Reverse Engineering the Apollo 11 Guidance Computer (AGC)
Source Code for Lunar Module (Luminary099)

Edited by Gary Young

proudindiv@gmail.com

July 28, 2016

Contents

Chapter 1

Luminary099

7 ⟨*Luminary099.agc* 7⟩≡
⟨*controlled constants* 8⟩
⟨*input output channel bit descriptions* 24⟩
⟨*flagword assignments* 31⟩
⟨*rcs failure monitor* 64c⟩
⟨*ags initialization* 70⟩
⟨*aotmark routine* 78b⟩
⟨*lem geometry* 101⟩
⟨*r63 routine* 107b⟩
⟨*attitude maneuver routine* 112⟩
⟨*imu performance test 2* 149b⟩
⟨*imu performance tests 4* 163a⟩
⟨*s band antenna for lm* 173b⟩
⟨*radar leadin routines* 179b⟩
⟨*p30-p37 routines* 181b⟩
⟨*p32-p35 p72-p75 routines* 187⟩
⟨*lambert aimpoint guidance* 236⟩
⟨*burn-baby-burn master ignition routine* 241⟩
⟨*the lunar landing* 278⟩
⟨*throttle control routines* 289b⟩
⟨*lunar landing guidance equations* 296b⟩
⟨*p70-p71 routines* 348⟩
⟨*p12 routine* 364b⟩
⟨*ascent guidance* 372b⟩
⟨*servicer routine* 396⟩
⟨*landing analog displays* 452b⟩
⟨*findcduw-guidap interface* 472⟩
⟨*lm down-telemetry program* 493⟩
⟨*interpretive constant* 507b⟩
⟨*lm agc block two self check* 509b⟩
⟨*rtb op codes* 527b⟩

⟨*t6 rupt programs* 533b⟩
⟨*dap interface subroutines* 538a⟩
⟨*dapidler program* 542b⟩
⟨*p axis rcs autopilot* 557b⟩
⟨*q-r axis rcs autopilot* 596⟩
⟨*tjet law* 627b⟩
⟨*kalman filter* 645⟩
⟨*trim gimbal cntrol system* 647b⟩
⟨*aostask and aosjob* 667⟩
⟨*sps back up rcs control* 705⟩

This code is written to file `Luminary099.agc`.

1.1 controlled constants

8 ⟨*controlled constants* 8⟩≡ (7)
⟨*Page LM0038* 9⟩
⟨*Page LM0039* 10⟩
⟨*Page LM0040* 11⟩
⟨*Page LM0041* 12⟩
⟨*Page LM0042* 13⟩
⟨*Page LM0043* 14⟩
⟨*Page LM0044* 15⟩
⟨*Page LM0045* 16⟩
⟨*Page LM0046* 17⟩
⟨*Page LM0047* 18⟩
⟨*Page LM0048* 19⟩
⟨*Page LM0049* 20⟩
⟨*Page LM0050* 21⟩
⟨*Page LM0051* 22⟩
⟨*Page LM0052* 23a⟩
⟨*Page LM0053* 23b⟩

9 ⟨*Page LM0038* 9⟩≡ (8 724)

```
# 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
```

10 ⟨*Page LM0039* 10⟩≡ (8 724)

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

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

(AT)A           2DEC    3.2883 E-4 B9   # INITIAL ASC. STG. ACCELERATION ** N
                                        # ASSUMPTIONS SAME AS FOR (1/DV)A.
(TBUP)A         2DEC    91902 B-17      # ESTIMATED BURN-UP TIME OF THE ASCEN
                                        # ASSUMPTIONS SAME AS FOR (1/DV)A WIT
                                        # ADDITIONAL ASSUMPTION THAT NET MASS
                                        # 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
```

11 ⟨*Page LM0040* 11⟩≡ (8 724)

```
# 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)
```

12 ⟨*Page LM0041* 12⟩≡ (8 724)

```
# 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
RANGCONV        2DEC    2.859024 B-3            # CONVERTS RR RANGE READING TO M. AT

                SETLOC  SERVICES
                BANK
                COUNT*  $$/SERV

HBEAMANT        2DEC    -.4687018041            # RANGE BEAM IN LR ANTENNA COORDINATE
                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/
VYSCAL          2DEC    +.7565672446            # SCALES 1.212 FT/SEC/BIT TO 2(18) M/
VXSCAL          2DEC    -.4020043770            # SCALES -.644 FT/SEC/BIT TO 2(18) M/

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

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.
```

13 ⟨*Page LM0042* 13⟩≡ (8 724)

```
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
```

14 ⟨*Page LM0043* 14⟩≡ (8 724)

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

                SETLOC  FRANDRES
                BANK
                COUNT*  $$/START

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

                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) = 905

                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)
```

15 ⟨*Page LM0044* 15⟩≡ (8 724)

```
# 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*
```

16 ⟨*Page LM0045* 16⟩≡ (8 724)
```
 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)

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

17 ⟨*Page LM0046* 17⟩≡ (8 724)

```
                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.
```

18 ⟨*Page LM0047* 18⟩≡ (8 724)

```
# 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
```

19 ⟨*Page LM0048* 19⟩≡ (8 724)

```
        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
```

20 ⟨*Page LM0049* 20⟩≡ (8 724)

```
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
```

21 ⟨*Page LM0050* 21⟩≡ (8 724)

```
                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
```

22 ⟨*Page LM0051* 22⟩≡ (8 724)

```
                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)        *
```

23a ⟨*Page LM0052* 23a⟩≡ (8 724)

```
                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
```

23b ⟨*Page LM0053* 23b⟩≡ (8 724)

```
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
```

1.2 input output channel bit descriptions

24 ⟨*input output channel bit descriptions* 24⟩≡ (7)
  ⟨*Page LM0054* 25⟩
  ⟨*Page LM0055* 26⟩
  ⟨*Page LM0056* 27⟩
  ⟨*Page LM0057* 28⟩
  ⟨*Page LM0058* 29⟩
  ⟨*Page LM0059* 30a⟩
  ⟨*Page LM0060* 30b⟩

25 ⟨*Page LM0054* 25⟩≡ (24 739)

```
# *** CHANNEL DESCRIPTION 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.
#               FACTOR IS B23 IN CSEC, SO MAX VALUE ABOUT 23.3 HOURS AND LEAST SIGNIFICANT BIT

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

# CHANNEL 5     PYJETS:  OUTPUT CHANNEL; PITCH RCS JET CONTROL.  (REACTION CONTROL SYSTEM) USES

# CHANNEL 6     ROLLJETS: OUTPUT CHANNEL; ROLL RCS JET CONTROL.  (REACTION CONTROL SYSTEM) USES

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

# CHANNEL 10    OUT0: OUTPUT CHANNEL; REGISTER USED TO TRANSMIT LATCHING-RELAY DRIVING INFORMAT
#               THE DISPLAY SYSTEM.  BITS 15-12 ARE SET TO THE ROW NUMBER (1-14 OCTAL) OF THE R
#               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 CONTRO
#               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
```

26 ⟨*Page LM0055* 26⟩≡ (24 739)

```
#               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 SPACECRAFT
#
#               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 RENDEZVOUS RADAR LOCK-ON; AUTO ANGLE TRACK'G
#               BIT 15          ISS TURN ON DELAY COMPLETE
```

27 ⟨*Page LM0056* 27⟩≡ (24 739)

```
# 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 IS 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, SPAC
#
#               BIT 1           OUTLINK ACTIVITY (NOT USED)
#               BIT 2           ALTITUDE RATE OR ALTITUDE 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.
```

28 ⟨*Page LM0057* 28⟩≡ (24 739)

```
#               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
#               PROGRAM INTERRUPT #5 IS RECEIVED.  USED BITS 5-1

# CHANNEL 16    NAVKEYIN: INPUT CHANNEL; OPTICS MARK INFORMATION AND NAVIGATION PANEL
#               CONTROL (LM) SENSED BY PROGRAM THEN PROGRAM INTERRUPT #6 IS RECEIVED
#
#               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
# 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
```

29 ⟨*Page LM0058* 29⟩≡ (24 739)

```
  #               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 CONT
  #               AND SPACECRAFT ATTITUDE CONTROL; USED BY RCS DAP.
  #
  #               BIT 1           ROTATION (BY RHC) COMMANDED IN POSITIVE PITCH DIRECTION; MUST B
  #                               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
  #               BIT 4           AS BIT 3 EXCEPT NEGATIVE YAW
  #               BIT 5           ROTATION (BY RHC) COMMANDED IN POSITIVE ROLL DIRECTION; MUST BE
  #                               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
```

30a ⟨*Page LM0059* 30a⟩≡ (24 739)
```
 #              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 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.
 #              FLOP BITS RESET BY A CHANNEL "WRITE" COMMAND THAT ARE RESET BY A REST
 #
 #              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
```

30b ⟨*Page LM0060* 30b⟩≡ (24 739)
```
 #              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, VO
 #              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: SECOND OF TWO WORDS SERIALIZATION
```

1.3 flagword assignments

31 ⟨*flagword assignments* 31⟩≡ (7)
⟨*Page LM0061* 32⟩
⟨*Page LM0062* 33⟩
⟨*Page LM0063* 35⟩
⟨*Page LM0064* 37⟩
⟨*Page LM0065* 39⟩
⟨*Page LM0066* 40⟩
⟨*Page LM0067* 41⟩
⟨*Page LM0068* 42⟩
⟨*Page LM0069* 43⟩
⟨*Page LM0070* 44⟩
⟨*Page LM0071* 45⟩
⟨*Page LM0072* 46⟩
⟨*Page LM0073* 48⟩
⟨*Page LM0074* 49⟩
⟨*Page LM0075* 50⟩
⟨*Page LM0076* 51⟩
⟨*Page LM0077* 52⟩
⟨*Page LM0078* 53⟩
⟨*Page LM0079* 54⟩
⟨*Page LM0080* 55⟩
⟨*Page LM0081* 56⟩
⟨*Page LM0082* 57⟩
⟨*Page LM0083* 59⟩
⟨*Page LM0084* 60⟩
⟨*Page LM0085* 61⟩
⟨*Page LM0086* 62⟩
⟨*Page LM0087* 63⟩
⟨*Page LM0088* 64a⟩
⟨*Page LM0089* 64b⟩

32 ⟨*Page LM0061* 32⟩≡ (31 733)

