMP2         # (R) IS PUT INTO ROTEMP2
                EXTEND
                MP      COEFFQ
                XCH     ROTEMP2         # (R) GOES TO A AND COEFFQ.(Q) TO ROTEMP2
                EXTEND
                MP      COEFFR
                TS      L               # COEFFR.(R) IS PUT INTO L
                AD      ROTEMP2
                TS      ROTEMP1         # COEFFQ.(Q)+COEFFR.(R) IS PUT IN ROTEMP1
                TCF     +4
                INDEX   A               # COEFFQ.(Q) + COEFFR.(R) HAS OVERFLOWED
                CS      LIMITS          # AND IS LIMITED TO POSMAX OR NEGMAX
                TS      ROTEMP1
                CS      ROTEMP2
                AD      L               # -COEFFQ.(Q) + COEFFR.(R) IS NOW IN A
                TS      7
                TCF     +3
                INDEX   A               # -COEFFQ.(Q) + COEFFR.(R) HAS OVERFLOWED
                CS      LIMITS          # AND IS LIMITED TO POSMAX OR NEGMAX
                LXCH    ROTEMP1         # COEFFQ.(Q) + COEFFR.(R) IS PUT INTO L
                TC      Q
SELCTSUB        INDEX   ROTINDEX
                CA      ALLJETS
                INDEX   NUMBERT
                MASK    TYPEPOLY
                TS      POLYTEMP
```

```
                MASK    CH5MASK
                CCS     A
                TCF     +2

                TC      Q

                CA      THREE
FAILOOP         TS      NUMBERT
                INDEX   ROTINDEX
                CA      ALLJETS
                INDEX   NUMBERT
                MASK    TYPEPOLY
                TS      POLYTEMP
                MASK    CH5MASK
                EXTEND
                BZF     FAILOOP -2
                CCS     NUMBERT
                TCF     FAILOOP
                INDEX   AXISCTR
                TS      TJU
                TC      ALARM
                OCT     02004
                TCF     NOROTAT
ALLJETS         OCT     00110           #       -U      6 13
                OCT     00022           #       -V      2 9
                OCT     00204           #       +U      5 14
                OCT     00041           #       +V      1 10
TYPEPOLY        OCT     00125           #       -X      1 5 9 13
                OCT     00252           #       +X      2 6 10 14
                OCT     00146           #       A       2 5 10 13
                OCT     00231           #       B       1 6 9 14
                OCT     00377           #       ALL     1 2 5 6 9 10 13 14

# THE FOLLOWING SETS THE INTERRUPT FLIP-FLOP AS SOON AS POSSIBLE, WHICH PERMITS A RETURN TO THE INTERRUPTED JOB.

CLOSEOUT        CA      ADRRUPT
                TC      MAKERUPT

ADRRUPT         ADRES   ENDJASK

ENDJASK         DXCH    DAPARUPT
                DXCH    ARUPT
                DXCH    DAPBQRPT
                XCH     BRUPT
                LXCH    Q
                CAF     NEGMAX          # NEGATIVE DAPZRUPT SIGNALS JASK IS OVER.
                DXCH    DAPZRUPT
                DXCH    ZRUPT
                TCF     NOQRSM
```

```
                BLOCK   3
                SETLOC  FFTAG6
                BANK

                COUNT*  $$/DAP

MAKERUPT        EXTEND
                EDRUPT  MAKERUPT
```

```
# PROGRAM DESCRIPTION
# DESIGNED BY:  R. D. GOSS AND P. S. WEISSMAN
# CODED BY:  P. S. WEISSMAN, 28 FEBRUARY 1968
#
# TJETLAW IS CALLED AS A SUBROUTINE WHEN THE LEM IS NOT DOCKED AND THE AUTOPILOT IS IN THE AUTOMATIC OR
# ATTITUDE-HOLD MODE TO CALCULATE THE JET-FIRING-TIME (TJET) REQUIRED FOR THE AXIS INDICATED BY AXISCTR:
#       -1      INDICATES THE P-AXIS
#       +0      INDICATES THE U-AXIS
#       +1      INDICATES THE V-AXIS
# THE REGISTERS E AND EDOT CONTAIN THE APPROPRIATE ATTITUDE ERROR AND ERROR RATE AND SENSETYP SHOWS WHETHER
# UNBALANCED COUPLES ARE PREFERRED.  TJETLAW ALSO USES VARIOUS FUNCTIONS OF ACCELERATION AND DEADBAND WHICH ARE
# COMPUTED IN THE 1/ACCONT SECTION OF 1/ACCS AND ARE STORED IN SUCH AN ORDER THAT THEY CAN BE CONVENIENTLY
# ACCESSED BY INDEXING.
#
# THE SIGN OF THE REQUIRED ROTATION IS CARRIED THROUGH TJETLAW AS ROTSENSE AND IS FINALLY APPLIED TO TJET JUST
# PREVIOUS TO ITS STORAGE IN THE LOCATION CORRESPONDING TO THE AXIS (TJP, TJU, OR TJV).  THE NUMBER OF JETS THAT
# TJETLAW ASSUMES WILL BE USED AS INDICATED BY THE SETTING OF NUMBERT FOR THE U- OR V-AXIS.  TWO JETS ARE ALWAYS
# ASSUMED FOR THE P-AXIS ALTHOUGH FOUR JETS WILL BE FIRED WHEN FIREFCT IS MORE NEGATIVE THAN -4.0 DEGREES
# (FIREFCT IS THE DISTANCE TO A SWITCH CURVE IN THE PHASE PLANE) AND A LONG FIRING IS CALLED FOR.
#
# IN ORDER TO AVOID SCALING DIFFICULTIES, SIMPLE ALGORITHMS TAGGED RUFLAW1, -2 AND -3 ARE RESORTED TO WHEN THE
# ERROR AND/OR ERROR RATE ARE LARGE.
#
# CALLING SEQUENCE:
#               TC      TJETLAW         # (MUST BE IN JASK)
#       OR
#               INHINT                  # (MUST BE IN JASK)
#               TC      IBNKCALL
#               CADR    TJETLAW
#               RELINT
#
# EXIT:         RETURN TO Q.
#
# INPUT:
#       FROM THE CALLER:  E, EDOT, AXISCTR, SENSETYP, TJP, -U, -V.
#       FROM 1/ACCONT:  48 ERASABLES BEGINNING AT BLOCKTOP (INCLUDING FLAT, ZONE3LIM AND ACCSWU, -V).
#
# OUTPUT:
#       TJP, -U OR -V, NUMBERT (DAPTEMP5), FIREFCT (DAPTEMP3).
#
# DEBRIS:
#       A, L, Q, E, EDOT, DAPTEMP1-6, DAPTEMP1-4.
#
# ALARM:  NONE

                BANK    17
                SETLOC  DAPS2
                BANK
                EBANK=  TJP
```

```
                COUNT*  $$/DAPTJ

TJETLAW         EXTEND                  # SAVE Q FOR RETURN.
                QXCH    HOLDQ

# SET INDEXERS TO CORRESPOND TO THE AXIS AND TO THE SIGN OF EDOT

                INDEX   AXISCTR         # AXISDIFF(-1)=NO OF LOCATIONS BET P AND U
                CAF     AXISDIFF        # AXISDIFF(0)=0
                TS      ADRSDIF1        # AXISDIFF(+1)=NO OF LOCATIONS BET V AND U

                CAE     EDOT            # IF EDOT NEGATIVE, PICK UP SET OF VALUES
                EXTEND                  #       THAT ALLOW USE OF SAME CODING AS FOR
                BZMF    NEGEDOT         #       POSITIVE EDOT.
                CAE     ADRSDIF1        # SET A SECOND INDEXER WHICH MAY BE
                TS      ADRSDIF2        #       MODIFIED BY A DECISION FOR MAX JETS.
                CAF     SENSOR          # FOR POSITIVE EDOT, ROTSENSE IS
                TCF     SETSENSE        #       INITIALIZED POSITIVE.

NEGEDOT         CS      E               # IN ORDER FOR NEG EDOT CASE TO USE CODING
                TS      E               #       OF POS EDOT, MUST MODIFY AS FOLLOWS:
                CS      EDOT            #       1. COMPLEMENT E AND EDOT.
                TS      EDOT            #       2. SET SENSE OF ROTATION TO NEGATIVE
                CAF     BIT1            #          (REVERSED LATER IF NECESSARY).
                ADS     ADRSDIF1        #       3. INCREMENT INDEXERS BY ONE SO THAT
                TS      ADRSDIF2        #          THE PROPER PARAMETERS ARE ACCESSED.
                CS      SENSOR
SETSENSE        TS      ROTSENSE

# TEST MAGNITUDE OF E (ATTITUDE ERROR, SINGLE-PRECISION, SCALED AT PI RADIANS):
#       IF GREATER THAN (OR EQUAL TO) PI/16 RADIANS, GO TO THE SIMPLIFIED TJET ROUTINE.
#       IF LESS THAN PI/16 RADIANS, RESCALE TO PI/4

                CAE     E               # PICK UP ATTITUDE ERROR FOR THIS AXIS
                EXTEND
                MP      BIT5            # SHIFT RIGHT TEN BITS:  IF A-REGISTER IS
                CCS     A               #       ZERO, RESCALE AND TEST EDOT.
                TCF     RUFLAW2
                TCF     SCALEE
                TCF     RUFLAW1
SCALEE          CAF     BIT13           # ERROR IS IN L SCALED AT PI/16.  RESCALE
                EXTEND                  #       IT TO PI/4 AND SAVE IT.
                MP      L
                TS      E

# TEST MAGNITUDE OF EDOT (ERROR RATE SCALED AT PI/4 RADIANS/SECOND)
#       IF GREATER THAN (OR EQUAL TO) PI/32 RADIANS/SECOND, GO TO THE SIMPLIFIED TJET ROUTINE.
#       IF LESS THAN PI/32 RADIANS/SECOND, THEN RESCALE TO PI/32 RADIANS/SECOND.

                CAE     EDOT            # PICK UP SINGLE-PRECISION ERROR-RATE
```

```
                EXTEND                  # FOR THIS AXIS=
                MP      BIT4            # SHIFT RIGHT ELEVEN BITS, IF THE A-REG IS
                EXTEND                  # ZERO, THEN RESCALE AND USE FINELAW.
                BZF     SCALEDOT
                TCF     RUFLAW3

# *** FINELAW STARTS HERE ***

SCALEDOT        LXCH    EDOT            # EDOT IS SCALED AT PI/32 RADIANS/SECOND.

                CAE     EDOT            # COMPUTE (EDOT)(EDOT)
                EXTEND
                SQUARE                  # PRODUCT SCALED AT PI(2)/2(10) RAD/SEC.
                EXTEND
                MP      BIT13           # SHIFT RIGHT TWO BITS TO RESCALE TO EDOTSQ
                TS      EDOTSQ          #       TO PI(2)/2(8) RAD(2)/SEC(2).

ERRTEST         CCS     E               # DOES BIG ERROR (THREE DEG BEYOND THE
                AD      -3DEG           # DEADBAND) REQUIRE MAXIMUM JETS?
                TCF     +2
                AD      -3DEG
                EXTEND
                INDEX   ADRSDIF1
                SU      FIREDB
                EXTEND
                BZMF    SENSTEST        # IF NOT:  ARE UNBALANCED JETS PREFERRED?
MAXJETS         CAF     TWO             # IF YES:  INCREMENT ADDRESS LOCATOR AND
                ADS     ADRSDIF2        #          SET SWITCH FOR JET SELECT LOGIC TO 4.
                CAF     FOUR            #          (ALWAYS DO THIS FOR P-AXIS)
                TCF     TJCALC
SENSTEST        CCS     SENSETYP        # DOES TRANSLATION PREFER MIN JETS.
                TCF     TJCALC          # YES.  USE MIN-JET PARAMETERS
                TCF     MAXJETS         # NO.  GET THE MAX-JET PARAMETERS.
TJCALC          TS      NUMBERT         # SET TO +0,1,4 FOR (U,V-AXES) JET SELECT.

# BEGINNING OF TJET CALCULATIONS:

                CS      EDOTSQ          # SCALED AT PI(2)/2(8).
                EXTEND
                INDEX   ADRSDIF2
                MP      1/ANET1         # .5/ACC SCALED AT 2(6)/PI SEC(2)/RADIAN.
                INDEX   ADRSDIF1
                AD      FIREDB          # DEADBAND SCALED AT PI/4 RADIAN.
                EXTEND
                SU      E               # ATTITUDE ERROR SCALED AT PI/4 RADIAN.
                TS      FIREFCT         # -E-.5(EDOTSQ)/ACC-DB AT PI/4 RADIAN.
                EXTEND
                BZMF    ZON1,2,3

ZONE4,5         INDEX   ADRSDIF1
                CAE     1/ACOAST        # .5/ACC SCALED AT 2(6)/PI WHERE
```

```
                EXTEND                  # ACC = MAX(AMIN, AOS-).
                MP      EDOTSQ          # SCALED AT PI/2(8).
                AD      E               # SCALED AT PI/4
                INDEX   ADRSDIF1
                AD      COASTDB         # SCALED AT PI/4 POS. FOR NEG. INTERCEPT.
                EXTEND                  # TEST E+.5(EDOTSQ)/ACC+DB AT PI/4 RADIAN.
                BZMF    ZONE5           # IF FUNCTION NEGATIVE, FIND TJET.
                                        # IF FUNCTION POSITIVE, IN ZONE 4.

# ZONE 4 IS THE COAST REGION.  HOWEVER, IF THE JETS ARE ON AND DRIVING TOWARD
#       A. THE AXIS WITHIN + OR - (DB + FLAT) FOR DRIFTING FLIGHT, OR
#       B. THE USUAL TARGET PARABOLA FOR POWERED FLIGHT
# THEN THE THRUSTERS ARE KEPT ON.

ZONE4           INDEX   AXISCTR         # IS THE CURRENT VALUE IN TJET NON-ZERO
                CS      TJETU           #       WITH SENSE OPPOSITE TO EDOT,
                EXTEND                  #       (I.E., ARE JETS ON AND FIRING TOWARD
                MP      ROTSENSE        #       THE DESIRABLE STATE).
                EXTEND
                BZMF    COASTTJ         # NO.  COAST.

JETSON          CCS     FLAT            # YES.  IS THIS DRIFTING OR POWERED FLIGHT?
                TCF     DRIFT/ON        # DRIFTING.  GO MAKE FURTHER TEST.

                CS      FIREFCT         # POWERED (OR ULLAGE).  CAN TARGET PARABOLA
                INDEX   ADRSDIF1        #       BE REACHED FROM THIS POINT IN THE
                AD      AXISDIST        #       PHASE PLANE?
                EXTEND
                BZMF    COASTTJ         # NO. SET TJET = 0.
                TC      Z123COMP        # YES.  CALCULATE TJET AS THOUGH IN ZONE 1
                CAE     FIREFCT         #       AFTER COMPUTING THE REQUIRED
                TCF     ZONE1           #       PARAMETERS.

DRIFT/ON        INDEX   ADRSDIF1        # CAN TARGET STRIP OF AXIS BE REACHED FROM
                CS      FIREDB          #       THIS POINT IN THE PHASE PLANE?
                DOUBLE
                AD      FIREFCT
                EXTEND
                BZMF    +3
COASTTJ         CAF     ZERO            # NO.  SET TJET = 0.
                TCF     RETURNTJ

                TC      Z123COMP        # YES. CALCULATE TJET AS THOUGH IN ZONE 2
                TCF     ZONE2,3         #       OR 3 AFTER COMPUTING REQUIRED VALUES.

ZONE5           TS      L               # TEMPORARILY STORE FUNCTION IN L.
                CCS     ROTSENSE        # MODIFY ADRSDIF2 FOR ACCESSING 1/ANET2
                TCF     +4              # AND ACCFCTZ5, WHICH MUST BE PICKED UP
                TC      CCSHOLE         # FROM THE NEXT LOWER REGISTER IF THE
                CS      TWO             # (ACTUAL) ERROR RATE IS NEGATIVE.
```

```
                ADS     ADRSDIF2

 +4             CAE     L
                EXTEND
                INDEX   ADRSDIF2        # TTOAXIS AND HH ARE THE PARAMETERS UPON
                MP      ACCFCTZ5        #       WHICH THE APPROXIMATIONS TO TJET ARE
                DDOUBL                  #       ABASED.
                DDOUBL
                DXCH    HH              # DOUBLE PRECISION H SCALED AT 8 SEC(2).
                INDEX   ADRSDIF2
                CAE     1/ANET2         # SCALED AT 2(7)/PI SEC(2)/RAD.
                EXTEND
                MP      EDOT            # SCALED AT PI/2(5)
                TS      TTOAXIS         # SCALED AT 4 SEC.

# TEST WHETHER TJET GREATER THAN 50 MSEC.

                EXTEND
                MP      -.05AT2         # H - .05 TTOAXIS - .00125 G.T. ZERO
                AD      HH              #       (SCALED AT 8 SEC(2) ).
                AD      NEG2
                EXTEND
                BZMF    FORMULA1

# TEST WHETHER TJET GREATER THAN 150 MSEC.

                CAE     TTOAXIS
                EXTEND
                MP      -.15AT2         # H - .15 TTOAXIS - .01125 G.T. ZERO
                AD      HH              #       (SCALED AT 8 SEC(2) )
                AD      -.0112A8
                EXTEND
                BZMF    FORMULA2

# IF TJET GREATER THAN 150 MSEC, ASSIGN IT VALUE OF 250 MSEC, SINCE THIS
# IS ENOUGH TO ASSURE NO SKIP NEXT CSP (100 MSEC).

FULLTIME        CAF     BIT11           # 250 MSEC SCALED AT 4 SEC.

# RETURN TO CALLING PROGRAM WITH JET TIME SCALED AS TIME6 AND SIGNED.

RETURNTJ        EXTEND                  # ALL BRANCHES TERMINATE HERE WITH TJET
                MP      ROTSENSE        #       (SCALED AT 4 SEC) IN THE ACCUMULATOR.
                INDEX   AXISCTR         # ROTSENSE APPLIES SIGN AND CHANGES SCALE.
                TS      TJETU
                EXTEND
                INDEX   AXISCTR
                MP      ACCSWU          # SET SWITCH FOR JET SELECT IF ROTATION IS
                CAE     L
                EXTEND                  #       IN A SENSE FOR WHICH 1/ACCS HAS FORCED
                BZMF    +3              #       A MAX-JET CALCULATION.
                CAF     FOUR
```

```
                TS      NUMBERT
                TC      HOLDQ           # RETURN VIA SAVED Q.

# TJET = H/(.025 + TTOAXIS)     FOR TJET LESS THAN 50 MSEC.

FORMULA1        CS      -.025AT4        # .025 SEC SCALED AT 4.
                AD      TTOAXIS         # SCALED AT 4 SECONDS.
                DXCH    HH              # STORE DENOMINATOR IN FIRST WORD OF H,
                EXTEND                  #       WHICH NEED NOT BE PRESERVED.  PICK UP
                DV      HH              #       DP H AND DIVIDE BY DENOMINATOR.
                EXTEND
                MP      BIT14           # RESCALE TJET FROM 2 TO USUAL 4 SEC.
                TCF     CHKMINTJ        # CHECK THAT TJET IS NOT LESS THAN MINIMUM

# TJET = (H + .00375)/(0.1 + TTOAXIS)   FOR TJET GREATER THAN 50 MSEC.

FORMULA2        EXTEND
                DCA     .00375A8        # .00375 SEC(2) SCALED AT 8.
                DAS     HH              # STORE NUMERATOR IN DP H, WHICH NEED NOT
                                        #       BE PRESERVED.
                CAE     TTOAXIS         # SCALED AT 4 SEC.
                AD      .1AT4           # 0.1 SEC SCALED AT 4.
                DXCH    HH              # STORE DENOMINATOR IN FIRST WORD OF H,
                EXTEND                  #       WHICH NEED NOT BE PRESERVED.  PICK UP
                DV      HH              #       DP NUMERATOR AND DIVIDE BY DENOMINATOR
                EXTEND
                MP      BIT14           # RESCALE TJET FROM 2 TO USUAL 4 SEC.
                TCF     RETURNTJ        # END SUBROUTINE.

# SUBROUTINIZED COMPUTATIONS REQUIRED FOR ALL ENTRIES INTO CODING FOR ZONES 1, 2, AND 3.
# REACHED BY TC FROM 3 POINTS IN TJETLAW.

Z123COMP        CS      ROTSENSE        # USED IN RETURNTJ SECTION TO RESCALE TJET
                TS      ROTSENSE        #       AS TIME6 AND GIVE IT PROPER SIGN.
                CAE     EDOT            # SCALED AT PI/2(5) RAD/SEC.
                EXTEND
                INDEX   ADRSDIF2
                MP      1/ANET1         # SCALED AT 2(7)/PI SEC(2)/RAD.
                TS      TTOAXIS         # STORE TIME-TO-AXIS SCALED AT 4 SECONDS.
                AD      -TJMAX
                EXTEND                  # IS TIME TO AXIS LESS THAN 150 MSEC.
                BZMF    +2
                TCF     FULLTIME        # NO. FIRE JETS, DO NOT CALCULATE TJET.
                RETURN                  # YES.  GO ON TO FIND TJET

ZON1,2,3        TC      Z123COMP        # SUBROUTINIZED PREPARATION FOR ZONE1,2,3.

# IF THE (NEG) DISTANCE BEYOND PARABOLA IS LESS THAN FLAT, USE SPECIAL
# LOGIC TO ACQUIRE MINIMUM IMPULSE LIMIT CYCLE.  DURING POWERED FLIGHT
```

```
# OR ULLAGE, FLAT = 0

                CAE     FIREFCT         # SCALED AT PI/4 RAD.
                AD      FLAT
                EXTEND
                BZMF    ZONE1           # NOT IN SPECIAL ZONES.

# FIRE FOR AXIS OR, IF CLOSE, FIRE MINIMUM IMPULSE.  IF ON AXIS, COAST.

ZONE2,3         CS      ZONE3LIM        # HEIGHT OF MIN-IMPULSE ZONE SET BY 1/ACCS
                AD      TTOAXIS         #       35 MSEC IN DRIFTING FLIGHT
                EXTEND                  #       ZERO WHEN TRYING TO ENTER GTS CONTROL.
                BZMF    ZONE3
ZONE2           CAE     TTOAXIS         # FIRE TO AXIS.
                TCF     RETURNTJ
ZONE3           CCS     EDOT            # CHECK IF EDOT IS ZERO.
                CAF     BIT6            # FIRE A ONE-JET MINIMUM IMPULSE.
                TCF     RETURNTJ        # TJET = +0.
                TC      CCSHOLE         # CANNOT BE BECAUSE NEG EDOT COMPLEMENTED.
                TCF     RETURNTJ        # TJET = +0.

ZONE1           EXTEND
                INDEX   ADRSDIF1
                SU      AXISDIST        # SCALED AT PI/4 RAD.
                EXTEND
                INDEX   ADRSDIF2
                MP      ACCFCTZ1        # SCALED AT 2(7)/PI SEC(2)/RAD.
                DDOUBL
                DDOUBL
                DXCH    HH              # DOUBLE PRECISION H SCALED AT 8 SEC(2).

# TEST WHETHER TOTAL TIME REQUIRED GREATER THAN 150 MSEC:
#                           2                                    2
#       IS .5(.150 - TTOAXIS)  - H  NEGATIVE (SCALED AT 8 SECONDS )

                CAE     TTOAXIS         # TTOAXIS SCALED AT 4 SECONDS.
                AD      -TJMAX          # -.150 SECOND SCALED AT 4.
                EXTEND
                SQUARE
                EXTEND
                SU      HH              # HIGH WORD OF H SCALED AT 8 SEC(2).
                EXTEND
                BZMF    FULLTIME        # YES.  NEED NOT CALCULATE TJET.

# TEST WHETHER TIME BEYOND AXIS GREATER THAN 50 MSEC TO DETERMINE WHICH APPROXIMATION TO USE.

                CAE     HH
                AD      NEG2
                EXTEND
                BZMF    FORMULA3
```

```
# TJET = H/0.1 + TTOAXIS + .0375         FOR APPROXIMATION OVER MORE THAN 50 MSEC.

                CAF     .1AT2           # STORE .1 SEC SCALED AT 2 FOR DIVISION.
                DXCH    HH              # DP H SCALED AT 8 SEC(2) NEED NOT BE
                EXTEND                  #       PRESERVED.
                DV      HH              # QUOTIENT SCALED AT 4 SECONDS.
                AD      TTOAXIS         # SCALED AT 4 SEC.
                AD      .0375AT4        # .0375 SEC SCALED AT 4.
                TCF     RETURNTJ        # END COMPUTATION.

# TJET - H/.025 + TTOAXIS       FOR APPROXIMATION OVER LESS THAN 50 MSEC.

FORMULA3        CS      -.025AT2        # STORE +.25 SEC SCALED AT 2 FOR DIVISION
                DXCH    HH              # PICK UP DP H AT 8, WHICH NEED NOT BE
                EXTEND                  #       PRESERVED.
                DV      HH              # QUOTIENT SCALED AT 4 SECONDS.
                AD      TTOAXIS         # SCALED AT 4 SEC.

# IF COMPUTED JET TIME IS LESS THAN TJMIN, TJET IS SET TO ZERO.
# MINIMUM IMPULSES REQUIRED IN ZONE 3 ARE NOT SUBJECT TO THIS CONSTRAINT, NATURALLY.

CHKMINTJ        AD      -TJMIN          # IS COMPUTED TIME LESS THAN THE MINIMUM.
                EXTEND
                BZMF    COASTTJ         # YES, SET TIME TO ZERO.
                AD      TJMIN           # NO, RESTORE COMPUTED TIME.
                TCF     RETURNTJ        # END COMPUTATION.
```

```
# *** ROUGHLAW ***
#
# BEFORE ENTRY TO RUFLAW:
#       1. INDEXERS ADRSDIF1 AND ADRSDIF2 ARE SET ON BASIS OF AXIS, AND SIGN OF EDOT.
#       2. IF EDOT WAS NEGATIVE, E AND EDOT ARE ROTATED INTO UPPER HALF-PLANE AND ROTSENSE IS MADE NEGATIVE.
#       3. E IS SCALED AT PI RADIANS AND EDOT AT PI/4 RAD/SEC.
#          (EXCEPT THE RUFLAW3 ENTRY WHEN E IS AT PI/4)
#
# RUFLAW1:      ERROR MORE NEGATIVE THAN PI/16 RAD.  FIRE TO A RATE OF 6.5 DEG/SEC (IF JET TIME EXCEEDS 20 MSEC.).
# RUFLAW2:      ERROR MORE POSITIVE THAN PI/16 RAD.  FIRE TO AN OPPOSING RATE OF 6.5 DEG/SEC.
# RUFLAW3:      ERROR RATE GREATER THAN PI/32 RAD/SEC AND ERROR WITHIN BOUNDS.  COAST IF BELOW FIREFCT, FIRE IF ABOVE

RUFLAW1         CS      RUFRATE         # DECREMENT EDOT BY .1444 RAD/SEC AT PI/4
                ADS     EDOT            #       WHICH IS THE TARGET RATE
                EXTEND
                BZMF    SMALRATE        # BRANCH IF RATE LESS THAN TARGET.
                TC      RUFSETUP        # REVERSE ROTSENSE AND INDICATE MAX JETS.
                CAE     EDOT            # PICK UP DESIRED RATE CHANGE.

RUFLAW12        EXTEND                  # COMPUTE TJET
                INDEX   ADRSDIF2        #       = (DESIRED RATE CHANGE)/(2-JET ACCEL.)
                MP      1/ANET1 +2
                AD      -1/8            # IF TJET, SCALED AT 32 SEC, EXCEEDS
                EXTEND                  #       4 SECONDS, SET TJET TO TJMAX.
                BZMF    +2
                TCF     FULLTIME
                EXTEND
                BZF     FULLTIME
                AD      BIT12           # RESTORE COMPUTED TJET TO ACCUMULATOR
                DAS     A
                DAS     A
                DAS     A               # RESCALED TJET AT 4 SECONDS.
                TCF     CHKMINTJ        # RETURN AS FROM FINELAW.

SMALRATE        TC      RUFSETUP +2     # SET NUMBERT AND FIREFCT FOR MAXIMUM JETS
                CCS     ROTSENSE
                CAF     ONE             # MODIFY INDEXER TO POINT TO 1/ANET
                TCF     +2              #       CORRESPONDING TO THE PROPER SENSE.
                CAF     NEGONE
                ADS     ADRSDIF2

                CS      EDOT            # (.144 AT PI/4 - EDOT) = DESIRED RATE CHNG.
                TCF     RUFLAW12

RUFLAW2         TC      RUFSETUP        # REVERSE ROTSENSE AND INDICATE MAX JETS.
                CAF     RUFRATE
                AD      EDOT            # (.144 AT PI/4 + EDOT) = DESIRED RATE CHNG.
                TS      A               # IF OVERFLOW SKIP, FIRE FOR FULL TIME.
                TCF     RUFLAW12        # OTHERWISE, COMPUTE JET TIME.
                TCF     FULLTIME
```

```
RUFLAW3         TC      RUFSETUP        # EXECUTE COMMON RUFLAW SUBROUTINE.
                INDEX   ADRSDIF1
                CS      FIREDB          # CALCULATE DISTANCE FROM SWITCH CURVE
                AD      E               #       1/ANET1*EDOT*EDOT +E - FIREDB = 0
                EXTEND                  #               SCALED AT 4 PI RADIANS
                MP      BIT11
                XCH     EDOT
                EXTEND
                SQUARE
                EXTEND
                INDEX   ADRSDIF1
                MP      1/ANET1 +2
                AD      EDOT
                EXTEND
                BZMF    COASTTJ         # COAST IF BELOW IT.
                TCF     FULLTIME        # FIRE FOR FULL PERIOD IF ABOVE IT.