```
# FLAGWORDS 0-11        ARE DOWNLINKED AND CAN BE SET AND CLEARED BY UP-FLAG AND DOWN
#                       INTERPRETER.  THESE WERE PREVIOUSLY LISTED UNDER "INTERPRETIV
#                       THE ERASABLE LOG SECTION.  FLAGWORDS 12 & 13 WERE PREVIOUSLY 
#                       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
```

33 ⟨*Page LM0062* 33⟩≡ (31 733)

```
# 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: D
# 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
```

35 ⟨*Page LM0063* 35⟩≡ (31 733)

```
 # 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: B
 # 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
# RCDU0FLG      182            BIT 13 FLAG 12          RCDU0BIT
# READLR        174            BIT  6 FLAG 11          READLBIT
```

37 ⟨*Page LM0064* 37⟩≡ (31 733)

```
 # READRFLG      051             BIT  9 FLAG  3          READRBIT        EQUIVALENT FLAG NAME FO
 # 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:
 # 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
```

39 ⟨*Page LM0065* 39⟩≡ (31 733)

```
# 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
JSWCHBIT        =       BIT14                   #       MATRIX                  VECTOR

# BIT 13 FLAG 0 (S)
MIDFLAG         =       002D                    #       INTEGRATION WITH        INTEGRATION WIT
                                                #       SECONDARY BODY AND      SOLAR PERTURBAT
MIDFLBIT        =       BIT13                   #       SOLAR PERTURBATIONS

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

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

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

40 ⟨*Page LM0066* 40⟩≡ (31 733)

```
# BIT 9 FLAG 0  (S)
P25FLAG         =       006D                    #       P25 OPERATING           P25 N
P25FLBIT        =       BIT9

# BIT 8 FLAG 0  (S)
IMUSE           =       007D                    #       IMU IN USE              IMU N
IMUSEBIT        =       BIT8

# BIT 7 FLAG 0  (S)
RNDVZFLG        =       008D                    #       P20 RUNNING (RADAR      P20 N
RNDVZBIT        =       BIT7                    #       IN USE)

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

# BIT 5 FLAG 0  (S)
LOKONSW         =       010D                    #       RADAR LOCK-ON           RADAR
LOKONBIT        =       BIT5                    #       DESIRED                 DESIR

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

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

# BIT 2 FLAG 0
R10FLAG         =       013D                    #       R10 OUTPUTS DATA TO     BESID
R10FLBIT        =       BIT2                    #       ALTITUDE & ALTITUDE     SET,
                                                #       RATE METERS ONLY        TO FO
                                                #                               VELOC

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


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

41 ⟨*Page LM0067* 41⟩≡ (31 733)

```
                                                        #       (SET)                   (RESET)

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

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

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

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

    # 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 VECTO
    VEHUPBIT        =       BIT8                        #       BEING UPDATED           BEING UPDATED

    # BIT 7 FLAG 1  (S)
    UPDATFLG        =       023D                        #       UPDATING BY MARKS       UPDATING BY MAR
    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
```

42 ⟨*Page LM0068* 42⟩≡ (31 733)

```
# BIT 5 FLAG 1  (S)
TRACKFLG        =       025D                            #       TRACKING ALLOWED        TRACK
TRACKBIT        =       BIT5

# BIT 4 FLAG 1
                =       026D
                =       BIT4

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

# BIT 2 FLAG 1  (S)
GUESSW          =       028D                            #       NO STARTING VALUE       START
GUESSBIT        =       BIT2                            #       FOR ITERATION           ITERA

# BIT 1 FLAG 1
                =       029D
                =       BIT1                            # OH 2009-05-15 Scan does not have th

FLAGWRD2        =       STATE +2                        # (030-044)

                                                        #       (SET)                   (RESE

# BIT 15 FLAG 2 (S)
DRIFTFLG        =       030D                            #       T3RUPT CALLS GYRO       T3RUP
DRFTBIT         =       BIT15                           #       COMPENSATION            COMPE

# BIT 14 FLAG 2 (S)
SRCHOPTN        =       031D                            #       RADAR IN AUTOMATIC      RADAR
SRCHOBIT        =       BIT14                           #       SEARCH OPTION (R24)     MATIC

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

# BIT 12 FLAG 2 (S)
LOSCMFLG        =       033D                            #       LINE OF SIGHT BEING     LINE
                                                        #       COMPUTED (R21)          BEING
LOSCMBIT        =       BIT12
```

43 ⟨*Page LM0069* 43⟩≡ (31 733)

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

# BIT 10 FLAG 2 (S)
                =       035D                    # OH 2009-05-15 These two line don't appear in
                =       BIT10

# BIT 9 FLAG 2 (S)
IMPULSW         =       036D                    #       MINIMUM IMPULSE         STEERING BURN (
                                                #       BURN (CUTOFF TIME       CUTOFF TIME YET
IMPULBIT        =       BIT9                    #       SPECIFIED)              AVAILABLE)

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

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

# BIT 7 FLAG 2 (L)
OPTNSW          =       ETPIFLAG                #       SOI PHASE OF P38/78     SOR PHASE OF P3
OPTNBIT         =       BIT7

# BIT 6 FLAG 2 (S)
FINALFLG        =       039D                    #       LAST PASS THROUGH       INTERIM PASS TH
                                                #       RENDEZVOUS PROGRAM      RENDEZVOUS PROG
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 ATTIT
PFRATBIT        =       BIT4                    #       COMPUTED                NOT COMPUTED

# BIT 3 FLAG 2 (S)
```

44 ⟨*Page LM0070* 44⟩≡ (31 733)

```
CALCMAN3        =       042D                    #       NO FINAL ROLL           FINAL
CALC3BIT        =       BIT3                    #                               NECES

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

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

FLAGWRD3        =       STATE +3                # (045-059)

                                                #       (SET)                   (RESE

# BIT 15 FLAG 3
                =       045D                    #
                =       BIT15                   # OH 2009-05-15 This line is not in s

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

# BIT 13 FLAG 3 *** PROTECTED FROM FRESH START ***
REFSMFLG        =       047D                    #       REFSMMAT GOOD           REFSM
REFSMBIT        =       BIT13

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

# BIT 11 FLAG 3 (L)
NOR29FLG        =       049D                    #       R29 NOT ALLOWED         R29 A
NR29FBIT        =       BIT11                   #                               IGNAT

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

# BIT 9 FLAG 3  (S)
R04FLAG         =       051D                    #       ALARM 521               ALARM
                                                #       SUPPRESSED
```

45 ⟨*Page LM0071* 45⟩≡ (31 733)

```
R04FLBIT        =       BIT9

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

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

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

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

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

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

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

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

# BIT 1 FLAG 3  (S)
DIM0FLAG        =       059D                    #       W MATRIX IS TO BE       W MATRIX IS NOT
DIM0BIT         =       BIT1                    #       USED                    USED

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

46 ⟨*Page LM0072* 46⟩≡ (31 733)

```
                                                #       (SET)                   (RESE

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

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

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

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

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

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

# BIT 9 FLAG 4  (S)
MRKNVFLG        =       066D                    #       ASTRONAUT USING         ASTRO
                                                #       KEYBOARD WHEN MARK      KEYBO
MRKNVBIT        =       BIT9                    #       DISPLAY INITIATED       DISPL

# BIT 8 FLAG 4  (S)
NRMNVFLG        =       067D                    #       ASTRONAUT USING         ASTRO
                                                #       KEYBOARD WHEN           KEYBO
NRMNVBIT        =       BIT8                    #       NORMAL DISPLAY          NORMA
                                                #       INITIATED               INITI

# BIT 7 FLAG 4  (S)
PRONVFLG        =       068D                    #       ASTRONAUT USING         ASTRO
```

48 ⟨*Page LM0073* 48⟩≡ (31 733)

```
                                                #       KEYBOARD WHEN           KEYB
PRONVBIT        =       BIT7                    #       PRIORITY DISPLAY        PRIO
                                                #       INITIATED               INIT

# BIT 6 FLAG 4  (S)
PINBRFLG        =       069D                    #       ASTRONAUT HAS           ASTR
                                                #       INTERFERED WITH         INTE
PINBRBIT        =       BIT6                    #       EXISTING DISPLAY        EXIS

# BIT 5 FLAG 4  (S)
MRUPTFLG        =       070D                    #       MARK DISPLAY            MARK
                                                #       INTERRUPTED BY          INTE
MRUPTBIT        =       BIT5                    #       PRIORITY DISPLAY        PRIO

# BIT 4 FLAG 4  (S)
NRUPTFLG        =       071D                    #       NORMAL DISPLAY          NORM
                                                #       INTERRUPTED BY          INTE
NRUPTBIT        =       BIT4                    #       PRIORITY OR MARK        PRIO
                                                #       DISPLAY                 DISP

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

# BIT 2 FLAG 4
                =       073D
                =       BIT2                    # OH 2009-05-15 Not in scan.


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

FLAGWRD5        =       STATE +5                # (075-089)

                                                #       (SET)                   (RES

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

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

49 ⟨*Page LM0074* 49⟩≡ (31 733)

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

# BIT 12 FLAG 5 (S)
NOTHROTL        =       078D                    #       INHIBIT FULL            PERMIT FULL THR
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
                                                #       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
DMENFBIT        =       BIT9                    #       FOR INCORPORATION       FOR INCORPORATI

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

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

# BIT 6 FLAG 5  (S)
3AXISFLG        =       084D                    #       MANEUVER SPECIFIED      MANEUVER SPECIF
                                                #       BY THREE AXES           BY ONE AXIS; R6
3AXISBIT        =       BIT6                    #                               CALLS VECPOINT.