# SUBROUTINE USED IN ALL ENTRIES TO ROUGHLAW.

RUFSETUP        CS      ROTSENSE        # REVERSE ROTSENSE WHEN ENTER HERE.
                TS      ROTSENSE
 +2             CAF     FOUR            # REQUIRE MAXIMUM (2) JETS IN U,V-AXES.
                TS      NUMBERT
                CAF     NEGMAX          # SUGGEST MAXIMUM (4) JETS IN P-AXIS.
                TS      FIREFCT
                TC      Q

# CONSTANTS FOR TJETLAW

                DEC     -16             # AXISDIFF(INDEX) = NUMBER OF REGISTERS
AXISDIFF        DEC     +0              #       BETWEEN STORED 1/ACCS PARAMETERS FOR
                DEC     16              #       THE INDEXED AXIS AND THE U-AXIS.
SENSOR          OCT     14400           # RATIO OF TJET SCALING WITHIN TJETLAW
                                        #       (4 SEC) TO SCALING FOR T6 (10.24 SEC).
-3DEG           DEC     -.06667         # -3.0 DEGREES SCALED AT 45.
-.0112A8        DEC     -.00141         # -.01125 SEC(2) SCALED AT 8.
.1AT4           DEC     .025            # 0.1 SECOND SCALED AT 4.
.1AT2           DEC     .05             # .1 SEC SCALED AT 2.
.0375AT4        DEC     .00938          # .0375 SEC SCALED AT 4.
-.025AT2        DEC     -.0125          # -.025 SEC SCALED AT 2.
-.025AT4        DEC     -.00625
-.05AT2         DEC     -.025
-.15AT2         DEC     -.075
.00375A8        2DEC    .00375 B-3

-TJMAX          DEC     -.0375          # LARGEST CALCULATED TIME.  .150 SEC AT 4.
TJMIN           DEC     .005            # SMALLEST ALLOWABLE TIME.  .020 SEC AT 4.
-TJMIN          DEC     -.005
RUFRATE         DEC     .1444           # CORRESPONDS TO TARGET RATE OF 6.5 DEG/S.
```

```
                EBANK=  NO.UJETS
                BANK    16
                SETLOC  DAPS1
                BANK

                COUNT*  $$/DAP

RATELOOP        CA      TWO
                TS      DAPTEMP6
                DOUBLE
                TS      Q
                INDEX   DAPTEMP6
                CCS     TJP
                TCF     +2
                TCF     LOOPRATE
                AD      -100MST6
                EXTEND
                BZMF    SMALLTJU
                INDEX   DAPTEMP6
                CCS     TJP
                CA      -100MST6
                TCF     +2
                CS      -100MST6
                INDEX   DAPTEMP6
                ADS     TJP
                INDEX   DAPTEMP6
                CCS     TJP
                CS      -100MS          # 0.1 AT 1
                TCF     +2
                CA      -100MS
LOOPRATE        EXTEND
                INDEX   DAPTEMP6
                MP      NO.PJETS
                CA      L
                INDEX   DAPTEMP6
                TS      DAPTEMP1        # SIGNED TORQUE AT 1 JET-SEC FOR FILTER
                EXTEND
                MP      BIT10           # RESCALE TO 32; ONE BIT ABOUT 2 JET-MSEC
                EXTEND
                BZMF    NEGTORK
STORTORK        INDEX   Q               # INCREMENT DOWNLIST REGISTER.
                ADS     DOWNTORK        #   NOTE: NOT INITIALIZED; OVERFLOWS.

                CCS     DAPTEMP6
                TCF     RATELOOP +1
                TCF     ROTORQUE
SMALLTJU        CA      ZERO
                INDEX   DAPTEMP6
                XCH     TJP
                EXTEND
```

```
                MP      ELEVEN          # 10.24 PLUS
                CA      L
                TCF     LOOPRATE
ROTORQUE        CA      DAPTEMP2
                AD      DAPTEMP3
                EXTEND
                MP      1JACCR
                TS      JETRATER
                CS      DAPTEMP3
                AD      DAPTEMP2
                EXTEND
                MP      1JACCQ
                TS      JETRATEQ
                TCF     BACKP
-100MST6        DEC     -160

NEGTORK         COM
                INCR    Q
                TCF     STORTORK
```

```
                BANK    21
                EBANK=  QDIFF
                SETLOC  DAPS4
                BANK

                COUNT*  $$/DAPGT

# CONTROL REACHES THIS POINT UNDER EITHER OF THE FOLLOWING TWO CONDITIONS ONCE THE DESCENT ENGINE AND THE DIGITAL
# AUTOPILOT ARE BOTH ON:
#       A) THE TRIM GIMBAL CONTROL LAW WAS ON DURING THE PREVIOUS Q,R-AXIS TIME5 INTERRUPT (OR THE DAPIDLER
#          INITIALIZATION WAS SET FOR TRIM GIMBAL CONTROL AND THIS IS THE FIRST PASS), OR
#       B) THE Q,R-AXES RCS AUTOPILOT DETERMINED THAT THE VEHICLE WAS ENTERING (OR HAD JUST ENTERED) A COAST
#          ZONE WITH A SMALL OFFSET ANGULAR ACCELERATION.
# GTS IS THE ENTRY TO THE GIMBAL TRIM SYSTEM FOR CONTROLLING ATTITUDE ERRORS AND RATES AS WELL AS ACCELERATIONS.

GTS             CAF     NEGONE          # MAKE THE NEXT PASS THROUGH THE DAP BE
                TS      COTROLER        #   THROUGH RCS CONTROL,
                CAF     FOUR            #   AND ENSURE THAT IT IS NOT A SKIP.
                TS      SKIPU
                TS      SKIPV

                CAF     TWO
                TS      INGTS           # SET INDICATOR OF GTS CONTROL POSITIVE.
                TS      QGIMTIMR        # SET TIMERS TO 200 MSEC TO AVOID BOTH
                TS      RGIMTIMR        # RUNAWAY AND INTERFERENCE BY NULLING.

# THE DRIVE SETTING ALGORITHM
#
# DEL = SGN(OMEGA + ALPHA*ABS(ALPHA)/(2*K))
#                                  2               1/2                  2       3/2
# NEGUSUM = ERROR*K + ALPHA*(DEL*OMEGA + ALPHA /(3*K)) + DEL*K   (DEL*OMEGA + ALPHA /(2*K))
#
# DRIVE = -SGN(NEGUSUM)

                CA      SR              # SAVE THE SR.  SHIFT IT LEFT TO CORRECT
                AD      A               # FOR THE RIGHT SHIFT DUE TO EDITING.
                TS      SAVESR

GTSGO+ON        CAF     TWO             # SET INDEXER FOR R-AXIS CALCULATIONS.
                TCF     GOQTRIMG +1

GOQTRIMG        CAF     ZERO            # SET INDEXER FOR Q-AXIS CALCULATIONS
                TS      QRCNTR
```

```
# RSB 2009 ----------------------------------------------------------------------
# EVERYTHING BETWEEN THIS LINE AND THE SIMILAR LINE BELOW WAS SIMPLY FILLED-IN
# AS-IS FROM LUMINARY 131, AND THEN VERIFIED TO ASSEMBLE TO THE PROPER BINARY
# VALUES.  THIS AREA IS BLANK ON THE LUMINARY 099 PRINT-OUT, AS IF THE PRINTER
# RIBBON HAD RUN OUT.
                INDEX   QRCNTR          # AOS SCALED AT PI/2
                CA      AOSQ
                EXTEND
                MP      BIT2            # RESCALE AOS TO PI/4
                EXTEND
                BZF     GTSQAXIS -3     # USE FULL SCALE FOR LARGER AOS ESTIMATES.

                INDEX   A
                CS      LIMITS          # LIMITS +1 CONTAINS NEGMAX.
                XCH     L               # LIMITS -1 CONTAINS POSMAX.

                CCS     QRCNTR          # PICK UP RATE FOR THIS AXIS.  RATE CELLS
                INDEX   A               # USE ADJACENT, NOT SEPARATED.  AT PI/4
                CA      EDOTQ
GTSQAXIS        DXCH    WCENTRAL

                INDEX   QRCNTR          # COLLECT K FOR THIS AXIS
                CA      KQ
                TS      KCENTRAL

                EXTEND                  # CONTROL AUTHORITY ZERO.  AVOID DRIVING
                BZF     POSDRIVE +1     # ENGINE BELL TO THE STOPS.

                INDEX   QRCNTR          # QDIFF, RDIFF ARE STORED IN D.P.
                CAE     QDIFF

ALGORTHM        EXTEND                  # Q(R)DIFF IS THETA (ERROR) SCALED AT PI.
                MP      KCENTRAL        # FORM K*ERROR AT PI(2)/2(8), IN D.P.
                LXCH    K2THETA
                EXTEND
                MP      BIT5            # RESCALE TO 4*PI(2)
                DXCH    K2THETA
                EXTEND
                MP      BIT5            # FIRST TERM OF NEGUSUM IN K2THETA.
                ADS     K2THETA +1      # NO CARRY NEEDED       D.P. AT 4*PI(2)

                CS      ACENTRAL        # FORM ALPHA(2)/(2*K) AT 16*PI, IN D.P.,
                EXTEND                  # LIMITING QUOTIENT TO AVOID OVERFLOW.
                MP      BIT14           # -ALPHA/2 IN A, SCALED AT PI/4
                EXTEND
                MP      ACENTRAL        # -ALPHA(2)/2 IN A,L, SCALED AT PI(2)/16)
                AD      KCENTRAL
                EXTEND
                BZMF    HUGEQUOT        # K-ALPHA(2)/2 SHOULD BE PNZ FO DIVISION

                EXTEND
                DCS     A               # ALPHA(2)/2 - K
                AD      KCENTRAL
# RSB 2009 ----------------------------------------------------------------------
                EXTEND
                DV      KCENTRAL        # HIGH ORDER OF QUOTIENT.
                XCH     A2CNTRAL
                CA      L               # SHIFT UP THE REMAINDER.
                LXCH    7               # ZERO LOW-ORDER DIVIDEND.
```

EXTEND

```
                DV      KCENTRAL
                XCH     A2CNTRAL +1     # QUOTIENT STORED AT 16*PI , D.P.
                TCF     HAVEQUOT

HUGEQUOT        CA      POSMAX
                TS      L
                DXCH    A2CNTRAL        # LIMITED QUOTIENT STORED AT 16*PI, D.P.

HAVEQUOT        CA      WCENTRAL
                EXTEND
                MP      BIT9            # RESCALE OMEGA AT 16*PI IN D.P.
                DXCH    K2CNTRAL        # LOWER WORD OVERLAYS OMEGA IN WCENTRAL

                EXTEND
                DCA     K2CNTRAL
                DXCH    FUNCTION

                CA      ACENTRAL        # GET ALPHA*ABS(ALPHA)/(2*K)
                EXTEND
                BZMF    +4

                EXTEND
                DCA     A2CNTRAL
                TCF     +3

                EXTEND
                DCS     A2CNTRAL

                DAS     FUNCTION        # OMEGA + ALPHA*ABS(ALPHA)/(2*K) AT 16*PI

                CCS     FUNCTION        # DEL = +1 FOR FUNCT1 GREATER THAN ZERO.
                TCF     POSFNCT1        # OTHERWISE DEL = -1
                TCF     +2
                TCF     NEGFNCT1

                CCS     FUNCTION +1     # USE LOW ORDER WORD SINCE HIGH IS ZERO
POSFNCT1        CAF     BIT1
                TCF     +2
NEGFNCT1        CS      BIT1
                TS      DEL

                CCS     DEL             # REPLACE OMEGA BY DEL*OMEGA
                TCF     FUNCT2          # POSITIVE DEL VALUE.  PROCEED.
                TCF     DEFUNCT
                TCF     NEGFNCT2

DEFUNCT         TS      K2CNTRAL
                TS      K2CNTRAL +1
                TCF     FUNCT2
```

```
NEG1/3          DEC     -.33333

NEGFNCT2        EXTEND
                DCS     K2CNTRAL
                DXCH    K2CNTRAL

FUNCT2          EXTEND
                DCA     A2CNTRAL
                DAS     K2CNTRAL        # DEL*OMEGA + ALPHA(2)/(2*K) AT 16*PI,D.P.

FUNCT3          CA      A2CNTRAL
                EXTEND
                MP      NEG1/3
                DXCH    A2CNTRAL
                CA      L
                EXTEND
                MP      NEG1/3
                ADS     A2CNTRAL +1
                TS      L
                TCF     +2              # A2CNTRAL NOW CONTAINS  -ALPHA(2)/(6*K),
                ADS     A2CNTRAL        # SCALED AT 16*PI, IN D.P.

                EXTEND
                DCA     K2CNTRAL        # DEL*OMEGA + ALPHA(2)/(3*K) IN A2CNTRAL,
                DAS     A2CNTRAL        # SCALED AT 16*PI, D.P.

                CA      A2CNTRAL
                EXTEND
                MP      ACENTRAL
                DAS     K2THETA
                CA      A2CNTRAL +1
                EXTEND
                MP      ACENTRAL        # ACENTRAL MAY NOW BE OVERLAID.
                ADS     K2THETA +1
                TS      L
                TCF     +2              # TWO TERMS OF NEGUSUM ACCUMULATED, SO FAR
                ADS     K2THETA         # SCALED AT 4*PI(2), IN D.P.

GETROOT         CA      K2CNTRAL        # K*(DEL*OMEGA + ALPHA(2)/(2*K)) IS THE
                EXTEND                  # TERM FOR WHICH A SQUARE ROOT IS NEEDED.
                MP      KCENTRAL        # K AT PI/2(8)
                DXCH    FUNCTION
                CA      K2CNTRAL +1
                EXTEND
                MP      KCENTRAL
                ADS     FUNCTION +1
                TS      L
                TCF     +2
                ADS     FUNCTION        # DESIRED TERM IN FUNCTION, AT PI(2)/16
```

```
                CCS     DEL
                TCF     RSTOFGTS
                TCF     NEGUSUM
                TCF     NEGATE
                TCF     NEGUSUM

NEGATE          EXTEND
                DCS     K2CNTRAL
                DXCH    K2CNTRAL
                TCF     RSTOFGTS

                BANK    16
                EBANK=  NEGUQ
                SETLOC  DAPS1
                BANK

# THE WRCHN12 SUBROUTINE SETS BITS 9,10,11,12 OF CHANNEL 12 ON THE BASIS OF THE CONTENTS OF NEGUQ,NEGUR WHICH ARE
# THE NEGATIVES OF THE DESIRED ACCELERATION CHANGES.  ACDT+C12 SETS Q(R)ACCDOT TO REFLECT THE NEW DRIVES.
#
# WARNING:  ACDT+C12 AND WRCHN12 MUST BE CALLED WITH INTERRUPT INHIBITED.

BGIM            OCTAL   07400
CHNL12          EQUALS  ITEMP6
ACDT+C12        CS      NEGUQ
                EXTEND                  # GIMBAL DRIVE REQUESTS.
                MP      ACCDOTQ
                LXCH    QACCDOT
                CS      NEGUR
                EXTEND
                MP      ACCDOTR
                LXCH    RACCDOT

                CCS     NEGUQ
                CAF     BIT10
                TCF     +2
                CAF     BIT9
                TS      CHNL12

                CCS     NEGUR
                CAF     BIT12
                TCF     +2
                CAF     BIT11
                ADS     CHNL12          # (STORED RESULT NOT USED AT PRESENT)

                CS      BGIM
                EXTEND
                RAND    CHAN12
                AD      CHNL12
                EXTEND
                WRITE   CHAN12
```

```
		CS	CALLGMBL	# TURN OFF REQUEST FOR ACDT+C12 EXECUTION.
		MASK	RCSFLAGS
		TS	RCSFLAGS

		TC	Q		# RETURN TO CALLER.

		BANK	21
		EBANK=	QDIFF
		SETLOC	DAPS4
		BANK
```

```
# SUBROUTINE TIMEGMBL:  MOD 0, OCTOBER 1967, CRAIG WORK
#
# TIMEGMBL COMPUTES THE DRIVE TIME NEEDED FOR THE TRIM GIMBAL TO POSITION THE DESCENT ENGINE NOZZLE SO AS TO NULL
# THE OFFSET ANGULAR ACCELERATION ABOUT THE Q (OR R) AXIS.  INSTEAD OF USING AOSQ(R), TIMEGMBL USES .4*AOSQ(R),
# SCALED AT PI/8.                          FOR EACH AXIS, THE DRIVE TIME IS COMPUTED AS ABS(ALPHA/ACCDOT).  A ZERO
# ALPHA OR ACCDOT OR A ZERO QUOTIENT TURNS OFF THE GIMBAL DRIVE IMMEDIATELY.  OTHERWISE, THE GIMBAL IS TURNED ON
# DRIVING IN THE CORRECT DIRECTION. THE Q(R)GIMTIMR IS SET TO TERMINATE THE DRIVE AND Q(R)ACCDOT
# IS STORED TO REFLECT THE NEW ACCELERATION DERIVATIVE.  NEGUQ(R) WILL CONTAIN +1,+0,-1 FOR A Q(R)ACCDOT VALUE
# WHICH IS NEGATIVE, ZERO, OR POSITIVE.
#
# INPUTS:  AOSQ,AOSR, SCALED AT P1/2, AND ACCDOTQ, ACCDOTR AT PI/2(7).    PI/2(7).
#
# OUTPUTS:   NEW GIMBAL DRIVE BITS IN CHANNEL 12,NEGUQ,NEGUR,QACCDOT AND RACCDOT, THE LAST SCALED AT PI/2(7).
#            Q(R)GIMTIMR WILL BE SET TO TIME AND TERMINATE GIMBAL DRIVE(S)
#
# DEBRIS:  A,L,Q, ITEMPS 2,3,6, RUPTREG2 AND ACDT+C12 DEBRIS.
#
# EXITS:  VIA TC Q.
#
# ALARMS, ABORTS, :  NONE
#
# SUBROUTINES:  ACDT+C12, IBNKCALL
#
# WARNING:  THIS SUBROUTINE WRITES INTO CHANNEL 12 AND USES THE ITEMPS.  THEREFORE IT MAY ONLY BE CALLED WITH
# INTERRUPT INHIBITED.
#
# ERASABLE STORAGE CONFIGURATION (NEEDED BY THE INDEXING METHODS):
# NEGUQ          ERASE   +2                     # NEGATIVE OF Q-AXIS GIMBAL DRIVE
# (SPWORD)       EQUALS  NEGUQ +1               # ANY S.P. ERASABLE NUMBER, NOW THRSTCMD
# NEGUR          EQUALS  NEGUQ +2               # NEGATIVE OF R-AXIS GIMBAL DRIVE
# ACCDOTQ        ERASE   +2                     # Q-JERK TERM SCALED AT PI/2(7) RAD/SEC(3)
# (SPWORD)       EQUALS  ACCDOTQ +1             # ANY S.P. ERASABLE NUMBER NOW QACCDOT
# ACCDOTR        EQUALS  ACCDOTQ +2             # R-JERK TERM SCALED AT PI/2(7) RAD/SEC(3)
#                                               # ACCDOTQ,ACCDOTR ARE MAGNITUDES.
# AOSQ           ERASE   +4                     # Q-AXIS ACC., D.P. AT PI/2 R/SEC(2)
# AOSR           EQUALS  AOSQ +2                # R-AXIS ACCELERATION SCALED AT PI/2 R/S2

QRNDXER          EQUALS  ITEMP6
OCT23146         OCTAL   23146                  # DECIMAL .6
NZACCDOT         EQUALS  ITEMP3

TIMEGMBL         CAF     ONE                    # INITIALZE ALLOWGTS.
                 TS      ALLOWGTS

                 CAF     TWO                    # SET UP LOOP FOR R AXIS.
                 LXCH    Q                      # SAVE RETURN ADDRESS.
                 LXCH    RUPTREG2
```

```
                TCF     +2
TIMQGMBL        CAF     ZERO                    # NOW DO THE Q-AXIS
                TS      QRNDXER
                INDEX   QRNDXER
                CA      ACCDOTQ                 # ACCDOT IS PRESUMED TO BE AT PI/2(7).
                EXTEND
                BZMF    TGOFFNOW                # IS ACCDOT LESS THAN OR EQUAL TO 0?
                TS      NZACCDOT                # NO.  STORE NON-ZERO, POSITIVE ACCDOT.

ALPHATRY        INDEX   QRNDXER
                CS      AOSQ
                EXTEND
                BZF     TGOFFNOW                # IS ALPHA ZERO?

                TS      Q                       # SAVE A COPY OF  -AOS.
                EXTEND                          # NO.  RESCALE FOR TIMEGMBL USE.
                MP      OCT23146                # OCTAL 23146 IS DECIMAL .6
                AD      Q                       # -1.6*AOS AT PI/2 = -.4*AOS AT PI/8.
                TS      L                       # WAS THERE OVERFLOW?
                TCF     SETNEGU                 # NO.  COMPUTE DRIVE TIME.

                CS      A                       # RECOVER  -SGN(AOS) IN THE A REGISTER.
                INDEX   QRNDXER                 # YES.  START DRIVE WITHOUT WAITLIST.
                XCH     NEGUQ
                TCF     NOTALLOW                # KNOCK DOWN THE ALLOWGTS FLAG.

SETNEGU         EXTEND
                BZMF    POSALPH

                COM
                TS      ITEMP2                  # STORE  -ABS(.4*AOS) SCALED AT PI/8.
                CS      BIT1
                TCF     POSALPH +2
POSALPH         TS      ITEMP2                  # STORE  -ABS(.4*AOS) SCALED AT PI/8.
                CA      BIT1
         +2     INDEX   QRNDXER                 # SGN(AOS) INTO NEGU
                TS      NEGUQ                   # STORE SGN(ALPHA) AS NEGU

                CA      NZACCDOT
                EXTEND
                MP      BIT12                   # 2*ACCDOT, SCALED AT PI/8.
                AD      ITEMP2                  # -ABS(ALPHS) + 2*ACCDOT, AT PI/8.
                EXTEND
                BZMF    NOTALLOW                # IS DRIVE TIME MORE THAN TWO SECONDS?
                CS      ITEMP2                  # NO.  COMPUTE DRIVE TIME.
                EXTEND                          # ABS(ALPHA) AT PI/8.
                MP      OCT00240                # DECIMAL 10/1024
                EXTEND                          # QUOTIENT IS DRIVE TIME AT WAITLIST.
                DV      NZACCDOT                # ABS(ALPHA)/ACCDOT AT 2(14)/100
```

```
                EXTEND
                BZF     TGOFFNOW                # DRIVE TIME MUST BE GREATER THAN ZERO.

                TCF     DRIVEON

TGOFFNOW        CAF     ZERO                    # TURN OFF GIMBAL NOW.
                INDEX   QRNDXER
                TS      NEGUQ

                TCF     DONEYET

NOTALLOW        CAF     OCT31
                INDEX   QRNDXER
                TS      QGIMTIMR
                CAF     ZERO                    # DRIVE TIME IS MORE THAN 2 SECONDS, SO
                TS      ALLOWGTS                # DO NOT PERMIT FURTHER GTS ATTITUDE-RATE
                                                # CONTROL UNTIL AOSTASK APPROVES.
                TCF     DONEYET                 # NO WAITLIST CALL IS MADE.

DRIVEON         INDEX   QRNDXER
                TS      QGIMTIMR                # CHOOSE Q OR R AXIS.

DONEYET         CCS     QRNDXER
                TCF     TIMQGMBL

                DXCH    RUPTREG3                # PROTECT IBNKCALL ERASABLES.  ACDT+C12
                DXCH    ITEMP2                  # LEAVES ITEMPS2,3 ALONE.

                TC      IBNKCALL                # TURN OFF CHANNEL BITS, SET Q(R)ACCDOTS.
                CADR    ACDT+C12

                DXCH    ITEMP2                  # RESTORE ERASABLES FOR IBNKCALL.
                DXCH    RUPTREG3

                TC      RUPTREG2                # RETURN TO CALLER.

OCT00240        OCTAL   00240                   # DECIMAL 10/1024
```

```
# THE FOLLOWING SECTION IS A CONTINUATION OF THE TRIM GIMBAL CONTROL FROM THE LAST GTS ENTRY. THE QUANTITY NEGUSUM
# IS COMPUTED FOR EACH AXIS (Q,R), .707*DEL*FUNCTION(3/2) + K2THETA = NEGUSUM.  NEW DRIVES ARE ENTERED TO CH 12.
#
# THE SUBROUTINE GTSQRT ACCEPTS A DOUBLE PRECISION VALUE IN FUNCTION, FUNCTION +1 AND RETURNS A SINGLE-PRECISION
# SQUARE ROOT OF THE FOURTEEN MOST SIGNIFICANT BITS OF THE ARGUMENT.  ALSO, THE CELL SHFTFLAG CONTAINS A BINARY
# EXPONENT S, SUCH THAT THE SQUARE ROOT (RETURNED IN THE A REGISTER) MUST BE SHIFTED RIGHT (MULTIPLIED BY 2 TO THE
# POWER (-S)) IN ORDER TO BE THE TRUE SQUARE ROOT OF THE FOURTEEN MOST SIGNIFICANT BITS OF FUNCTION, FUNCTION +1.
# SQUARE ROOT ERROR IS NOT MORE THAN 2 IN THE 14TH SIGNIFICANT BIT.  CELLS CLOBBERED ARE A,L,SHFTFLAG,ININDEX,
# HALFARG,SCRATCH,SR,FUNCTION, FUNCTION +1.  GTSQRT IS CALLED BY TC GTSQRT AND RETURNS VIA TC Q OR TC FUNCTION +1.
# ZERO OR NEGATIVE ARGUMENTS YIELD ZERO FOR SQUARE ROOTS.

GTSQRT          CCS     FUNCTION
                TCF     GOODARG         # FUNCTION IS POSITIVE.  TAKE SQUARE ROOT.
                TCF     +2              # HIGH ORDER WORD IS ZERO.  TRY THE LOWER.
                TCF     ZEROOT          # NEGATIVE.  USE ZERO FOR 1/2 POWER.

                CA      FUNCTION +1
                EXTEND
                BZMF    ZEROOT

                TCF     ZEROHIGH        # PROCEED.
ZEROOT          CA      ZERO
                TS      SHFTFLAG
                TC      Q

ZEROHIGH        XCH     FUNCTION        # 14 MOST SIGNIFICANT BITS ARE IN THE
                XCH     FUNCTION +1     # LOWER WORD.  EXCHANGE THEM.
                CA      SEVEN
                TCF     GOODARG +1

GOODARG         CA      ZERO
                TS      SHFTFLAG
                CA      TWELVE          # INITIALIZE THE SCALING LOOP.
                TS      ININDEX
                TCF     SCALLOOP

SCALSTRT        CA      FUNCTION
                TCF     SCALDONE

MULBUSH         CA      NEG2            # IF ARG IS NOT LESS THAN 1/4, INDEX IS
                ADS     ININDEX         # ZERO, INDICATING NO SHIFT NEEDED.
                EXTEND                  # BRANCH IF ARG IS NOT LESS THAN 1/4.
                BZMF    SCALSTRT        # OTHERWISE COMPARE ARG WITH A REFERENCE
                                        # WHICH IS 4 TIMES LARGER THAN THE LAST.
SCALLOOP        CS      FUNCTION
                INDEX   ININDEX
                AD      BIT15           # REFERENCE MAGNITUDE LESS OR EQUAL TO 1/4
                EXTEND
                BZMF    MULBUSH         # IF ARG IS NOT LESS THAN REFERENCE, GO
                                        # AROUND THE MULBERRY BUSH ONCE MORE.
```

```
                INDEX   ININDEX
                CA      BIT15           # THIS IS THE SCALE MAGNITUDE
                XCH     HALFARG         # 2**(-ININDEX) IS THE SHIFT DIVISOR.
                EXTEND                  # RESCALE ARGUMENT.
                DCA     FUNCTION
                EXTEND
                DV      HALFARG
                                        # ININDEX AND SHFTFLAG PRESERVE INFO FOR
                                        # RESCALING AFTER ROOT PROCESS.
SCALDONE        EXTEND
                QXCH    FUNCTION +1     # SAVE Q FOR RETURN
                EXTEND
                MP      BIT14
                TS      HALFARG
                MASK    BIT13
                CCS     A
                CA      OCT11276
                AD      ROOTHALF        # INITIAL GUESS IS ROOT 1/2 OR POSMAX
                TC      ROOTCYCL
                TC      ROOTCYCL
                TC      ROOTCYCL
                TC      FUNCTION +1
# ***************************************************************************************************************

RSTOFGTS        TC      GTSQRT
PRODUCT         XCH     K2CNTRAL
                EXTEND
                MP      K2CNTRAL
                DXCH    K2CNTRAL
                EXTEND                  #              THE PRODUCT OF
                MP      L               #  1/2                   2       1/2
                ADS     K2CNTRAL +1     # K   *(DEL*OMEGA + ALPHA /(2*K))
                TS      L               #                   AND
                TCF     +2              #                        2
                ADS     K2CNTRAL        #  DEL*(DEL*OMEGA + ALPHA /(2*K)) NOW IN
                                        # K2CNTRAL

DOSHIFT         CA      ININDEX
                EXTEND                  # MULTIPLY IN THE FACTOR 2(-S), RETURNED
                MP      BIT14           # BY THE GTSQRT SUBROUTINE
                ADS     SHFTFLAG
                EXTEND
                BZF     ADDITIN
                INDEX   SHFTFLAG
                CA      BIT15
```

```
                XCH     K2CNTRAL
                EXTEND
                MP      K2CNTRAL
                DAS     K2THETA
                XCH     K2CNTRAL
                EXTEND
                MP      K2CNTRAL +1
                ADS     K2THETA  +1
                TS      L
                TCF      +2
                ADS     K2THETA

                TCF     NEGUSUM

ADDITIN         EXTEND
                DCA     K2CNTRAL
                DAS     K2THETA         # NOW ADD IN THE K2THETA TERM.
NEGUSUM         CCS     K2THETA         # TEST SIGN OF HIGH ORDER PART.
                TCF     NEGDRIVE
                TCF      +2
                TCF     POSDRIVE

                CCS     K2THETA  +1     # SIGN TEST FOR LOW ORDER PART.
NEGDRIVE        CA      BIT1
                TCF     +2              # STOP GIMBAL DRIVE FOR A ZERO NEGUSUM.
POSDRIVE        CS      BIT1
                TS      L               # SAVE FOR DRIVE REVERSAL TEST.
                INDEX   QRCNTR
                XCH     NEGUQ

                EXTEND
                MP      L               # MULTIPLY OLD NEGU AND NEW NEGU.
                CCS     L
                TCF     LOUPE           # NON-ZERO GIMBAL DRIVE BEING CONTINUED.

                TCF     ZEROLOUP        # NO REVERSAL PROBLEM HERE.

                TCF     REVERSAL        # NON-ZERO GIMBAL DRIVE BEING REVERSED.
                TCF     ZEROLOUP        # NO REVERSAL PROBLEM HERE.