# BIT 5 FLAG 5
                =       085D
                =       BIT5                    # OH 2009-05-15 Not in scan

# BIT 4 FLAG 5  (S)
```

50 ⟨*Page LM0075* 50⟩≡ (31 733)
```
NORRMON         =       086D                    #       BYPASS RR GIMBAL        PERF
NORRMBIT        =       BIT4                    #       MONITOR                 RR GI

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

# BIT 2 FLAG 5  (S)
MGLVFLAG        =       088D                    #       LOCAL VERTICAL          MIDD
                                                #       COORDINATES             COMP
MGLVFBIT        =       BIT2                    #       COMPUTED

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


FLAGWRD6        =       STATE +6                # (090-104)

                                                #       (SET)                   (RES

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

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

# BIT 13 FLAG 6 (S)
S32.1F3A        =       092D                    # BIT 13 AND BIT 12 FUNCTION AS AN O
S32BIT3A        =       BIT13                   # PAIR (13,12) INDICATING THE POSSIB
                                                # CURRENCE OF 2 NEWTON ITERATIONS FO
                                                # IN THE PROGRAM IN THE FOLLOWING OR
# BIT 12 FLAG 6 (S)                             # (0,1) (I.E. BIT 13 RESET, BIT 12 S
S32.1F3B        =       093D                    #      = FIRST NEWTON ITERATION BEIN
S32BIT3B        =       BIT12                   # (0,0)= FIRST PASS OF SECOND NEWTON
                                                # (1,1)= 50 FT/SEC STAGE OF SECOND N
                                                # (1,0)= REMAINDER OF SECOND NEWTON

# BIT 11 FLAG 6 (S)
                =       094D                    #
                =       BIT11                   #
```

51 ⟨*Page LM0076* 51⟩≡ (31 733)

```
# BIT 10 FLAG 6 (S)
GMBDRVSW        =       095D                            #       TRIMGIMB OVER           TRIMGIMB NOT OV
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 N
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 N
                                                        #       OVERWRITE DELTA         OVERWRITE DELTA
NTARGBIT        =       BIT3                            #       VELOCITY AT TPI         VELOCITY
                                                        #       OR TPM (P34,35)

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

# BIT 1 FLAG 6  (L)
ATTFLAG         =       104D                            #       LEM ATTITUDE EXISTS     NO LEM ATTITUDE
```

```
                                                #       IN MOON-FIXED           AVAI
```

52 ⟨*Page LM0077* 52⟩≡ (31 733)

```
ATTFLBIT        =       BIT1                    #       COORDINATES             FIXE

FLAGWRD7        =       STATE +7                # (105-119)

                                                #       (SET)                   (RES

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

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

# BIT 13 FLAG 7 (S)
IGNFLAG         =       107D                    #       TIG HAS ARRIVED         TIG 
IGNFLBIT        =       BIT13                   #

# BIT 12 FLAG 7 (S)
ASTNFLAG        =       108D                    #       ASTRONAUT HAS           ASTR
ASTNBIT         =       BIT12                   #       OKAYED IGNITION         OKAY

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

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

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

# BIT 8 FLAG 7  (S)
V67FLAG         =       112D                    #       ASTRONAUT OVERWRITE     ASTR
                                                #       W-MATRIX INITIAL        OVER
V67FLBIT        =       BIT8                    #       VALUES                  INIT
```

53 ⟨Page LM0078 53⟩≡ (31 733)

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

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

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

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

54 ⟨*Page LM0079* 54⟩≡ (31 733)

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

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

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

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

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

# BIT 8 FLAG 8  *** PROTECTED FROM FRESH START ***
SURFFLAG        =       127D                    #       LM ON LUNAR SURFACE     LM N
SURFFBIT        =       BIT8                    #                               SURF

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

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

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

# BIT 4 FLAG 8  (S)
COGAFLAG        =       131D                    #       NO CONIC SOLUTION --    CONI
                                                #       TOO CLOSE TO RECTI-     EXIS
```

55 ⟨*Page LM0080* 55⟩≡ (31 733)

```
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 THR
INITABIT        =       BIT2                    #       P57                     (CHECK RESET-MI

# BIT 1 FLAG 8  (S)
360SW           =       134D                    #       TRANSFER ANGLE NEAR     TRANSFER ANGLE
360SWBIT        =       BIT1                    #       360 DEGREES             NEAR 360 DEGREE


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

                                                #       (SET)                   (RESET)

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

# BIT 14 FLAG 9 (L)
FLVR            =       136D                    #       VERTICAL RISE           NON-VERTICAL RI
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 CONTRO
FLPCBIT         =       BIT12                   #       (ASCENT GUIDANCE)

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

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

# BIT 9 FLAG 9  (L)
```

56 ⟨*Page LM0081* 56⟩≡ (31 733)

```
LETABORT        =       141D                        #       ABORT PROGRAMS          ABORT
LETABBIT        =       BIT9                        #       ARE ENABLED             ARE N

# BIT 8 FLAG 9  (L)
FLAP            =       142D                        #       APS CONTINUED ABORT     APS A
                                                    #       AFTER DPS STAGING       CONTI
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 A
ROTFLBIT        =       BIT6                        #       FORCE VEHICLE           FORCE
                                                    #       ROTATION IN THE         ROTAT
                                                    #       PREFERRED DIRECTION     PREFE

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

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

# BIT 3 FLAG 9  (L)
MID1FLAG        =       147D                        #       INTEGRAT TO TDEC        INTEG
MID1FBIT        =       BIT3                        #                               THEN-

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

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


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

57 ⟨*Page LM0082* 57⟩≡ (31 733)

```
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
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 ROU
REINTBIT        =       BIT7                    #       TO BE RESTARTED         NOT TO BE RESTA

# 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
```

59 ⟨*Page LM0083* 59⟩≡ (31 733)
```
# 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 L
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)
```

60 ⟨*Page LM0084* 60⟩≡ (31 733)

```
NOLRREAD        =       170D                    #       LANDING RADAR           LR NO
                                                #       REPOSITIONING;
NOLRRBIT        =       BIT10                   #       BYPASS UPDATE

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

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

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

# BIT 6 FLAG 11 (L)(R12)
READLR          =       174D                    #       OK TO READ LR           DO NO
READLBIT        =       BIT6                    #       RANGE DATA              DATA

# BIT 5 FLAG 11 (L)(R12)
READVEL         =       175D                    #       OK TO READ LR           DO NO
READVBIT        =       BIT5                    #       VELOCITY DATA           VELOC

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

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

# BIT 2 FLAG 11 (L)(R12)
VFLSHFLG        =       178D                    #       LR VELOCITY FAIL        LR VE
                                                #       LAMP SHOULD BE          SHOUL
VFLSHBIT        =       BIT2                    #       FLASHING

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

61 ⟨*Page LM0085* 61⟩≡ (31 733)

```
 HFLSHFLG        =       179D                    #       LR ALTITUDE FAIL        LR ALTITUDE FAI
 HFLSHBIT        =       BIT1                    #       LAMP SHOULD BE          LAMP SHOULD NOT
                                                 #       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
 CDESBIT         =       BIT15                   #       NATE, LGC COMMANDS      ON WHEN ANTENNA
                                                 #       RR REGARDLESS OF        BEING DESIGNATE
                                                 #       LOCK-ON

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

 # BIT 13 FLAG 12
 RCDU0FLG        =       182D                    #       RR CDU'S BEING          RR CDU'S NOT BE
 RCDU0BIT        =       BIT13                   #       ZEROED                  ZEROED

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

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

 # BIT 10 FLAG 12
 DESIGFLG        =       185D                    #       RR DESIGNATE            RR DESIGNATE NO
 DESIGBIT        =       BIT10                   #       REQUESTED OR IN         REQUESTED OR IN
                                                 #       PROGRESS                PROGRESS

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

62 ⟨*Page LM0086* 62⟩≡ (31 733)

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

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

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

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

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

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

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

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

DAPBOOLS        EQUALS  FLGWRD13                # DIGITAL AUTOPILOT FLAGWORD
```

63 ⟨*Page LM0087* 63⟩≡ (31 733)

```
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
                                                #       "ATT HOLD" (V76)        (V77)

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

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

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

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

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

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

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

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

64a ⟨*Page LM0088* 64a⟩≡ (31 733)
```
# BIT 6 FLAG 13
ULLAGFLG        =       204D                        #       ULLAGE REQUEST BY       NO IN
ULLAGER         =       BIT6                        #       MISSION PROGRAM         REQUE