REVERSAL        INDEX   QRCNTR          # A ZERO-DRIVE PAUSE IS NEEDED HERE.  ZERO
                TS      QACCDOT         # IS IN A REGISTER FROM CCS ON (-1).
                INDEX   QRCNTR
                CS      GMBLBITA
                EXTEND
                WAND    CHAN12

ZEROLOUP        CS      RCSFLAGS        # SET UP REQUEST FOR ACDT+C12 CALL.
                MASK    CALLGMBL
                ADS     RCSFLAGS
```

```
LOUPE           CCS     QRCNTR          # HAVE BOTH AXES BEEN PROCESSED?
                TCF     GOQTRIMG        # NO.  DO Q AXIS NEXT.

                CA      SAVESR          # RESTORE THE SR
                TS      SR

GOCLOSE         EXTEND                  # TERMINATE THE JASK.
                DCA     CLOSEADR
                DTCB

                EBANK=  AOSQ
CLOSEADR        2CADR   CLOSEOUT        # TERMINATE THE JASK.

TWELVE          EQUALS  OCT14
ROOTHALF        OCTAL   26501           # SQUARE ROOT OF 1/2
GMBLBITA        OCTAL   01400           # INDEXED WRT GMBLBITB   DO NOT MOVE******
OCT11276        OCTAL   11276           # POSMAX - ROOTHALF
GMBLBITB        OCTAL   06000           # INDEXED WRT GMBLBITA   DO NOT MOVE******

# SUBROUTINE ROOTCYCL:  BY CRAIG WORK,3 APRIL 68
# ROOTCYCL IS A SUBROUTINE WHICH EXECUTES ONE NEWTON SQUARE ROOT ALGORITHM ITERATION.  THE INITIAL GUESS AT THE
# SQUARE ROOT IS PRESUMED TO BE IN THE A REGISTER AND ONE-HALF THE SQUARE IS TAKEN FROM HALFARG.  THE NEW APPROXI-
# MATION TO THE SQUARE ROOT IS RETURNED IN THE A REGISTER.  DEBRIS:   A,L,SR,SCRATCH.  ROOTCYCL IS CALLED FROM
# LOCATION (LOC) BY A TC ROOTCYCL, AND RETURNS (TC Q) TO LOC +1.
# WARNING:  IF THE INITIAL GUESS IS NOT GREATER THAN THE SQUARE, DIVIDE OR ADD OVERFLOW IS A REAL POSSIBILITY.

ROOTCYCL        TS      SCRATCH         # STORE X
                TS      SR              # X/2 NOW IN SR
                CA      HALFARG         # ARG/2 IN THE A REG
                ZL                      # PREPARE FOR DIVISION
                EXTEND
                DV      SCRATCH         # (ARG/X)/2
                AD      SR              # (X + ARG/X)/2 IN THE A REG
                TC      Q
```

```
# PROGRAM NAME:          1/ACCS
# PROGRAM WRITTEN BY:    BOB COVELLI AND MIKE HOUSTON
# LAST MODIFICATION:     FEB. 14, 1969 BY G. KALAN
#
# PROGRAM DESCRIPTION:
#       1/ACCS PROVIDES THE INTERFACE BETWEEN THE GUIDANCE PROGRAMS AND THE DIGITAL AUTOPILOT.  WHENEVER THERE IS A
#       CHANGE IN THE MASS OF THE VEHICLE, IN THE DEADBAND SELECTED, IN THE VEHICLE CONFIGURATION (ASCENT-DESCENT-
#       DOCKED), AND DURING A FRESH START OR A RESTART, 1/ACCS IS CALLED TO COMMUNICATE THE DATA CHANGES TO THE DAP.
#
#       THE INPUTS TO 1/ACCS ARE MASS, ACCELERATION (ABDELV), DEADBAND (DB), OFFSET ACCELERATIONS (AOSQ AND AOSR),
#       STAGE VERIFY BIT (CHAN30, BIT2), DOCKED BIT (DAPBOOLS, BIT13), DRIFT BIT (DAPBOOLS, BIT8), USEQRJTS (DAPBOOLS,
#       BIT14), AND SURFACE FLAG (FLAGWRDB, BIT8), AND CH5MASK.
#
#       1/ACCS COMPUTES THE JET ACCELERATIONS (1JACC, 1JACCQ, 1JACCR) AS FUNCTIONS OF MASS.  1JACCU AND 1JACCV ARE
#       FORMED BY RESOLVING 1JACCQ AND 1JACCR.  IN THE DESCENT CASE, THE DESCENT ENGINE MOMENT ARM (L, PVT-CG) IS ALSO
#       COMPUTED AS A FUNCTION OF MASS.  THE RATE OF CHANGE OF ACCELERATION DUE TO ROTATION OF THE GIMBAL (ACCDOTQ,
#       ACCDOTR) IS ALSO COMPUTED IN THE DESCENT CASE.
#
#       AFTER THE ABOVE COMPUTATIONS, THE PROGRAM 1/ACCONT COMPUTES THE RECIPROCAL NET ACCELERATIONS ABOUT THE P, U,
#       AND V AXES (2 JETS FOR P-AXIS, BOTH 1 AND 2 JETS FOR U AND V AXES), AND THE RECIPROCAL COAST ACCELERATIONS ABOUT
#       THE P, U, AND V AXES.  THE ACCELERATION FUNCTIONS (ACCFCTZ1 AND ACCFCTZ5) ARE ALSO COMPUTED FOR THESE AXES.  THE
#       FIRE AND COAST DEADBANDS AND AXISDIST ARE COMPUTED FOR EACH AXIS.  FLAT AND ZONE3LIM, THE WIDTH AND HEIGHT OF THE
#       MINIMUM IMPULSE ZONE, ARE COMPUTED.  1/ACCONT ALSO SETS ACCSWU AND ACCSWV, WHICH INDICATE WHEN 1 JET ACCELERATION
#       IS NOT SUFFICIENT TO PRODUCE MINIMUM ACCELERATION.  AT THE COMPLETION OF 1/ACCS, THE ACCSOKAY BIT IS SET.
#
# SUBROUTINES CALLED:
#       TIMEGMBL
#       MAKECADR
#       ROT45DEG
#
# CALLING SEQUENCE:
#               TC      BANKCALL        # (1/ACCS MUST BE CALLED BY BANKCALL)
#               CADR    1/ACCS
#
# NORMAL EXIT:  VIA BANKJUMP
#
# ALARM AND EXIT MODES:  NONE
#
# INPUT/OUTPUT:  SEE PROGRAM DESCRIPTION.
#
# DEBRIS:
#       ALL OF THE EXECUTIVE TEMPORARY REGISTERS, EXCEPT FIXLOC AND OVFIND, AND THE CORE SET AREA FROM MPAC TO BANKSET.
#
# RESTRICTIONS:
#       1/ACCS MUST BE CALLED BY BANKCALL
#       EBANK IS SET TO 6, BUT NOT RESTORED.
```

```
                BANK    20
                SETLOC  DAPS3
                BANK

                COUNT*  $$/DAPAO

                EBANK=  AOSQ

# ENTRY IS THROUGH 1/ACCJOB OR 1/ACCSET WHEN 1/ACCS IS TO BE DONE AS A SEPARATE NOVAC JOB.
# IT IS POSSIBLE FOR MORE THAN ONE OF THESE JOBS TO BE SET UP CONCURRENTLY.  HOWEVER, SINCE THERE IS NO CHECK OF
# NEWJOB, A SECOND MANIFESTATION CANNOT BE STARTED UNTIL THE FIRST IS COMPLETED.

1/ACCSET        CAF     ZERO            # ENTRY FROM FRESH START/RESTART CODING.
                TS      AOSQ            #       NULL THE OFFSET ESTIMATES FOR 1/ACCS.
                TS      AOSR
                TS      ALPHAQ          #       NULL THE OFFSET ESTIMATES FOR DOWNLIST
                TS      ALPHAR

1/ACCJOB        TC      BANKCALL        # 1/ACCS ASSUMES ENTRY VIA BANKCALL.
                CADR    1/ACCS  +2      # SKIP EBANK SETTING.

                TC      ENDOFJOB

1/ACCS          CA      EBANK6          # ***** EBANK SET BUT NOT RESTORED *****
                TS      EBANK

                TC      MAKECADR        # SAVE RETURN SO THAT BUF2 MAY BE USED
                TS      ACCRETRN

# DETERMINE MASS OF THE LEM.

                CA      DAPBOOLS        # IS THE CSM DOCKED
                MASK    CSMDOCKD
                TS      DOCKTEMP        # STORE RECORD OF STATE IN TEMP (MPAC +3).
                CCS     A
                CS      CSMMASS         #       DOCKED:  LEMMAS = MASS - CSMMASS
                AD      MASS            #       LEM ALONE:  LEMMASS = MASS
                TS      LEMMASS

# ON THE BASIS OF APSFLAG:
#       SET THE P-AXIS RATE COMMAND LIMIT FOR 2-JET/2-JET CONTROL
#       SET MPAC, WHICH INDICATES THE PROPER SET OF COEFFICIENTS FOR THE LEM-ALONE F(MASS) CALCULATIONS
#       ENSURE THAT THE LEM MASS VALUE IS WITHIN THE ACCEPTABLE RANGE

                INHINT
                CAE     FLGWRD10        # DETERMINE WHETHER STAGED.
                MASK    APSFLBIT
                EXTEND
                BZF     DPSFLITE
```

```
                CS      POSMAX          # ASCENT (OR ON LUNAR SURFACE)
                TS      -2JETLIM        # ALWAYS 2 JETS FOR P-AXIS RATE COMMAND
                CAF     OCT14           # INITIALIZE INDEX AT 12.
                TS      MPAC
                CS      LEMMASS         # CHECK IF MASS TOO HIGH.  CATCH STAGING.
                AD      HIASCENT
                EXTEND
                BZMF    MASSFIX
                CS      LEMMASS         # CHECK IF MASS TOO LOW.  THIS LIMITS THE
                AD      LOASCENT        #       DECREMENTING BY MASSMON.
                EXTEND
                BZMF    F(MASS)

MASSFIX         ADS     LEMMASS         # STORE THE VIOLATED LIMIT AS LEMMASS.
                ZL                      #       ALSO CORRECT TOTAL MASS, ZEROING THE
                CCS     DOCKTEMP        #       LOW-ORDER WORD.
                CAE     CSMMASS         #               DOCKED:  MASS = LEMMASS + CSMMASS
                AD      LEMMASS         #               LEM ALONE:  MASS = LEMMASS
                DXCH    MASS
                TCF     F(MASS)

DPSFLITE        CS      BIT10           # FOUR JETS FOR P-AXIS RATE COMMAND ERRORS
                TS      -2JETLIM        #       EXCEEDING 1.4 DEG/SEC (SCALED AT 45)
                CAF     SIX             # INITIALIZE INDEX AT 6.
                TS      MPAC
                CS      LEMMASS         # CHECK IF MASS TOO HIGH.  SHOULD NEVER
                AD      HIDESCNT        #       OCCUR EXCEPT PERHAPS BEFORE THE PAD
                EXTEND                  #       LOAD IS DONE.
                BZMF    MASSFIX
                CS      LEMMASS         # CHECK IF MASS TOO LOW.  THIS LIMITS THE
                AD      LODESCNT        #       DECREMENTING BY MASSMON.
                AD      HIASCENT
                EXTEND
                BZMF    F(MASS)
                TCF     MASSFIX

# COMPUTATION OF FUNCTIONS OF MASS

F(MASS)         RELINT
                CCS     DOCKTEMP
                TCF     DOCKED          # DOCKED:  USE SEPARATE COMPUTATION.
                CA      TWO
STCTR           TS      MPAC    +1      # J=2,1,0 FOR 1JACCR,1JACCQ,1JACC

                CS      TWO
                ADS     MPAC            # JX=10,8,6 OR 4,2,0 TO INDEX COEFS.

STCTR1          CAE     LEMMASS
                INDEX   MPAC
                AD      INERCONC
                TS      MPAC    +2      # MASS + C
```

```
                EXTEND
                INDEX   MPAC
                DCA     INERCONA
                EXTEND
                DV      MPAC    +2
                INDEX   MPAC
                AD      INERCONB
                INDEX   MPAC    +1              # 1JACC(J)=A(JX)/(MASS+C(JX) + B(JX)
                TS      1JACC                   # 1JACC(-1)=L,PVT-CG  SCALED AT 8 FEET

                CCS     MPAC    +1
                TCF     STCTR
                TCF     COMMEQS
                TCF     LRESC

# COEFFQ AND COEFFR ARE COMPUTED IN THIS SECTION.  THEY ARE USED TO RESOLVE Q-R COMPONENTS INTO NON-ORTHOGONAL
# U AND V COMPONENTS (SEE ROT-TOUV SECTION).

COMMEQS         CS      1JACCR
                AD      1JACCQ
                EXTEND
                BZMF    BIGIQ
                EXTEND                          # EPSILON IS A MEASURE OF COUPLING AND IS
                DV      1JACCQ                  # DEFINED=1-IQ/IR FOR IR GREATER THAN IQ.
                TS      EPSILON                 # THE COMPUTED EXPRESSION IS EQUIVALENT
                AD      -EPSMAX
                EXTEND
                BZMF    GOODEPS1
                CS      -EPSMAX
                TS      EPSILON                 # EPSILON IS LIMITED TO A MAX. OF .42265
GOODEPS1        CA      EPSILON
                EXTEND
                MP      0.35356
                AD      .7071
                TS      COEFFR                  # IN THIS CASE WHERE IR IS GREATER THAN
                CS      POSMAX                  # IQ, COEFFQ=-.707(1+.5EPSILON)(1-EPSILON)
                AD      EPSILON                 # AND COEFFR=.707(1+.5EPSILON)
                EXTEND
                MP      COEFFR
                TS      COEFFQ
                TCF     JACCUV
BIGIQ           EXTEND                          # EPSILON IS DEFINED AS 1-IR/IQ FOR IQ
                DV      1JACCR                  # GREATER THAN IR.  -EPSILON IS COMPUTED
                TS      -EPSILON                # RATHER THAN EPSILON FOR CONVENIENCE
                CS      -EPSILON
                AD      -EPSMAX
                EXTEND
                BZMF    GOODEPS2
                CA      -EPSMAX
                TS      -EPSILON                # EPSILON IS LIMITED TO A MAX. OF .42265
```

```
GOODEPS2        CA      -EPSILON
                EXTEND
                MP      0.35356
                AD      -.7071
                TS      COEFFQ          # IN THIS CASE WHERE IQ IS GREATER THAN
                CS      -EPSILON        # IR, COEFFQ=-.707(1+.5EPSILON) AND
                AD      NEGMAX          # COEFFR=.707(1+.5EPSILON)(1-EPSILON)
                EXTEND
                MP      COEFFQ
                TS      COEFFR
JACCUV          CS      COEFFQ
                EXTEND
                MP      1JACCQ          # 1JACCQ IS SCALED AT PI/4
                TS      1JACCU          # 1JACCU USED AS TEMPORARY STORAGE
                CA      COEFFR
                EXTEND
                MP      1JACCR
                AD      1JACCU
                EXTEND
                MP      BIT14           # SCALING CHANGED FROM PI/4 TO PI/2
                TS      1JACCU
                TS      1JACCV          # SCALED AT PI/2 RADIANS/SEC(2)
                CCS     MPAC            # COMPUTE L,PVT-CG IF IN DESCENT
                CAF     ZERO            # ZERO SWITCHES AND GO TO 1/ACCONT IN
                TS      ALLOWGTS        #       ASCENT
                TCF     1/ACCONT -1

                CS      TWO
                TS      MPAC
                CS      ONE
                TS      MPAC    +1
                TCF     STCTR1

# THIS SECTION COMPUTES THE RATE OF CHANGE OF ACCELERATION DUE TO THE ROTATION OF THE GIMBALS.  THE EQUATION
# IMPLEMENTED IN BOTH THE Y-X PLANE AND THE Z-X PLANE IS -- D(ALPHA)/DT = TL/I*D(DELTA)/DT, WHERE
#       T = ENGINE THRUST FORCE
#       L = PIVOT TO CG DISTANCE OF ENGINE
#       I = MOMENT OF INERTIA

LRESC           CAE     ABDELV          # SCALED AT 2(13) CM/SEC(2)
                EXTEND
                MP      MASS            # SCALED AT B+16 KGS
                TC      DVOVSUB         # GET QUOTIENT WITH OVERFLOW PROTECTION
                ADRES   GFACTM

# MASS IS DIVIDED BY ACCELERATION OF GRAVITY IN ORDER TO MATCH THE UNITS OF IXX,IYY,IZZ, WHICH ARE SLUG-FT(2).
# THE RATIO OF ACCELERATION FROM PIPAS TO ACCELERATION OF GRAVITY IS THE SAME IN METRIC OR ENGINEERING UNITS, SO
# THAT IS UNCONVERTED.  2.20462 CONVERTS KG. TO LB.  NOW T IN IN A SCALED AT 2(14).