# BIT 5 FLAG 13
AORBSFLG        =       205D                        #       P-AXIS COUPLES 7.15     P-AXI
AORBSYST        =       BIT5                        #       AND 8.16 PREFERRED      AND 3

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

# BIT 3 FLAG 13
ACCOKFLG        =       207D                        #       CONTROL AUTHORITY       RESTA
ACCSOKAY        =       BIT3                        #       VALUES FROM 1/ACCS      SINCE
                                                    #       USABLE                  OUTPU

# BIT 2 FLAG 13
AUTR2FLG        =       208D                        # THESE FLAGS ARE USED TOGETHER TO IN
AUTRATE2        =       BIT2                        # ASTRONAUT-CHOSEN KALCMANU MANEUVER
                                                    # (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
```

64b ⟨*Page LM0089* 64b⟩≡ (31 733)

1.4 rcs failure monitor

64c ⟨*rcs failure monitor* 64c⟩≡ (7)
⟨*Page LM0190* 65⟩
⟨*Page LM0191* 67⟩
⟨*Page LM0192* 69⟩

65 ⟨*Page LM0190* 65⟩≡ (64c 762)

```
# 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
# OF CHANNEL 32 TO SEE IF ANY ISOLATION-VALVE CLOSURE BITS HAVE APPEARED OR DISAPPEARED (THE CR
# FAILURES BY LAMPS LIT BY THE GRUMMAN FAILURE-DETECTION CIRCUITRY; THEY MAY RESPOND BY OPERATI
# ISOLATE PAIRS OF JETS FROM THE PROPELLANT TANKS AND SET BITS IN CHANNEL 32).  IN THE EVENT TH
# DIFFER FROM 'PVALVEST', THE RECORD OF ACTIONS TAKEN BY THIS ROUTINE, THE APPROPRIATE BITS IN
# 'CH6MASK', USED BY THE DAP JET-SELECTION LOGIC, ARE UPDATED, AS IS 'PVALVEST'.  TO SPEED UP &
# ROUTINE, NO MORE THAN ONE CHANGE IS ACCEPTED PER ENTRY.  THE HIGHEST-NUMBERED BIT IN CHANNEL
# ACTION IS THE ONE PROCESSED.
#
# THE CODING IN THE FAILURE MONITOR HAS BEEN WRITTEN SO AS TO HAVE ALMOST COMPLETE RESTART PROT
# EXAMPLE, NO ASSUMPTION IS MADE WHEN SETTING A 'CH5MASK' BIT TO 1 THAT THE PREVIOUS STATE IS C
# COURSE SHOULD BE.  ONE CASE WHICH MAY BE SEEN TO EVADE PROTECTION IS THE OCCURRENCE OF A REST
# ONE OR BOTH DAP MASK-WORDS BUT BEFORE UPDATING 'PVALVEST', COUPLED WITH A CHANGE IN THE VALVE
# FORMER STATE.  THE CONSEQUENCE OF THIS IS THAT THE NEXT ENTRY WOULD NOT SEE THE CHANGE INCOMP
# ORATED BY THE LAST PASS (BECAUSE IT WENT AWAY AT JUST THE RIGHT TIME), BUT THE DAP MASK-WORDS
# THIS COMBINATION OF EVENTS SEEMS QUITE REMOTE, BUT NOT IMPOSSIBLE UNLESS THE CREW OPERATES TH
# SECOND INTERVALS OR LONGER.  IN ANY EVENT, A DISAGREEMENT BETWEEN REALITY AND THE DAP MASKS W
# 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 FORM
#               PVALTEST UPDATED (1'S WHEN VALVE CLOSURES HAVE BEEN TRANSLATED INTO CH5MASK & C
#               JOB TO DO 1/ACCS.
#
# DEBRIS:  A, L, AND Q AND DEBRIS OF NOVAC.
#
# SUBROUTINE CALLED:  NOVAC.
```

```
EBANK=  CH5MASK

BANK    23
SETLOC  RCSMONT
BANK
```

67 ⟨*Page LM0191* 67⟩≡ (64c 762)

```
                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 (CH32)).
                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.
```

69 ⟨*Page LM0192* 69⟩≡ (64c 762)

```
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
```

1.5 ags initialization

70 ⟨*ags initialization* 70⟩≡ (7)
  ⟨*Page LM0206* 71⟩
  ⟨*Page LM0207* 73⟩
  ⟨*Page LM0208* 75⟩
  ⟨*Page LM0209* 77⟩
  ⟨*Page LM0210* 78a⟩

71 ⟨*Page LM0206* 71⟩≡ (70 713)

```
# 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 ST
#                (POSITION,VELOCITY,TIME) IN LEM IMU COORDINATES BY MEANS OF THE LGC DIGITAL DOW
#
#                (2) TO ZERO THE ICDU, LGC, AND AEA GIMBAL ANGLE COUNTER SIMULTANEOUSLY IN ORDER
#                COMMON ZERO REFERENCE FOR THE MEASUREMENT OF GIMBAL (EULER) ANGLES WHICH DEFINE
#
#                (3) TO ESTABLISH THE GROUND ELAPSED TIME OF AEA CLOCK ZERO.  (IF AN AEA CLOCK Z
#                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         (140)   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
```

73 ⟨*Page LM0207* 73⟩≡ (70 713)

```
                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
                        SCALEVEC                        # PROVIDE PROPER SCALING
                STODL   AGSBUFF                         # (LEMPREC AND CSMPREC LEAVE TDEC1 IN T
```

```
                TAT                                     # TAT = TIME TO WHICH RATT1 A
        STCALL  TDEC1                                   # COMPUTED (CSEC SINCE CLOCK
                CSMPREC                                 # CALCULATE CSM STATE VECTOR
        CALL
                SCALEVEC
```

75 ⟨*Page LM0208* 75⟩≡ (70 713)

```
                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
```

77 ⟨*Page LM0209* 77⟩≡ (70 713)

```
                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 VECTO
                        AGSRND1                 # ARE ADDED TO DEFEAT ALSIGNAG IN THE
                VAD     RTB                     # CASE OF A HIGH-ORDER ZERO COUPLED WIT
                        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
S
```

78a ⟨*Page LM0210* 78a⟩≡ (70 713)

```
                  2OCT      0000037777

                  SBANK=    LOWSUPER                    # FOR SUBSEQUENT LOW 2CADRS.
```

1.6 aotmark routine

78b ⟨*aotmark routine* 78b⟩≡ (7)

⟨*Page LM0244* 79⟩
⟨*Page LM0245* 80⟩
⟨*Page LM0246* 81⟩
⟨*Page LM0247* 82⟩
⟨*Page LM0248* 83⟩
⟨*Page LM0249* 84⟩
⟨*Page LM0250* 86⟩
⟨*Page LM0251* 87⟩
⟨*Page LM0252* 88⟩
⟨*Page LM0253* 90⟩
⟨*Page LM0254* 92⟩
⟨*Page LM0255* 94⟩
⟨*Page LM0256* 95⟩
⟨*Page LM0257* 96⟩
⟨*Page LM0258* 97⟩
⟨*Page LM0259* 98⟩
⟨*Page LM0260* 99⟩
⟨*Page LM0261* 100⟩

79 ⟨*Page LM0244* 79⟩≡ (78b 716)

```
                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      P00DOO
                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
```

80 ⟨*Page LM0245* 80⟩≡ (78b 716)

```
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
```

81 ⟨*Page LM0246* 81⟩≡ (78b 716)

```
 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
```

82 ⟨*Page LM0247* 82⟩≡ (78b 716)

```
                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
```

83 ⟨*Page LM0248* 83⟩≡ (78b 716)

```
# 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
```

84 ⟨*Page LM0249* 84⟩≡ (78b 716)

```
# 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
```

86 ⟨*Page LM0250* 86⟩≡ (78b 716)

```
# 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
```

87 ⟨*Page LM0251* 87⟩≡ (78b 716)

```
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
```

88 ⟨*Page LM0252* 88⟩≡ (78b 716)

```
# 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
```

90 ⟨*Page LM0253* 90⟩≡ (78b 716)

```
                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
```

92 ⟨*Page LM0254* 92⟩≡ (78b 716)

```
# 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
```

94 ⟨*Page LM0255* 94⟩≡ (78b 716)

```
                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
```

95 ⟨*Page LM0256* 95⟩≡ (78b 716)

```
 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
```

96 ⟨*Page LM0257* 96⟩≡ (78b 716)

```
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
```

97 ⟨*Page LM0258* 97⟩≡ (78b 716)

```
# 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
```

98 ⟨*Page LM0259* 98⟩≡ (78b 716)

```
                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
```

99 ⟨*Page LM0260* 99⟩≡ (78b 716)

```
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
```

100 ⟨*Page LM0261* 100⟩≡ (78b 716)

```
# 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
```

1.7 lem geometry

101 ⟨*lem geometry* 101⟩≡ (7)
⟨*page LM0320* 102⟩
⟨*page LM0321* 103⟩
⟨*page LM0322* 104⟩
⟨*page LM0323* 105⟩
⟨*page LM0324* 106⟩
⟨*page LM0325* 107a⟩

102 ⟨page LM0320 102⟩≡ (101 745)

```
                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 DEVIATIONS R,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
```

103 ⟨*page LM0321* 103⟩≡ (101 745)

```
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
```

104 ⟨*page LM0322* 104⟩≡ (101 745)

```
# THE FOLLOWING ROUTINE TAKES A HALF UNIT TARGET VECTOR REFERRED TO NAV BASE COORDINA
# GIMBAL ORIENTATIONS AT WHICH THE RR MIGHT SIGHT THE TARGET.  THE GIMBAL ANGLES CORR
# ARE LEFT IN MODEA AND THOSE WHICH WOULD BE USED AFTER A REMODE IN MODEB.  THIS ROUT
# ANGLE LESS THAN 90 DEGS IN ABS VALUE WITH ARBITRARY SHAFT, WITH A CORRESPONDING DEF
# SELECTION AND LIMIT CHECKING ARE DONE ELSEWHERE.
#
# THE MODE 1 CONFIGURATION IS CALCULATED FROM THE VECTOR AND THEN MODE 2 IS FOUND USI
#
#       S(2) = 180 + S(1)
#       T(2) = 180 - T(1)
#
# THE VECTOR ARRIVES IN MPAC WHERE TRG*SMNG 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
```

105 ⟨*page LM0323* 105⟩≡ (101 745)

```
        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
```

106 ⟨*page LM0324* 106⟩≡ (101 745)

```
# 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 RESP

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
```

107a ⟨*page LM0325* 107a⟩≡ (101 745)

```
  # (This page has nothing on it.)
```

1.8 r63 routine

107b ⟨*r63 routine* 107b⟩≡ (7)

⟨*Page LM0338* 108⟩
⟨*Page LM0339* 110⟩
⟨*Page LM0340* 111a⟩
⟨*Page LM0341* 111b⟩

108 ⟨*Page LM0338* 108⟩≡ (107b 760)

```
# 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 3AXISFLAG AND CALL R60LEM FOR ATTITUDE MANEUVER.
#
# CALLING SEQUENCE:     V 89 E.
#
# SUBROUTINES CALLED:   CHECKP00H, 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, PCINTVSM, SCAXIS, CPHI, CTHETA, CPSI,
```

110 ⟨*Page LM0339* 110⟩≡ (107b 760)

```
 #                3AXISFLAG.

                  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)
```

111a ⟨*Page LM0340* 111a⟩≡ (107b 760)

```
V89CALL1        STCALL  SCAXIS          # STORE SELECTED ALIGNMENT AXIS
                        VECPOINT        # PUTS DESIRED GIM ANG (OG,IG,MG) IN TMPAC
                STORE   CPHI            # STOR 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
```

111b ⟨*Page LM0341* 111b⟩≡ (107b 760)

```
DP1MIN          2DEC    6000
```

1.9 attitude maneuver routine

112 ⟨*attitude maneuver routine* 112⟩≡ (7)
⟨*Page LM0342* 113⟩
⟨*Page LM0343* 115⟩
⟨*Page LM0344* 117⟩
⟨*Page LM0345* 119⟩
⟨*Page LM0346* 121⟩
⟨*Page LM0347* 123⟩
⟨*Page LM0348* 125⟩
⟨*Page LM0349* 126⟩
⟨*Page LM0350* 127⟩
⟨*Page LM0351* 128⟩
⟨*Page LM0352* 130⟩
⟨*Page LM0353* 132⟩
⟨*Page LM0354* 134⟩
⟨*Page LM0355* 135⟩
⟨*Page LM0356* 136⟩
⟨*Page LM0357* 138⟩
⟨*Page LM0358* 140⟩
⟨*Page LM0359* 142⟩
⟨*Page LM0360* 144⟩
⟨*Page LM0361* 146⟩
⟨*Page LM0362* 148⟩
⟨*Page LM0363* 149a⟩

113 ⟨*Page LM0342* 113⟩≡ (112 720)

```
# 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 S
# DURING FREE FALL.  IT IS DESIGNED TO MANEUVER THE SPACECRAFT FROM ITS INITIAL ORIENTATION TO
# ORIENTATION SPECIFIED BY THE PROGRAM WHICH CALLS KALCMANU, AVOIDING GIMBAL LOCK IN THE PROCES
# MOD 2 VERSION, THIS DESIRED ATTITUDE IS SPECIFIED BY A SET OF OF THREE COMMANDED CDU ANGLES S
# 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 VE
# 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 (CO
# MAGNITUDE OF THE ROTATION (AM) TO BRING THE S/C FROM ITS INITIAL ORIENTATION TO ITS FINAL ORI
# THIS DIRECTION REMAINS FIXED BOTH IN INERTIAL COORDINATES AND IN COMMANDED S/C AXES THROUGHOU
#                  -
# MANEUVER.  ONCE COF AND AM HAVE BEEN DETERMINED, KALCMANU THEN EXAMINES THE MANEUVER TO SEE I
#                                     -
# THE S/C THROUGH GIMBAL LOCK.  IF SO, COF AND AM ARE READJUSTED SO THAT THE S/C WILL JUST SKIM
# LOCK ZONE AND ALIGN THE X-AXIS.  IN GENERAL A FINAL YAW ABOUT X WILL BE NECESSARY TO COMPLETE
# 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 I
# KALCMANU DOES THIS BY GENERATING A REFERENCE OF DESIRED GIMBAL ANGLES (CDUXD, CDUYD, CDUZD) W
# EVERY ONE SECOND DURING THE MANEUVER.  TO ACHIEVE A SMOOTHER SEQUENCE OF COMMANDS BETWEEN SUC
# THE PROGRAM ALSO GENERATES A SET OF INCREMENTAL CDU ANGLES (DELDCDU) TO BE ADDED TO CDU DESIR
# AUTOPILOT.  KALCMANU ALSO CALCULATES THE COMPONENT MANEUVER RATES (OMEGAPD, OMEGAQD, OMEGARD)
#                                    -
# BE DETERMINED SIMPLY BY MULTIPLYING COF BY SOME SCALAR (ARATE) CORRESPONDING TO THE DESIRED R
#
# AUTOMATIC MANEUVERS ARE TIMED WTH THE HELP OF WAITLIST SO THAT AFTER A SPECIFIED INTERVAL THE
# DESIRED RATES ARE SET TO ZERO AND THE DESIRED CDU ANGLES (CDUYD, CDUZD) ARE SET EQUAL TO THE
# ANGLES (CTHETA, CPSI).  IF ANY YAW REMAINS DUE TO GIMBAL LOCK AVOIDANCE, THE FINAL YAW MANEUV
# CALCULATED AND THE DESIRED YAW RATE SET TO SOME FIXED VALUE (ROLLRATE = + OR - 2 DEGREES PER
# IN THIS CASE ONLY AN INCREMENTAL CDUX ANGLE (DELFROLL) IS SUPPLIED TO THE DAP.  AT THE END OF
# MANEUVER OR IN THE EVENT THAT THERE WAS NO FINAL YAW, CDUXD IS SET EQUAL TO CPHI AND THE X-AX
# RATE SET TO ZERO.  THUS, UPON COMPLETION OF THE MANEUVER THE S/C WILL FINISH UP IN A LIMIT CY
# DESIRED GIMBAL ANGLES.
```

```
#
# PROGRAM LOGIC FLOW
#
# KALCMANU IS CALLED AS A HIGH PRIORITY JOB WITH ENTRY POINTS AT KALCMAN3 AND VECPOIN
# UP THE CURRENT CDU ANGLES TO BE USED AS THE BASIS FOR ALL COMPUTATIONS INVOLVING TH
```

115 ⟨*Page LM0343* 115⟩≡ (112 720)

```
# IT THEN DETERMINES THE DIRECTION COSINE MATRICES RELATING BOTH THE INITIAL AND FINAL S/C ORIE
#                *   *
# MEMBER AXES (MIS,MFS).  IT ALSO COMPUTES THE MATRIX RELATING FINAL S/C AXES TO INITIAL S/C AX
# 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
# 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
#   _                                                          *
# (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 (MFISY
#               _
# TO DETERMINE COF.
#
# THE PROGRAM THEN CHECKS TO SEE IF THE MANEUVER AS COMPUTED WILL BRING THE S/C THROUGH GIMBAL
# SO, A NEW MANEUVER IS CALCULATED WHICH WILL JUST SKIM THE GIMBAL LOCK ZONE AND ALIGN THE S/C
# METHOD ASSURES THAT THE ADDITIONAL MANEUVERING TO AVOID GIMBAL LOCK WILL BE KEPT TO A MINIMUM
# P AXIS YAW WILL BE NECESSARY, A SWITCH IS RESET (STATE SWITCH 31) TO ALLOW FOR THE COMPUTATIO
# YAW.
#
# AS STATED PREVIOUSLY, KALCMANU GENERATES A SEQUENCE OF DESIRED GIMBAL ANGLES WHICH ARE UPDATE
#
# SECOND.  THIS IS ACCOMPLISHED BY A SMALL ROTATION OF THE DESIRED S/C FRAME ABOUT THE VECTOR C
# 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
#      *
# FROM MIS.
#
# AT THE BEGINNING OF THE MANEUVER THE AUTOPILOT DESIRED RATES (OMEGAPD, OMEGAQD, OMEGARD) AND
# MANEUVER TIMINGS ARE ESTABLISHED.  ON THE FIRST PASS AND ON ALL SUBSEQUENT UPDATES THE CDU DE
# ANGLES ARE LOADED WITH THE APPROPRIATE VALUES AND THE INCREMENTAL CDU ANGLES ARE COMPUTED.  T
# (TIME1 AND TIME2) ARE THEN CHECKED TO SEE IF THE MANEUVER WILL TERMINATE BEFORE THE NEXT UPDA
# NOT, KALCMANU CALLS FOR ANOTHER UPDATE (RUN AS A JOB WITH PRIORITY TBD) IN ONE SECOND.  ANY D
# 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 (
```