                EXTEND
                MP      L,PVT-CG        # SCALED AT 8 FEET.
```

```
                INHINT
                TS      MPAC
                EXTEND
                MP      1JACCR
                TC      DVOVSUB         # GET QUOTIENT WITH OVERFLOW PROTECTION
                ADRES   TORKJET1

                TS      ACCDOTR         # SCALED AT PI/2(7)
                CA      MPAC
                EXTEND
                MP      1JACCQ
                TC      DVOVSUB         # GET QUOTIENT WITH OVERFLOW PROTECTION
                ADRES   TORKJET1

SPSCONT         TS      ACCDOTQ         # SCALED AT PI/2(7)
                EXTEND
                MP      DGBF            # .3ACCDOTQ SCALED AT PI/2(8)
                TS      KQ
                CAE     ACCDOTR         # .3ACCDOTR AT PI/2(8)
                EXTEND
                MP      DGBF
                TS      KRDAP
                EXTEND                  # NOW COMPUTE QACCDOT, RACCDOT, THE SIGNED
                READ    CHAN12          # JERK TERMS.  STORE CHANNEL 12. WITH GIMBAL
                TS      MPAC    +1      # DRIVE BITS 9 THROUGH 12 SET LOOP
                CAF     BIT2            # INDEX TO COMPUTE RACCDOT, THEN QACCDOT.
                TCF     LOOP3
                CAF     ZERO            # ACCDOTQ AND ACCDOTR ARE NOT NEGATIVE,
LOOP3           TS      MPAC            # BECAUSE THEY ARE MAGNITUDES
                CA      MPAC    +1
                INDEX   MPAC            # MASK CHANNEL IMAGE FOR ANY GIMBAL MOTION
                MASK    GIMBLBTS
                EXTEND
                BZF     ZACCDOT         # IF NONE, Q(R)ACCDOT IS ZERO.
                CA      MPAC    +1
                INDEX   MPAC            # GIMBAL IS MOVING.  IS ROTATION POSITIVE.
                MASK    GIMBLBTS +1
                EXTEND
                BZF     FRSTZERO        # IF NOT POSITIVE, BRANCH
                INDEX   MPAC            # POSITIVE ROTATION, NEGATIVE Q(R)ACCDOT.
                CS      ACCDOTQ
                TCF     STACCDOT
FRSTZERO        INDEX   MPAC            # NEGATIVE ROTATION, POSITIVE Q(R)ACCDOT.
                CA      ACCDOTQ
                TCF     STACCDOT
ZACCDOT         CAF     ZERO
STACCDOT        INDEX   MPAC
                TS      QACCDOT         # STORE Q(R)ACCDOT.
                CCS     MPAC
                TCF     LOOP3   -1      # NOW DO QACCDOT.
```

```
                CS      DAPBOOLS        # IS GIMBAL USABLE?
                MASK    USEQRJTS
                EXTEND
                BZF     DOWNGTS         # NO. BE SURE THE GIMBAL SWITCHES ARE DOWN
                CS      T5ADR           # YES.  IS THE DAP RUNNINT?
                AD      PAXISADR
                EXTEND
                BZF     +2
                TCF     DOWNGTS         # NO. BE SURE THE GIMBAL SWITCHES ARE DOWN
                CCS     INGTS           # YES.  IS GTS IN CONTROL?
                TCF     DOCKTEST        # YES.  PROCEED WITH 1/ACCS.
                TC      IBNKCALL        # NO. NULL OFFSET AND FIND ALLOWGTS
                CADR    TIMEGMBL

DOCKTEST        CCS     DOCKTEMP        # BYPASS 1/ACCONT WHEN DOCKED.
                TCF     1/ACCRET
                TCF     1/ACCONT
```

```
# SUBROUTINE:   DVOVSUB
# AUTHOR:       C. WORK, MOD 0, 12 JUNE 68
# PURPOSE:      THIS SUBROUTINE PROVIDES A SINGLE-PRECISION MACHINE LANGUAGE DIVISION OPERATION WHICH RETURNS
#               (1) THE QUOTIENT, IF THE DIVISION WAS NORMAL.
#               (2) NEGMAX, IF THE QUOTIENT WAS IMPROPER AND NEGATIVE.
#               (3) POSMAX, IF THE QUOTIENT WAS IMPROPER AND POSITIVE OR IF THERE WAS A ZERO DIVISOR.
#               THE CALLING PROGRAM IS PRESUMED TO BE A JOB IN THE F BANK WHICH CONTAINS DVOVSUB.  E BANK MUST BE 6.
#               THE DIVISOR FOR THIS ROUTINE MAY BE IN EITHER FIXED OR ERASABLE STORAGE.  SIGN AGREEMENT IS
#               ASSUMED BETWEEN THE TWO HALVES OF THE DIVIDEND.  (THIS IS CERTAIN IF THE A AND L REGISTERS ARE THE
#               RESULT OF A MULTIPLICATION OPERATION.)
# CALL SEQUENCE:        L       TC      DVOVSUB
#                       L +1    ADRES   (DIVISOR)
#                       L +2    RETURN HERE, WITH RESULT IN A,L
# INPUT:        DIVIDEND IN A,L (SIGN AGREEMENT ASSUMED), DIVISOR IN LOCATION DESIGNATED BY "ADRES".
#               DIVISOR MAY BE IN THE DVOVSUB FBANK,FIXED-FIXED FBANK,EBANK 6, OR UNSWITCHED ERASABLE.
# OUTPUT:       QUOTIENT AND REMAINDER, OR POSMAX (NEGMAX), WHICHEVER IS APPROPRIATE.
# DEBRIS:       SCRATCHX,SCRATCHY,SCRATCHZ,A,L  (NOTE: SCRATCHX,Y,Z ARE EQUATED TO MPAC +4,+5, AND +6.)
# ABORTS OR ALARMS:  NONE
# EXITS:        TO THE CALL POINT +2.
# SUBROUTINES CALLED:  NONE.

DVOVSUB         TS      SCRATCHY        # SAVE UPPER HALF OF DIVIDEND
                TS      SCRATCHX
                INDEX   Q               # OBTAIN ADDRESS OF DIVISOR.
                CA      0
                INCR    Q               # STEP Q FOR PROPER RETURN SEQUENCE.
                INDEX   A
                CA      0               # PICK UP THE DIVISOR.
                EXTEND                  # RETURN POSMAX FOR A ZERO DIVISOR.
                BZF     MAXPLUS

                TS      SCRATCHZ        # STORE DIVISOR.

                CCS     A               # GET ABS(DIVISOR) IN THE A REGISTER.
                AD      BIT1
                TCF     ZEROPLUS
                AD      BIT1

ZEROPLUS        XCH     SCRATCHY        # STORE ABS(DIVISOR).  PICK UP TOP HALF OF
                EXTEND                  # DIVIDEND.
                BZMF    GOODNEG         # GET -ABS(DIVIDEND)
```

```
                CS      A

GOODNEG         AD      SCRATCHY        # ABS(DIVISOR) - ABS(DIVIDEND)
                EXTEND
                BZMF    MAKEMAX         # BRANCH IF DIVISION IS NOT PROPER.

                CA      SCRATCHX        # RE-ESTABLISH THE DIVIDEND
                EXTEND
                DV      SCRATCHZ        # QUOTIENT IN THE A, REMAINDER IN L.
                TC      Q               # RETURN TO CALLER.

MAKEMAX         CCS     SCRATCHX        # DETERMINE THE SIGN OF THE QUOTIENT.
                CCS     SCRATCHZ        # SCRATCHX AND SCRATCHZ ARE NON-ZERO.
                TCF     MAXPLUS
                CCS     SCRATCHZ
                CAF     NEGMAX          # +,- OR -,+
                TC      Q
MAXPLUS         CAF     POSMAX          # -,- OR +,+
                TC      Q

# COEFFICIENTS FOR THE JET ACCELERATION CURVE FITS
# THE CURVE FITS ARE OF THE FORM --
#
#       1JACC = A/(MASS + C) + B
#
# A IS SCALED AT PI/4 RAD/SEC**2 B+16KG, B IS SCALED AT PI/4 RAD/SEC**2, AND C IS SCALED AT B +16 KG.
#
# THE CURVE FIT FOR L,PVT-CG IS OF THE SAME FORM, EXCEPT THAT A IS SCALED AT 8 FT B+16 KG, B IS SCALED AT 8 FT,
# AND C IS SCALED AT B+16 KG.

                2DEC    +.0410511917    # L             A               DESCENT

INERCONA        2DEC    +.0059347674    # 1JACCP        A               DESCENT

                2DEC    +.0014979264    # 1JACCQ        A               DESCENT

                2DEC    +.0010451889    # 1JACCR        A               DESCENT

                2DEC    +.0065443852    # 1JACCP        A               ASCENT

                2DEC    +.0035784354    # 1JACCQ        A               ASCENT

                2DEC    +.0056946631    # 1JACCR        A               ASCENT

                DEC     +.155044        # L             B               DESCENT
                DEC     -.025233        # L             C               DESCENT
```

```
INERCONB        DEC     +.002989        # 1JACCP        B               DESCENT
INERCONC        DEC     +.008721        # 1JACCP        C               DESCENT
                DEC     +.018791        # 1JACCQ        B               DESCENT
                DEC     -.068163        # 1JACCQ        C               DESCENT
                DEC     +.021345        # 1JACCR        B               DESCENT
                DEC     -.066027        # 1JACCR        C               DESCENT

                DEC     +.000032        # 1JACCP        B               ASCENT
                DEC     -.006923        # 1JACCP        C               ASCENT
                DEC     +.162862        # 1JACCQ        B               ASCENT
                DEC     +.002588        # 1JACCQ        C               ASCENT
                DEC     +.009312        # 1JACCR        B               ASCENT
                DEC     -.023608        # 1JACCR        C               ASCENT

GIMBLBTS        OCTAL   01400
                OCTAL   01000
                OCTAL   06000
                OCTAL   04000
DGBF            DEC     0.6             # .3 SCALED AT 1/2
0.35356         DEC     0.35356         # .70711 SCALED AT 2
GFACTM          OCT     337             # 979.24/2.20462 AT B+15
.7071           DEC     .70711
-.7071          DEC     -.70711
-EPSMAX         DEC     -.42265

# CSM-DOCKED INERTIA COMPUTATIONS

DOCKED          CA      ONE             # COEFTR  = 1 FOR INERTIA COEFFICIENTS
SPSLOOP1        TS      COEFCTR         #         = 7 FOR CG COEFFICIENTS
                CA      ONE             # MASSCTR = 1 FOR CSM
                TS      MASSCTR         #         = 0 FOR LEM

                INDEX   COEFCTR
                CA      COEFF   -1      # COEFF -1 = C
                EXTEND
                MP      LEMMASS
                EXTEND
                MP      CSMMASS         # LET X = CSMMASS AND Y = LEMMASS

                INDEX   COEFCTR
                AD      COEFF           # COEFF = F
                TS      MPAC            # MPAC = C X Y + F
                TCF     +4

SPSLOOP2        TS      MASSCTR         # LOOP TWICE THROUGH HERE TO OBTAIN
                EXTEND                  # MPAC = MPAC + (A X +D)X + (B Y +E)Y
                DIM     COEFCTR         #               LOOP #1     LOOP #2
                INDEX   COEFCTR
                CA      COEFF   +2      # COEFF +2 = A OR B
                EXTEND
```

```
                INDEX   MASSCTR
                MP      LEMMASS
                INDEX   COEFCTR
                AD      COEFF   +4      # COEFF +4 = E OR D
                EXTEND
                INDEX   MASSCTR
                MP      LEMMASS
                ADS     MPAC

                CCS     MASSCTR
                TCF     SPSLOOP2
                CCS     COEFCTR         # IF COEFCTR IS POS, EXIT FROM LOOP WITH
                TCF     +7              # CG X DELDOT = MPAC X 4 PI RAD-CM/SEC
TORQCONS        2DEC    0.51443 B-14    # CORRESPONDS TO 500 LB-FT

                CA      MPAC
                TS      MPAC    +1      # INERTIA = (MPAC +1) X 2(38) KG-CM(2)
                CA      SEVEN
                TCF     SPSLOOP1

                CA      1JACCCON        # 1JACC=1JACCCON/MASS
                ZL
                TC      DVOVSUB
                ADRES   MASS
                TS      1JACC           # SCALED AT PI/4

                CA      POSMAX          # SET INVERSE JET ACCELERATIONS TO POSMAX,
                TS      1/ANETP         # WHICH CORRESPONDS TO ACCEL. OF 1.4 D/SS.
                TS      1/ANET2 +1
                TS      1/ANET2 +2
                TS      1/ANET2 +17D
                TS      1/ANET2 +18D
                EXTEND
                DCA     TORQCONS
                EXTEND
                DV      MPAC    +1
                INHINT
                TS      1JACCQ          # SCALED AT PI/4
                TS      1JACCR

                CA      -.7071
                TS      COEFFQ          # COEFFQ AND COEFFR ARE CHOSEN TO MAKE U-
                CA      .7071           # AND V-AXES ORTHOGONAL FOR DOCKED CASE
                TS      COEFFR
                CA      MASS            # SCALED AT 2(16) KG
                EXTEND
                MP      MPAC            # SCALED AT 4 PI RAD-CM/SEC
                EXTEND
                MP      ABDELV          # SCALED AT 2(13) CM/SEC(2)
                TC      DVOVSUB         # GET QUOTIENT WITH OVERFLOW PROTECTION
```

```
                ADRES   MPAC    +1

                TS      ACCDOTR
                TCF     SPSCONT         # CONTINUE K, KSQ CALCULATIONS

1JACCCON        OCT     00167           # SCALED AT PI/4X2(16) RAD/SEC(2)-KG

#                                                2     2
# COEFFICIENTS FOR CURVE FIT OF THE FORM Z = A X  +B Y  +C X Y +D X +E Y +F

COEFF           DEC     .19518          # C  COEFFICIENT OF INERTIA
                DEC     -.00529         # F              "
                DEC     -.17670         # B              "
                DEC     -.03709         # A              "
                DEC     .06974          # E              "
                DEC     .02569          # D              "