```
# LIST TASK) TO STOP THE MANEUVER AT THE APPROPRIATE TIME AS EXPLAINED ABOVE.
```

117 ⟨*Page LM0344* 117⟩≡ (112 720)
```
# CALLING SEQUENCE
#
# IN ORDER TO PERFORM A KALCMANU SUPERVISED MANEUVER, THE COMMANDED GIMBAL ANGLES MUST BE PRECO
# STORED IN LOCATIONS CPHI, CTHETA, CPSI.  THE USER'S PROGRAM MUST THEN CLEAR STATE SWITCH NO 3
# ATTITUDE MANEUVER ROUTINE TO PERFORM ANY FINAL P-AXIS YAW INCURRED BY AVOIDING GIMBAL LOCK.
# 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 U
# 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 COMP
# SUCCESSFULLY, OR AT L+2 IF THE MANEUVER WAS ABORTED.  THIS ABORT WOULD OCCUR IF THE INITIAL O
# WAS IN GIMBAL LOCK.
#
# *** NOTA BENE ***  IF IT IS ASSUMED THAT THE DESIRED MANEUVERING RATE (0.5, 2, 5, 10 DEG/SEC)
# 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 DUR
# MANEUVER, KALCMANU WILL TERMINATE VIA GOODEND WITHIN 1 SECOND SO THAT R60 MAY REQUEST A TRIM
# SUBROUTINES.
#
# KALCMANU USES A NUMBER OF INTERPRETIVE SUBROUTINES WHICH MAY BE OF GENERAL INTEREST.  SINCE T
# WERE PROGRAMMED EXCLUSIVELY FOR KALCMANU, THEY ARE NOT, AS YET, GENERALLY AVAILABLE FOR USE B
#
# MXM3
# ----
#
# THIS SUBROUTINE MULTIPLIES TWO 3X3 MATRICES AND LEAVES THE RESULT IN THE FIRST 18 LOCATIONS O
# 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 ]
```

119 ⟨*Page LM0345* 119⟩≡ (112 720)

```
#                                                                              *
# INDEX REGISTER X1 MUST BE LOADED WITH THE COMPLEMENT OF THE STARTING ADDRESS FOR M1, AND X2 M
#                                                                  *
# LOADED WITH THE COMPLEMENT OF THE STARTING ADDRESS FOR M2.  THE ROUTINE USES THE FIRST 20 LOC
# 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 TH
#                       *
# SEQUENT COMPONENTS OF M.  THIS SUBROUTINE ALSO USES THE FIRST 20 LOCATIONS OF THE PUSH DOWN L
#
# CDU TO DCM
# ----------
#
# THIS SUBROUTINE CONVERTS THREE CDU ANGLES IN T(MPAC) TO A DIRECTION COSINE MATRIX (SCALED BY
# THE CORRESPONDING S/C ORIENTATIONS TO THE STABLE MEMBER FRAME.  THE FORMULAS FOR THIS CONVERS
#
#       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
```

121 ⟨*Page LM0346* 121⟩≡ (112 720)

```
#       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 SAM
#     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
#
# PROGRAM.  TO DO THIS THE CALLING PROGRAM MUST FIRST LOAD X2 WITH THE COMPLEMENT OF THE STARTI
#
# INTERNALLY, THE ROUTINE USES THE FIRST 16 LOCATIONS OF THE PUSH DOWN LIST, ALSO STEP REGISTER
# REGISTER X2.
#
# DCM TO CDU
# ----------
#                                                                            *
# THIS ROUTINE EXTRACTS THE CDU ANGLES FROM A DIRECTION COSINE MATRIX (M SCALED BY 2) RELATING
#                                                                                       *
# STABLE MEMBER AXES.  X1 MUST CONTAIN THE COMPLEMENT OF THE STARTING ADDRESS FOR M.  THE SUBRO
# CORRESPONDING GIMBAL ANGLES IN V(MPAC) AS DOUBLE PRECISION 1'S COMPLEMENT ANGLES SCALED BY 2P
# 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
```

123 ⟨*Page LM0347* 123⟩≡ (112 720)

```
#       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 DO
# RETURNS THE POINTER TO ITS ORIGINAL SETTING.  THIS PROCEDURE ALLOWS THE CALLER TO STORE THE M
# 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
#
#        *              *        _ _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                    ]
 #
```

125 ⟨*Page LM0348* 125⟩≡ (112 720)

```
#       -
#       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
#     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
#                                                                                              -
# THE LOCATION CALLED DEL.  IN ORDER TO USE THE ROUTINE, THE CALLING PROGRAM MUST FIRST STORE U
# DOUBLE PRECISION VECTOR) IN THE SET OF ERASABLE LOCATIONS BEGINNING WITH THE ADDRESS CALLED C
# 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 O
#
# PROGRAM STORAGE ALLOCATION
#
#       1)      FIXED MEMORY            1059 WORDS
#       2)      ERASABLE MEMORY           98
#       3)      STATE SWITCHES             3
```

126 ⟨*Page LM0349* 126⟩≡ (112 720)

```
#       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
#                                       1               MANEUVER DID NOT GO THROUGH G
#         *
#       32              13              0               CONTINUE UPDATE PROCESS
#                                       1               START UPDATE PROCESS
#
#       33              12              0               PERFORM FINAL P AXIS YAW IF R
#                                       1               IGNORE ANY FINAL P-AXIS YAW
#
#       34              11              0               SIGNAL END OF KALCMANU
#                                       1               KALCMANU IN PROCESS.    USER
#
#       * 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
```

127 ⟨*Page LM0350* 127⟩≡ (112 720)

```
 #        TRANSPGS
 #        SIGNMPAC
 #        READCDUK
 #        CDUTODCM
```

128 ⟨*Page LM0351* 128⟩≡ (112 720)

```
                BANK    15
                SETLOC  KALCMON1
                BANK

                EBANK=  BCDU

# THE THREE DESIRED CDU ANGLES MUST BE STORED AS SINGLE PRECISION TWO'S COMPLEMENT AN
# 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
```

130 ⟨*Page LM0352* 130⟩≡ (112 720)

```
                        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
```

132 ⟨*Page LM0353* 132⟩≡ (112 720)

```
                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
```

134 ⟨*Page LM0354* 134⟩≡ (112 720)

```
                        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
```

135 ⟨*Page LM0355* 135⟩≡ (112 720)

```
                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
```

136 ⟨*Page LM0356* 136⟩≡ (112 720)

```
# 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
```

138 ⟨*Page LM0357* 138⟩≡ (112 720)

```
                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
```

140 ⟨*Page LM0358* 140⟩≡ (112 720)

```
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)
```

142 ⟨*Page LM0359* 142⟩≡ (112 720)

```
                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 U 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 U 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
```

144 ⟨*Page LM0360* 144⟩≡ (112 720)

```
                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)
```

146 ⟨*Page LM0361* 146⟩≡ (112 720)

```
                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
#
#       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
```

148 ⟨*Page LM0362* 148⟩≡ (112 720)

```
#
# 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
```

149a ⟨*Page LM0363* 149a⟩≡ (112 720)

```
# 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
```

1.10 imu performance test 2

149b ⟨*imu performance test 2* 149b⟩≡ (7)

⟨*Page LM0373* 150⟩
⟨*Page LM0374* 152⟩
⟨*Page LM0375* 154⟩
⟨*Page LM0376* 156⟩
⟨*Page LM0377* 157⟩
⟨*Page LM0378* 158⟩
⟨*Page LM0379* 159⟩
⟨*Page LM0380* 160⟩
⟨*Page LM0381* 162⟩

150 ⟨*Page LM0373* 150⟩≡ (149b 735)

```
# 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
```

152 ⟨*Page LM0374* 152⟩≡ (149b 735)

```
                  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
```

154 ⟨*Page LM0375* 154⟩≡ (149b 735)

```
                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
```

156 ⟨*Page LM0376* 156⟩≡ (149b 735)

```
                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*
```

157 ⟨*Page LM0377* 157⟩≡ (149b 735)

```
                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
```

158 ⟨*Page LM0378* 158⟩≡ (149b 735)

```
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
```

159 ⟨*Page LM0379* 159⟩≡ (149b 735)

```
                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
```

160 ⟨*Page LM0380* 160⟩≡ (149b 735)

```
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
```

162 ⟨*Page LM0381* 162⟩≡ (149b 735)

```
                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
```

1.11 imu performance tests 4

163a ⟨*imu performance tests 4* 163a⟩≡ (7)
⟨*Page LM0382* 163b⟩
⟨*Page LM0383* 164⟩
⟨*Page LM0384* 165⟩
⟨*Page LM0385* 166⟩
⟨*Page LM0386* 168⟩
⟨*Page LM0387* 170⟩
⟨*Page LM0388* 172⟩
⟨*Page LM0389* 173a⟩

163b ⟨*Page LM0382* 163b⟩≡ (163a 737)

```
# 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
```

164 ⟨*Page LM0383* 164⟩≡ (163a 737)

```
                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
```

165 ⟨*Page LM0384* 165⟩≡ (163a 737)