                DEC     .20096          # C  COEFFICIENT OF CG
                DEC     .13564          # F           "
                DEC     .75704          # B           "
                DEC     -.37142         # A           "
                DEC     -.63117         # E           "
                DEC     .41179          # D           "

# ASSIGNMENT OF TEMPORARIES FOR 1/ACCS (EXCLUDING 1/ACCONT)
# MPAC, MPAC +1, MPAC +2  USED EXPLICITLY
COEFCTR         EQUALS  MPAC    +4
MASSCTR         EQUALS  MPAC    +5
SCRATCHX        EQUALS  MPAC    +4      # SCRATCH AREA FOR DVOVSUB ROUTINE.
SCRATCHY        EQUALS  SCRATCHX +1
SCRATCHZ        EQUALS  SCRATCHX +2

DOCKTEMP        EQUALS  MPAC    +3      # RECORD OF CSMDOCKED BIT OF DAPBOOLS
EPSILON         EQUALS  MPAC    +1
-EPSILON        EQUALS  EPSILON
-.1875          DEC     -.18750
```

```
                BANK    20
                SETLOC  DAPS3
                BANK

                EBANK=  AOSQ

                COUNT*  $$/DAPAO

 -1             TS      INGTS           # ZERO INGTS IN ASCENT
1/ACCONT        CA      DB              # INITIALIZE DBVAL1,2,3
                EXTEND
                MP      BIT13
                TS      L               # 0.25 DB
                AD      A
                TS      DBVAL3          # 0.50 DB
                CS      DBVAL1
                AD      L
                TS      DBVAL2          # -.75 DB

GETAOSUV        INHINT
                CAE     AOSR            # COMPUTE AOSU AND AOSV BY ROTATING
                TS      L               #       AOSQ AND AOSR.
                CAE     AOSQ
                TC      IBNKCALL
                CADR    ROT-TOUV
                DXCH    AOSU

                RELINT
                CA      DAPBOOLS
                MASK    DRIFTBIT        # ZERO DURING ULLAGE AND POWERED FLIGHT.
                CCS     A               # IF DRIFTING LIGHT,
                CA      ONE             #       SET DRIFTER TO 1
                TS      DRIFTER         # SAVE TO TEST FOR DRIFTING FLIGHT LATER
                AD      ALLOWGTS        # NON-ZERO IF DRIFT OR GTS NEAR
                CCS     A
                CA      FLATVAL         # DRIFTING FLIGHT, STORE .8 IN FLAT
                TS      FLATEMP         # IN POWERED FLIGHT, STORE ZERO IN FLAT
                EXTEND
                BZF     DOPAXIS         # IF POWERED AND NO GTS, START P AXIS,
                CCS     DRIFTER         # OTHERWISE SET ZONE3LIM
                CA      ZONE3MAX        # 17.5 MS, SCALED AT 4 SECONDS.
                TS      Z3TEM

DOPAXIS         CA      1JACC           # 1JACC AT PI/4 = 2JACC AT PI/2 =
                                        # ANET AT PI/2 = ANET/ACOAST AT 2(6).
                AD      BIT9            # 1 + ANET/ACOAST AT 2(6)
                TS      FUNTEM

                CA      1JACC
```

```
                TC      INVERT
                INHINT                  # P AXIS DATA MUST BE CONSISTENT
                TS      1/ANETP         # SCALED AT 2(7)/PI.
                TS      1/ANETP +1

                CS      BIT9            # -1 AT 2(6)
                EXTEND
                MP      1/ANETP         # -1/ANET AT 2(13)/PI
                EXTEND
                DV      FUNTEM          # -1/(ANET + ANET**2/ACOAST) AT 2(7)/PI
                TS      PACCFUN
                TS      PACCFUN +1

                CA      1/.03           # NO AOS FOR P AXIS, ACOAST = AMIN
                TS      1/ACOSTP
                TS      1/ACOSTP +1
                RELINT

                ZL
                CCS     DRIFTER
                DXCH    AOSU            # ZERO AOSU,V IF IN DRIFT, JUST TO BE SURE

UAXIS           CA      ZERO            # DO U AXIS COMPUTATIONS
                TS      UV              # ZERO FOR U AXIS, ONE FOR V AXIS.

BOTHAXES        TS      SIGNAOS         # CODING COMMON TO U,V AXES
                INDEX   UV
                CCS     AOSU            # PICK UP ABS(AOSU OR AOSV)
                AD      ONE             # RESTORE TO PROPER VALUE
                TCF     +3              # AND LEAVE SIGNAOS AT ZERO
                AD      ONE             # NEGATIVE, RESTORE TO PROPER VALUE
                INCR    SIGNAOS         # AND SET SIGNAOS TO ONE TO SHOW AOS NEG
                TS      ABSAOS          # SAVE ABS(AOS)
                CS      SIGNAOS
                TS      -SIGNAOS        # USED AS AN INDEX

                CA      DBVAL1          # SET DB1, DB2 TO DBVAL1 (= DB)
                TS      DBB1
                TS      DBB2

                CA      ABSAOS          # TEST MAGNITUDE OF ABS(AOS)
                AD      -.03R/S2
                EXTEND
                BZMF    NOTMUCH         # ABS(AOS) LESS THAN AMIN
BIGAOS          CCS     FLATEMP         # AGS(AOS) GREATER THAN AMIN
                TCF     SKIPDB1         # I DRIFT OR GTS, DO NOT COMPUTE DB

                CA      DBVAL1
                INDEX   -SIGNAOS
```

```
                ADS     DBB2            # DB2(1) = 2 DB
                INDEX   SIGNAOS
                TS      DBB4            # DB4(3) = 1 DB
                CA      -.1875          # -.1875 PI/2 RAD/SEC(2) SCALED AT PI/2
                AD      ABSAOS          # ABSAOS IS SCALED AT PI/2
                EXTEND
                BZMF    +3
                CS      DBVAL3          # -.5 DB
                TCF     DBONE
                CS      ABSAOS
                DOUBLE
                DOUBLE
                AD      BIT14
                DOUBLE                  # 1-8 ABSAOS.  (8 IS 16/PI SCALED AT 2/PI)
                EXTEND
                MP      DB
DBONE           INDEX   SIGNAOS         # DB1(2)=(1-8 ABSAOS) DB.  IF ABSAOS IS
                TS      DBB1            # GREATER THAN .1875 THEN DB1(2) = -.5 DB
                CA      DBVAL2
                INDEX   -SIGNAOS
                TS      DBB3            # DB3(4) = -.75 DB

SKIPDB1         CA      ABSAOS          # ABS(AOS) GREATER THAN AMIN, SO IT IS
                EXTEND
                MP      BIT12
                AD      ABSAOS          # (9/8) ABSAOS.
                TC      INVERT          # ALL RIGHT TO DIVIDE
                INDEX   -SIGNAOS
                TS      1/ACOSTT +1     # 1/ACOASTPOS(NET) = 1/ABS(AOS)
                CA      1/.03
                INDEX   SIGNAOS
                TS      1/ACOSTT        # 1/ACOASTNEG(POS) = 1/AIN

                CA      ABSAOS
                AD      1JACCU
                AD      1JACCU          # 2 JACC + ABS(AOS)
                AD      BIT9            # MAXIMUM VALUE IN COMPUTATIONS
                TS      A               # TEST FOR OVERFLOW
                TCF     SKIPDB2         # NO OVERFLOW, DO NORMAL COMPUTATION

                CA      ABSAOS          # RESCALE TO PI TO PREVENT OVERFLOW
                EXTEND
                MP      BIT14
                AD      1JACCU          # 1 JACC AT PI/2 = 2JACC AT PI
                TS      ANET            # ANETPOS(NEG) MAX SCALED AT PI =
                                        # ANETPOS(NEG) MAX/ACOASTNEG(POS) AT 2(7)
                AD      BIT8            # 1 + ANETPOS/ACOASTNEG AT 2(7)
                XCH     ANET            # SAVE IN ANET, WHILE PICKING UP ANET
                TC      INVERT
                EXTEND
```

```
                MP      BIT14           # SCALE 1/ANET AT 2(7)/PI
                TS      1/ANET

                CA      ACCHERE         # SET UP RETURN FROM COMPUTATION ROUTINE
                TS      ARET
                CS      BIT8            # -1 AT 2(7)
                TCF     DOACCFUN        # FINISH ACCFUN COMPUTATION

ACCHERE         TCF     ACCTHERE

NOTMUCH         TS      L               # ABS(AOS) LESS THAN AMIN, SAVE IN L
                CA      1/.03           # ACOASTPOS,NEG = AMIN
                TS      1/ACOSTT
                TS      1/ACOSTT +1

                CCS     FLATEMP
                TCF     SKIPDB2         # DO NOT COMPUTE DB IF DRIFT OR GTS

                CA      .0125RS         # AMIN/2
                AD      L               # L HAS ABS(AOS) - AMIN
                EXTEND                  # RESULT IS ABS(AOS)- AMIN/2
                BZMF    NOAOS           # ABS(AOS) LESS THAN AMIN/2

SOMEAOS         CA      DBVAL3          # AMIN/2 LT ABS(AOS) LT AMIN
                INDEX   -SIGNAOS
                TS      DBB3            # DB3(4) = DB/2
                AD      A
                INDEX   SIGNAOS
                TS      DBB4            # DB4(3) = DB
                TCF     SKIPDB2

NOAOS           CA      DBVAL1
                TS      DBB3            # DB3,4 = DB
                TS      DBB4

SKIPDB2         CA      ABSAOS          # ANETPOS(NEG) MAX = 2 JACC + ABS(AOS)
                AD      1JACCU
                AD      1JACCU
                TS      ANET            # CANNOT OVERFLOW HERE
CL1/NET+        TC      DO1/NET+        # COMPUTE 1/ANET, ACCFUN

ACCTHERE        INDEX   -SIGNAOS
                TS      Z5TEM   +2      # STORE ACCFUN IN TEMPORARY BUFFER
                CA      1/ANET
                INDEX   -SIGNAOS
                TS      1/ATEM2 +2      # STORE 1/ANET IN TEMPORARY BUFFER

                CA      ABSAOS          # SEE IF OVERFLOW IN MIN CASE
                AD      1JACCU
```

```
                AD      BIT9            # MAXIMUM POSSIBLE VALUE
                TS      A               # OVERFLOW POSSIBLE BUT REMOTE
                TCF     +2
                CA      POSMAX          # IF OVERFLOW, TRUNCATE TO PI/2
                AD      -.03R/S2        # RESTORE TO CORRECT VALUE
                TS      ANET
                TC      DO1/NET+        # COMPUTE 1/ANET, ACCFUN

                INDEX   -SIGNAOS        # STORE MIN VALUES JUST AS MAX VALUES
                TS      Z5TEM
                CA      1/ANET
                INDEX   -SIGNAOS
                TS      1/ATEM2

                CS      ABSAOS          # NOW DO NEG(POS) CASES
                AD      1JACCU
                AD      1JACCU          # ANETNEG(POS) MAX
                TC      1/ANET-         # COMPUTE 1/ANET, ACCFUN, AND ACCSW
                INDEX   SIGNAOS         # STORE NEG(POS) VALUES JUST AS POS(NEG)
                TS      Z1TEM   +2
                TS      L               # SAVE IN L FOR POSSIBLE FUTURE USE
                CA      1/ANET
                INDEX   SIGNAOS
                TS      1/ATEM1 +2
                CS      ABSAOS
                AD      1JACCU          # 1/ANETNEG(POS) MIN
                TS      ANET
                AD      -.03R/S2        # TEST FOR AMIN
                EXTEND                  # IF ANET LESS THAN AMIN, STORE MAX JET
                BZMF    FIXMIN          # VALUES FOR MIN JETS AND SET ACCSW

                TC      1/NETMIN        # OTHERWISE DO MIN JET COMPUTATIONS
STMIN-          INDEX   SIGNAOS         # STORE VALUES
                TS      Z1TEM
                CA      1/ANET
                INDEX   SIGNAOS
                TS      1/ATEM1

                INDEX   UV
                CA      +UMASK
                MASK    CH5MASK         # TEST FOR +U (+V) JET FAILURES
                EXTEND
                BZF     FAIL-
                CA      1/ATEM2         # REPLACE FUNCTION VALUES DEPENDING ON THE
                TS      1/ATEM2 +2      # FAILED JET PAIR WITH CORRESPONDING ONE-
                CA      Z5TEM           # JET (OR AMIN) FUNCTION VALUES
                TS      Z5TEM   +2
FAIL-           INDEX   UV
```

```
                CA      -UMASK
                MASK    CH5MASK         # TEST FOR -U (-V) JET FAILURES
                EXTEND
                BZF     DBFUN
                CA      1/ATEM1         # REPLACE FUNCTION VALUES DEPENDING ON THE
                TS      1/ATEM1 +2      # FAILED JET PAIR WITH CORRESPONDING ONE-
                CA      Z1TEM           # JET (OR AMIN) FUNCTION VALUES
                TS      Z1TEM   +2

DBFUN           CS      DBB3            # COMPUTE AXISDIST
                AD      DBB1
                AD      FLATEMP
                TS      AXDSTEM
                CS      DBB4
                AD      DBB2
                AD      FLATEMP
                TS      AXDSTEM +1

                INHINT
                CCS     UV              # TEST FOR U OR V AXIS
                TCF     STORV           # V AXIS        STORE V VALUES

                CA      ACCSW           # U AXIS        STORE U VALUES
                TS      ACCSWU

                CA      NINE            # TRANSFER 10 WORDS VIA GENTRAN
                TC      GENTRAN +1
                ADRES   1/ATEM1         # TEMPORARY BUFFER
                ADRES   1/ANET1         # THE REAL PLACE

                RELINT
                DXCH    DBB1            # SAVE U DBS FOR LATER STORING
                DXCH    UDB1
                DXCH    DBB4
                DXCH    UDB4

                DXCH    AXDSTEM
                DXCH    UAXDIST

                CA      ONE             # NOW DO V AXIS
                TS      UV
                CA      ZERO
                TCF     BOTHAXES        # AND DO IT AGAIN

STORV           CA      ACCSW           # STORE V AXIS VALUES
                TS      ACCSWV
                CA      NINE
                TC      GENTRAN +1
```

```
                ADRES   1/ATEM1         # TEMPORARY BUFFER
                ADRES   1/ANET1 +16D    # THE REAL PLACE

                                        # NOW STORE DEADBANDS FOR ALL AXES
                DXCH    FLATEMP         # FLAT AND ZONE3LIM
                DXCH    FLAT

                CA      DBVAL1          # COMPUTE P AXIS DEADBANDS
                TS      PDB1
                TS      PDB2
                AD      FLAT
                TS      PDB3
                TS      PDB4
                CA      ZERO
                TS      PAXDIST
                TS      PAXDIST +1

                CCS     FLAT
                TCF     DRFDB           # DRIFT OR GTS -- COMPUTE DBS

                DXCH    UDB1            # STORE U DEADBANDS
                DXCH    FIREDB          # CANNOT USE GENTRAN BECAUSE OF RELINT
                DXCH    UDB4
                DXCH    COASTDB
                DXCH    UAXDIST
                DXCH    AXISDIST
                DXCH    DBB1            # STORE V AXIS DEADBANDS
                DXCH    FIREDB  +16D    # COULD USE GENTRAN IF DESIRED
                DXCH    DBB4
                DXCH    COASTDB +16D
                DXCH    AXDSTEM
                DXCH    AXISDIST +16D