```
 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
```

166 ⟨*Page LM0385* 166⟩≡ (163a 737)

```
 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
```

168 ⟨*Page LM0386* 168⟩≡ (163a 737)

```
                        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
```

170 ⟨*Page LM0387* 170⟩≡ (163a 737)

```
                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
```

172 ⟨*Page LM0388* 172⟩≡ (163a 737)

```
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
```

173a ⟨*Page LM0389* 173a⟩≡ (163a 737)

```
                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
```

1.12 s band antenna for lm

173b ⟨*s band antenna for lm* 173b⟩≡ (7)
⟨*Page LM0486* 174⟩
⟨*Page LM0487* 176⟩
⟨*Page LM0488* 178⟩
⟨*Page LM0489* 179a⟩

174 ⟨*Page LM0486* 174⟩≡ (173b 764)

```
# 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
#       BLANKRET
#
# 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
```

176 ⟨*Page LM0487* 176⟩≡ (173b 764)

```
                CADR    R02BOTH         # CHECK IF IMU IS ON AND ALIGNED
                TC      INTPRET
                SETPD   RTB
                        0D
                        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
                        0D              # 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(B0)
                        RLM     +2
                PUSH    DSU
                        RLM
                DMP
                        1OVSQRT2
                STODL   RLM     +2
                DAD     DMP
```

```
                RLM
                1OVSQRT2
        STOVL   RLM             # R B-1
                RLM
        UNIT    PDVL
```

178 ⟨*Page LM0488* 178⟩≡ (173b 764)

```
                        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 B0
            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
```

179a ⟨*Page LM0489* 179a⟩≡ (173b 764)

```
                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 ***
```

1.13 radar leadin routines

179b ⟨*radar leadin routines* 179b⟩≡ (7)

⟨*Page LM0490* 180⟩
⟨*Page LM0491* 181a⟩

180 ⟨*Page LM0490* 180⟩≡ (179b 761)

```
                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
```

181a ⟨*Page LM0491* 181a⟩≡ (179b 761)

```
                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
```

1.14 p30-p37 routines

181b ⟨*p30-p37 routines* 181b⟩≡ (7)

⟨*Page LM0614* 182⟩
⟨*Page LM0615* 183⟩
⟨*Page LM0616* 184⟩
⟨*Page LM0617* 186⟩

182 ⟨*Page LM0614* 182⟩≡ (181b 750)

```
# PROGRAM DESCRIPTION P30       DATE 3-6-67
#
# MOD.1 BY RAMA AIYAWAR
#
# FUNCTIONAL DESCRIPTIONS
#       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          # RETURN 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
```

183 ⟨*Page LM0615* 183⟩≡ (181b 750)

```
                TC      INTPRET
                SETGO
                        XDELVFLG        # FOR P40'S: EXTERNAL DELTA-V GUIDANCE.
                        REVN1645        # TRKMKCNT, T60, +MGA  DISPLAY

V06N33          VN      0633
V06N42          VN      0642
```

184 ⟨*Page LM0616* 184⟩≡ (181b 750)

```
# 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
```

186 ⟨*Page LM0617* 186⟩≡ (181b 750)

```
                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
```

1.15 p32-p35 p72-p75 routines

187 ⟨*p32-p35 p72-p75 routines* 187⟩≡ (7)
⟨*Page LM0618* 188⟩
⟨*Page LM0619* 190⟩
⟨*Page LM0620* 192⟩
⟨*Page LM0621* 194⟩
⟨*Page LM0622* 196⟩
⟨*Page LM0623* 197⟩
⟨*Page LM0624* 198⟩
⟨*Page LM0625* 200⟩
⟨*Page LM0626* 202⟩
⟨*Page LM0627* 204⟩
⟨*Page LM0628* 206⟩
⟨*Page LM0629* 207⟩
⟨*Page LM0630* 208⟩
⟨*Page LM0631* 209⟩
⟨*Page LM0632* 210⟩
⟨*Page LM0633* 212⟩
⟨*Page LM0634* 214⟩
⟨*Page LM0635* 216⟩
⟨*Page LM0636* 218⟩
⟨*Page LM0637* 220⟩
⟨*Page LM0638* 222⟩
⟨*Page LM0639* 224⟩
⟨*Page LM0640* 226⟩
⟨*Page LM0641* 228⟩
⟨*Page LM0642* 229⟩
⟨*Page LM0643* 230a⟩
⟨*Page LM0644* 230b⟩
⟨*Page LM0645* 231a⟩
⟨*Page LM0646* 231b⟩
⟨*Page LM0647* 232⟩
⟨*Page LM0648* 234⟩
⟨*Page LM0649* 235a⟩
⟨*Page LM0650* 235b⟩

188 ⟨*Page LM0618* 188⟩≡ (187 751)

```
# 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
```

190 ⟨*Page LM0619* 190⟩≡ (187 751)

```
#               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
```

192 ⟨*Page LM0620* 192⟩≡ (187 751)

```
#       (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
```

194 ⟨*Page LM0621* 194⟩≡ (187 751)

```
#       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
```

196 ⟨*Page LM0622* 196⟩≡ (187 751)

```
                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
```

197 ⟨*Page LM0623* 197⟩≡ (187 751)

```
                        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
```

198 ⟨*Page LM0624* 198⟩≡ (187 751)

```
# 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
```

200 ⟨*Page LM0625* 200⟩≡ (187 751)

```
#                       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
```

202 ⟨*Page LM0626* 202⟩≡ (187 751)
```
#       (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
#       VNP00H
#       SELECTMU
#       ADVANCE
#       CDHMVR
#       INTINT3P
#       ACTIVE
#       PASSIVE
```

```
#       S33/S34.1
#       ALARM
#       BANKCALL
#       GOFLASH
#       GOTOPOOH
#       S32/33.1
```

204 ⟨*Page LM0627* 204⟩≡ (187 751)

```
 #        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
                        0D
                        VACT3
                PDVL    CALL
                        RACT2
                        INTINT3P
                CALL
                        ACTIVE
                SETPD   VLOAD
                        0D
                        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      GOTOP00H
        TC      +2
```

206 ⟨*Page LM0628* 206⟩≡ (187 751)

```
                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
```

207 ⟨*Page LM0629* 207⟩≡ (187 751)

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

208 ⟨*Page LM0630* 208⟩≡ (187 751)

```
# ***** DISDVLVC *****
#
# SUBROUTINES USED
#
#       S32/33.X
#       VNP00H

DISDVLVC        STORE   DELVLVC
                STQ     CALL
                        NORMEX
                        S32/33.X
                VLOAD   MXV
                        DELVLVC
                        0D
                VSL1    SXA,1
                        VERBNOUN
                STORE   DELVLVC
                EXIT
                CA      VERBNOUN
                TC      VNP00H
                TC      INTPRET
                GOTO
                        NORMEX
```

209 ⟨*Page LM0631* 209⟩≡ (187 751)

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

210 ⟨*Page LM0632* 210⟩≡ (187 751)

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

212 ⟨*Page LM0633* 212⟩≡ (187 751)

```
                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
```

214 ⟨*Page LM0634* 214⟩≡ (187 751)

```
                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
```

216 ⟨*Page LM0635* 216⟩≡ (187 751)

```
                        CIRCL
                DLOAD
                        ECC
                DSU     BMN
                        ONETHTH
                        CIRCL
                DLOAD   CALL
                        R1
                        SHIFTR1
                SETPD   NORM
                        2D
                        X1
                PDVL    DOT             #                                       PL04I
                        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 PL02I
                PDDL    DMP
                        P
                        R1
                CALL
                        SHIFTR1
                SL4     SL1
                SQRT    DMP             # ((P/MU)**.5)/R1       B14+B-14 = B-31 PL02I
                BOFF    SL3
                        CMOONFLG
                        CSI/B3
CSI/B3          PDVL    DOT
                        RACT1
                        VACT4
                STORE   RDOTV
                ABS
                NORM    DMP             # ((P/MU)**.5)RDOTV/R1                  PL02I
                        X2
```

```
         XSU,1   SL*             #                      B-31+B36-B3 = B2
                 X2
                 3,1
         STODL   12D
                 P30ZERO
```

218 ⟨*Page LM0636* 218⟩≡ (187 751)

```
                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
```

220 ⟨*Page LM0637* 220⟩≡ (187 751)

```
                0D
                RPASS1
        CALL
                INTINT2C
        STOVL   RPASS2
                VATT
        STCALL  VPASS2
                CDHMVR
        VLOAD   SETPD
                RACT2
                0D
        PDVL    CALL
                VACT3
                PERIAPO1
        CALL
                SHIFTR1
        STOVL   POSTCDH
                VACT3
        SETPD   PDVL
                0D
                RACT2
        PDDL    PDDL
                TCDH
                TTPI
        PDDL    PUSH
                TWOPI
        CALL
                INTINT
        CALL
                ACTIVE
        DLOAD
                ELEV
        SETPD   SINE
                6D
        PDVL    UNIT
                RACT3
        STORE   00D             # URA3 AT 00D
        PDVL    VXV             # PL14D, PL08D
                UP1
        UNIT
        PDDL    COSINE          # UNIT(URA3 X UVA3 X URA3) = UH3        B1 PI
                ELEV
        VXSC    STADR           # (COSLOS)(UH3)                         B2 PI
        STORE   18D             #       PLUS
        DLOAD   VXSC            # (SINLOS)(URA3) = U                    B2 PI
```