                TCF     1/ACCRET +1     # ALL DONE
DRFDB           CA      DBVAL1          # DRIFT DEADBANDS
                TS      FIREDB
                TS      FIREDB  +1
                TS      FIREDB  +16D
                TS      FIREDB  +17D
                AD      FLAT
                TS      COASTDB
                TS      COASTDB +1
                TS      COASTDB +16D
                TS      COASTDB +17D
                CA      ZERO
                TS      AXISDIST
                TS      AXISDIST +1
                TS      AXISDIST +16D
                TS      AXISDIST +17D
```

```
1/ACCRET        INHINT
                CS      DAPBOOLS        # SET BIT TO INDICATE DATA GOOD.
                MASK    ACCSOKAY
                ADS     DAPBOOLS
                RELINT
                CA      ACCRETRN
                TC      BANKJUMP        # RETURN TO CALLER

INVERT          TS      HOLD            # ROUTINE TO INVERT -INPUT AT PI/2
                CA      BIT9            # 1 AT 2(6)
                ZL                      # ZERO L FOR ACCURACY AND TO PREVENT OVFLO
                EXTEND
                DV      HOLD
                TC      Q               # RESULT AT 2(7)/PI

DOWNGTS         CAF     ZERO            # ZERO SWITCHES WHEN USEQRJTS BIT IS UP
                TS      ALLOWGTS        #       OR DAP IS OFF
                TS      INGTS
                TCF     DOCKTEST

1/ANET-         ZL
                LXCH    ACCSW           # ZERO ACCSW
                TS      ANET            # SAVE ANET
                AD      -.03R/S2        # TEST FOR MIN VALUE
                EXTEND
                BZMF    NETNEG          # ANET LESS THAN AMIN, SO FAKE IT
1/NETMIN        CA      ANET
                EXTEND
                INDEX   -SIGNAOS
                MP      1/ACOSTT +1     # ANETNEG(POS)/ACOASTPOS(NEG) AT 2(6)

# THE FOLLOWING CODING IS VALID FOR BOTH POS OR NEG
#       VALUES OF AOS

DO1/NET+        AD      BIT9            # 1 + ANET/ACOAST AT 2(6)
                XCH     ANET            # SAVE AND PICK UP ANET
                EXTEND
                QXCH    ARET            # SAVE RETURN
                TC      INVERT
                TS      1/ANET          # 1/ANET AT 2(7)/PI
                CS      BIT9            # -1 AT 2(6)
DOACCFUN        EXTEND
                MP      1/ANET          # -1/ANET AT 2(13)/PI
                EXTEND
                DV      ANET            # ACCFUN AT 2(7)/PI
                TC      ARET            # RETURN

NETNEG          CS      -.03R/S2        # ANET LESS THAN AMIN -- SET EQUAL TO AMIN
                TS      ANET
```

```
                TCF     1/NETMIN +1     # CONTINUE AS IF NOTHING HAPPENED.

FIXMIN          CCS     SIGNAOS
                CA      TWO             # IF AOS NEG, ACCSW = +1
                AD      NEGONE          # IF AOS POS, ACCSW = -1
                TS      ACCSW
                AD      UV              # IF ACCSW = +1, TEST FOR +U (+V) JET FAIL
                INDEX   A               # IF ACCSW = -1, TEST FOR -U (-V) JET FAIL
                CA      -UMASK  +1
                MASK    CH5MASK
                EXTEND
                BZF     +4
                CS      -.03R/S2        # JET FAILURE -- CANNOT USE 2-JET VALUES
                TS      ANET            # ANET = AMIN
                TCF     STMIN-  -1      # CALCULATE FUNCTIONS USING AMIN
                CA      L               # L HAS ACCFUN
                TCF     STMIN-          # STORE MAX VALUES FOR MIN JETS

# ERASABLE ASSIGNMENTS FOR 1/ACCONT

1/ANETP         EQUALS  BLOCKTOP +2
1/ACOSTP        EQUALS  BLOCKTOP +4
PACCFUN         EQUALS  BLOCKTOP +8D
PDB1            EQUALS  BLOCKTOP +10D
PDB2            EQUALS  BLOCKTOP +11D
PDB4            EQUALS  BLOCKTOP +12D
PDB3            EQUALS  BLOCKTOP +13D
PAXDIST         EQUALS  BLOCKTOP +14D

ACCSW           EQUALS  VBUF            # EXECUTIVE TEMPORARIES
                                        # CANNOT DO CCS NEWJOB DURING 1/ACCS
1/ATEM1         EQUALS  ACCSW   +1      # TEMP BUFFER FOR U AND V AXES
1/ATEM2         EQUALS  1/ATEM1 +1
1/ACOSTT        EQUALS  1/ATEM1 +4
Z1TEM           EQUALS  1/ATEM1 +6
Z5TEM           EQUALS  1/ATEM1 +7

UDB1            EQUALS  1/ATEM1 +10D    # UAXIS DEADBAND BUFFER
UDB2            EQUALS  1/ATEM1 +11D
UDB4            EQUALS  1/ATEM1 +12D
UDB3            EQUALS  1/ATEM1 +13D
UAXDIST         EQUALS  1/ATEM1 +14D

DBB1            EQUALS  1/ATEM1 +16D    # TEMP DEADBAND BUFFER, ALSO V AXIS
DBB2            EQUALS  1/ATEM1 +17D
DBB4            EQUALS  1/ATEM1 +18D
DBB3            EQUALS  1/ATEM1 +19D
AXDSTEM         EQUALS  1/ATEM1 +20D
```

```
FLATEMP         EQUALS  1/ATEM1 +22D
Z3TEM           EQUALS  1/ATEM1 +23D    # MUST FOLLOW FLATEMP

DBVAL1          EQUALS  DB
DBVAL2          EQUALS  INTB15+
DBVAL3          EQUALS  INTB15+ +1

DRIFTER         EQUALS  INTB15+ +2

UV              EQUALS  MPAC
ANET            EQUALS  MPAC    +3
FUNTEM          EQUALS  MPAC    +3
1/ANET          EQUALS  MPAC    +4
ARET            EQUALS  MPAC    +5
ABSAOS          EQUALS  MPAC    +6
SIGNAOS         EQUALS  MPAC    +7
-SIGNAOS        EQUALS  MPAC    +8D
HOLD            EQUALS  MPAC    +9D
ACCRETRN        EQUALS  FIXLOC  -1

ZONE3MAX        DEC     .004375         # 17.5 MS (35 MS FOR 1 JET) AT 4 SECONDS
FLATVAL         DEC     .01778          # .8 AT PI/4 RAD
-.03R/S2        OCT     77377           # -PI/2(7) AT PI/2

.0125RS         EQUALS  BIT8            # PI/2(+8) AT PI/2
1/.03           EQUALS  POSMAX          # 2(7)/PI AT 2(7)/PI

PAXISADR        GENADR  PAXIS

                                        # THE FOLLOWING 4 CONSTANTS ARE JET
                                        # FAILURE MASKS AND ARE INDEXED
-UMASK          OCT     00110           # -U
                OCT     00022           # -V
+UMASK          OCT     00204           # +U
                OCT     00041           # +V
```

```
# PROGRAM NAME:         SPSRCS
# AUTHOR:               EDGAR M. OSHIKA (AC ELECTRONICS)
# MODIFIED:             TO RETURN TO ALL AXES VIA Q BY P. S. WEISSMAN, OCT 7, 1968
# MODIFIED TO IMPROVE BENDING STABILITY BY G. KALAN, FEB. 14, 1969
#
# FUNCTIONAL DESCRIPTION:
#       THE PROGRAM CONTROLS THE FIRING OF ALL RCS JETS IN THE DOCKED CONFIGURATION ACCORDING TO THE FOLLOWING PHASE
#       PLANE LOGIC.
#
#       1. JET SENSE TEST (SPSRCS)
#               IF JETS ARE FIRING NEGATIVELY, SET OLDSENSE NEGATIVE AND CONTINUE
#               IF JETS ARE FIRING POSITIVELY, SET OLDSENSE POSITIVE AND CONTINUE
#               IF JETS ARE NOT FIRING, SET OLDSENSE TO ZERO AND GO TO OUTER RATE LIMIT TEST
#
#       2. RATE DEAD BAND TEST
#               IF JETS ARE FIRING NEGATIVELY AND RATE IS GREATER THAN TARGET RATE, LEAVE
#                       JETS ON AND GO TO INHIBITION LOGIC.  OTHERWISE, CONTINUE.
#               IF JETS ARE FIRING POSITIVELY AND RATE IS   LESS  THAN TARGET RATE, LEAVE
#                       JETS ON AND GO TO INHIBITION LOGIC.  OTHERWISE, CONTINUE.
#
#       3. OUTER RATE LIMIT TEST (SPSSTART)
#               IF MAGNITUDE OF EDOT IS GREATER THAN 1.73 DEG/SEC SET JET FIRING TIME
#                       TO REDUCE RATE AND GO TO INHIBITION LOGIC.  OTHERWISE, CONTINUE.
#
#       4. COAST ZONE TEST
#               IF STATE (E,EDOT) IS BELOW LINE E + 4 X EDOT > -1.4 DEG AND EDOT IS LESS THAN 1.30 DEG/SEC SET JET TIME
#                       POSITIVE AND CONTINUE.  OTHERWISE, SET JET FIRING TIME TO ZERO AND CONTINUE.
#               IF STATE IS ABOVE LINE E + 4 X EDOT > +1.4 DEG AND EDOT IS GREATER THAN -1.30 DEG/SEC, SET JET TIME NEGATIVE
#                       AND CONTINUE.  OTHERWISE, SET JET FIRING TIME TO ZERO AND CONTINUE.
#
#       5. INHIBITION LOGIC
#               IF OLDSENSE IS NON-ZERO:
#                       A) RETURN IF JET TIME AS THE SAME SIGN AS OLDSENSE
#                       B) SET INHIBITION COUNTER* AND RETURN IF JET TIME IS ZERO
#                       C) SET INHIBITION COUNTER,* SET JET TIME TO ZERO AND RETURN IF SIGN
#                          OF JET TIME IS OPPOSITE TO THAT OF OLDSENSE
#               IF OLDSENSE IS ZERO:
#                       A) RETURN IF INHIBITION COUNTER IS NOT POSITIVE
#                       B) SET JET TIME TO ZERO AND RETURN IF INHIBITION COUNTER IS POSITIVE
#               *NOTE: INHIBITION COUNTERS CAN BE SET TO 4 OR 10 FOR THE P AND UV AXES,
#               RESPECTIVELY, IN SPSRCS.  THEY ARE DECREMENTED BY ONE AT THE BEGINNING OF
```

```
#               EACH DAP PASS.
#
#       THE MINIMUM PULSE WIDTH OF THIS CONTROLLER IS DETERMINED BY THE REPETITION RATE AT WHICH THIS ROUTINE IS CALLED
#       AND IS NOMINALLY 100 MS FOR ALL AXES IN DRIFTING FLIGHT.  DURING POWERED FLIGHT THE MINIMUM IS 100 MS FOR THE
#       P AXIS AND 200 MS FOR THE CONTROL OF THE U AND V AXES.
#
# CALLING SEQUENCE:
#               INHINT
#               TC      IBNKCALL
#               CADR    SPSRCE
#
# EXIT:
#               TC      Q
#
# ALARM/ABORT MODE:     NONE
#
# SUBROUTINES CALLED:   NONE
#
# INPUT:                E, EDOT
#                       TJP, TJV, TJU           TJ MUST NOT BE NEGATIVE ZERO
#
# OUTPUT:               TJP, TJV, TJU

                BANK    21
                SETLOC  DAPS4
                BANK

                COUNT*  $$/DAPBU

                EBANK=  TJU
RATELIM2        OCT     00632           # 1.125 DEG/SEC
POSTHRST        CA      HALF

                NDX     AXISCTR
                TS      TJU
                CCS     OLDSENSE
                TCF     POSCHECK        # JETS FIRING POSITIVELY
                TCF     CTRCHECK        # JETS OFF.  CHECK INHIBITION CTR
NEGCHECK        INDEX   AXISCTR         # JETS FIRING NEGATIVELY
                CS      TJU
                CCS     A
                TC      Q               # RETURN
                TCF     +2
                TCF     +1              # JETS COMMANDED OFF.  SET CTR AND RETURN
SETCTR          INDEX   AXISCTR         # JET FIRING REVERSAL COMMANDED.  SET CTR,
                CA      UTIME           # SET JET TIME TO ZERO, AND RETURN
```

```
                INDEX   AXISCTR
                TS      UJETCTR
ZAPTJ           CA      ZERO
                INDEX   AXISCTR
                TS      TJU
                TC      Q
POSCHECK        INDEX   AXISCTR
                CA      TJU
                TCF     NEGCHECK +2
CTRCHECK        INDEX   AXISCTR         # CHECK JET INHIBITION COUNTER
                CCS     UJETCTR
                TCF     +2
                TC      Q               # CTR IS NOT POSITIVE.  RETURN
                TCF     ZAPTJ           # CTR IS POSITIVE.  INHIBIT FIRINGS
                TC      Q               # CTR IS NOT POSITIVE.  RETURN
                OCT     00004
UTIME           OCT     00012
                OCT     00012
OLDSENSE        EQUALS  DAPTREG1
NEGFIRE         CS      ONE             # JETS FIRING NEGATIVELY
                TS      OLDSENSE
                CA      EDOT
                TCF     +4
PLUSFIRE        CA      ONE
                TS      OLDSENSE
                CS      EDOT            # RATE DEAD BAND TEST
                LXCH    A
                CS      DAPBOOLS        # IF DRIFTBIT = 1, USE ZERO TARGET RATE
                MASK    DRIFTBIT        # IF DRIFTBIT = 0, USE 0.10 RATE TARGET
                CCS     A
                CA      RATEDB1
                AD      L
                EXTEND
                BZMF    SPSSTART
                TCF     POSTHRST +3

SPSRCS          INDEX   AXISCTR         # JET SENSE TEST
                CCS     TJU
                TCF     PLUSFIRE        # JETS FIRING POSITIVELY
                TCF     +2
                TCF     NEGFIRE         # JETS FIRING NEGATIVELY
                TS      OLDSENSE        # JETS OFF
SPSSTART        CA      EDOT            # OUTER RATE LIMIT TEST
                EXTEND
                MP      RATELIM1
                CCS     A
                TCF     NEGTHRST        # OUTER RATE LIMIT EXCEEDED
                TCF     +2
                TCF     POSTHRST        # OUTER RATE LIMIT EXCEEDED
                CA      EDOT            # COAST ZONE TEST
```

```
                AD      E
                EXTEND
                MP      DKDB            # PAD LOADED DEADBAND.  FRESHSTART: 1.4 DEG
                EXTEND
                BZF     TJZERO

                EXTEND
                BZMF    +7
                CA      EDOT
                AD      RATELIM2
                EXTEND
                BZMF    TJZERO
NEGTHRST        CS      HALF
                TCF     POSTHRST +1
 +7             CS      RATELIM2
                AD      EDOT
                EXTEND
                BZMF    POSTHRST
TJZERO          CA      ZERO
                TCF     POSTHRST +1

RATELIM1        =       CALLCODE        # = 00032, CORRESPONDING TO 1.73 DEG/SEC
RATEDB1         =       TBUILDFX        # = 00045, CORRESPONDS TO 0.101 DEG/SEC

# *** END OF LMDAP  .015 ***
```