```
        VAD     VSL1
                18D             #                                       B1
        PUSH    DOT             #                                          PL06D
                RACT3           # (U . RA3) = TEMP1           B1 +B29 = B30
        SL1     PUSH            #                                     B29 PL08D
```

222 ⟨*Page LM0638* 222⟩≡ (187 751)

```
                    DSQ      TLOAD           # TEMP1**2                                B58
                             MPAC
                    PDVL     DOT             #                                             PI
                             RACT3
                             RACT3
                    TLOAD    DCOMP           # RA3 . RA3
                             MPAC
                    PDVL     DOT             # RP3 . RP3                               B58 PI
                             RPASS3
                             RPASS3          #                                             PI
                    TAD      TAD             # TEMP1**2 + RA3.RA3 + RP3.RP3 = TEMP2        PI
                    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 PI
                    DCOMP    DSU
                             06D             # -TEMP1-TEMP3 = K2 AT 10D
                    STODL    10D             #                                             PI
                    DSU      STADR           #                                             PI
                    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
```

224 ⟨*Page LM0639* 224⟩≡ (187 751)

```
            VXV     PDVL            # UVP3 X URP3                              PI
                    06D
                    06D
            VXV     DOT
                    00D
            STADR                   #                                          PI
            STOVL   12D             # (URP3 X V).(UVP3 X URP3)=TEMP            PI
            DOT     SL1             #                                          PI
            ARCCOS  SIGN
                    12D             #                                       B0
            SR1     PUSH            # GAMMA = SIGN(TEMP)ARCOS(UNITV.URP3)      PI
            BON     DLOAD
                    S32.1F2
                    FRSTPAS
                    00D             # NOT THE FIRST PASS OF A CYCLE
            DSU     PDDL            # GAMMA-GAMPREV                         B1 PI
                    GAMPREV
                    DELVCSI
            DSU     NORM            #                                       B7
                    DVPREV
                    X1
            BDDV    PDDL            # (GAM-GAMPREV)/(DV-DVPREV)         B-6+N1 I
                    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
```

226 ⟨*Page LM0640* 226⟩≡ (187 751)

```
                  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          #                                        PI
                  PUSH    DSU
                          EPSILN1
                  BMN     DLOAD
                          CSI/SOL
                  DSU     BMN
                          DELMAX1
                          CSISTEP
                  DLOAD   SIGN
                          DELMAX1
                          DELDV
```

```
                STORE   DELDV
 CSISTEP        DLOAD   DSU
                        DELVCSI
                        DELDV
                STCALL  DELVCSI
```

228 ⟨*Page LM0641* 228⟩≡ (187 751)

```
                        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
```

229 ⟨*Page LM0642* 229⟩≡ (187 751)

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

230a ⟨*Page LM0643* 230a⟩≡ (187 751)

```
# ***** ROTATE *****

ROTATE          PUSH    PUSH
                DOT     VXSC
                        UP1
                        UP1
                VSL2    BVSU
                UNIT    PDVL
                ABVAL   VXSC
                VSL1    RVQ
```

230b ⟨*Page LM0644* 230b⟩≡ (187 751)

```
# ***** INTINTNA *****

INTINT2C        PDDL    PDDL
                        TCSI
                        TCDH
                PDDL    PUSH
                        TWOPI
                GOTO
                        INTINT
INTINT3P        PDDL    PDDL
                        TCDH
                        TTPI
                PDDL    PUSH
                        P30ZERO
                GOTO
                        INTINT
```

231a ⟨*Page LM0645* 231a⟩≡ (187 751)

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

231b ⟨*Page LM0646* 231b⟩≡ (187 751)

```
# ***** S32/33.X *****

S32/33.X        SETPD   VLOAD
                        6D
                        UP1
                VCOMP   PDVL
                        RACT1
                UNIT    VCOMP
                PUSH    VXV
                        UP1
                VSL1
                STORE   0D
                RVQ
```

232 ⟨*Page LM0647* 232⟩≡ (187 751)

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

234 ⟨*Page LM0648* 234⟩≡ (187 751)

```
                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
```

235a ⟨*Page LM0649* 235a⟩≡ (187 751)

```
                STCALL  DELVEET2        # DELTA VCDH -- REFERENCE COORDINATES
                        SUBEXIT
```

235b ⟨*Page LM0650* 235b⟩≡ (187 751)

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

1.16 lambert aimpoint guidance

236 ⟨*lambert aimpoint guidance* 236⟩≡ (7)
⟨*Page LM0651* 237⟩
⟨*Page LM0652* 239⟩
⟨*Page LM0653* 240⟩

237 ⟨*Page LM0651* 237⟩≡ (236 742)

```
# GENERAL LAMBERT AIMPOINT GUIDANCE **
# WRITTEN BY RAMA M AIYAWAR

# PROGRAM P-31 DESCRIPTION **
#
# 1.    TO ACCEPT TARGETING PARAMETERS OBTAINED FROM A SOURCE EXTERNAL
#       TO THE LEM AND COMPUTE THERE FROM THE REQUIRED-VELOCITY AND
#       OTHER INITIAL CONDITIONS REQUIRED BY LM FOR DESIRED MANEUVER.
#       THE TARGETING PARAMETERS ARE TIG (TIME OF IGNITION), TARGET
#       VECTOR (RTARG), AND THE TIME FROM TIG UNTIL THE TARGET IS
#       REACHED (DELLT4), DESIRED TIME OF FLIGHT FROM RINIT TO RTARG.

# ASSUMPTIONS **
#
# 1.    THE TARGET PARAMETERS MAY HAVE BEEN LOADED PRIOR TO THE
#       EXECUTION OF THIS PROGRAM.
# 2.    THIS PROGRAM IS APPLICABLE IN EITHER EARTH OR LUNAR ORBIT.
# 3.    THIS PROGRAM IS DESIGNED FOR ONE-MAN OPERATION, AND SHOULD
#       BE SELECTED BY THE ASTRONAUT BY DSKY ENTRY V37 E31.

# SUBROUTINES USED **
#
# MANUPARM, TTG/N35, R02BOTH, MIDGIM, DISPMGA, FLAGDOWN, BANKCALL,
# GOTOPOOH, ENDOFJOB, PHASCHNG, GOFLASHR, GOFLASH.
#
# MANUPARM      CALCULATES APOGEE, PERIGEE ALTITUDES AND DELTAV DESIRED
#               FOR THE MANEUVER.
#
# TTG/N35       CLOCKTASK - UPDATES CLOCK.
#
# MIDGIM        CALCULATES MIDDLE GIMBAL ANGLE FOR DISPLAY.
#
# R02BOTH       IMU - STATUS CHECK ROUTINE.

# DISPLAYS USED IN P-31LM **
#
# V06N33        DISPLAY SOTRED TIG (IN HRS. MINS. SECS.)
# V06N42        DISPLAY APOGEE, PERIGEE, DELTAV.
# V16N35        DISPLAY TIME FROM TIG.
# V06N45        TIME FROM IGNITION AND MIDDLE GIMBAL ANGLE.

# ERASABLE INITIALIZATION REQUIRED **
#
# TIG           TIME OF IGNITION                DP      (B+28) CS.
```

```
#
# DELLT4         DESIRED TIME OF FLIGHT          DP      (B+28) CS
#                FROM RINIT TO RTARG.
#
# RTARG          RADIUS VECTOR OF TARGET POSITION VECTOR
#                RADIUS VECTOR SCALED TO (B+29)METERS IF EARTH ORBIT
```

239 ⟨*Page LM0652* 239⟩≡ (236 742)

```
#               RADIUS VECTOR SCALED TO (B+27)METERS IF MOON ORBIT

# OUTPUT **
#
# HAPO          APOGEE ALTITUDE
# HPER          PERIGEE ALTITUDE
# VGDISP        MAG. OF DELTAV FOR DISPLAY, SCALING     B+7 M/CS EARTH
#               MAG. OF DELTAV FOR DISPLAY, SCALING     B+5 M/CS MOON
# MIDGIM        MIDDLE GIMBAL ANGLE
# XDELVFLG      RESETS XDELVFLG FOR LAMBERT VG COMPUTATIONS

# ALARMS OR ABORTS      NONE **

# RESTARTS ARE VIA GROUP 4 **

                SETLOC  GLM
                BANK

                EBANK=  SUBEXIT

                COUNT*  $$/P31
P31             TC      P20FLGON
                CAF     V06N33          # T16
                TC      VNP00H
                TC      INTPRET
                CLEAR   DLOAD
                        UPDATFLG
                        TIG
                STCALL  TDEC1           # INTEGRATE STATE VECTORS TO TIG
                        LEMPREC
                VLOAD   SETPD
                        RATT
                        0D
                STORE   RTIG
                STOVL   RINIT
                        VATT
                STORE   VTIG
                STODL   VINIT
                        P30ZERO
                PUSH    PDDL            # E4 AND NUMIT = 0
                        DELLT4
                DAD     SXA,1
                        TIG
                        RTX1
                STORE   TPASS4
```

```
                SXA,2   CALL
                        RTX2
                        INITVEL
                VLOAD   PUSH
```

240 ⟨*Page LM0653* 240⟩≡ (236 742)

```
                        DELVEET3
                STORE   DELVSIN
                ABVAL   CLEAR
                        XDELVFLG
                STCALL  VGDISP
                        GET.LVC
                VLOAD   PDVL
                        RTIG
                        VIPRIME
                CALL
                        PERIAPO1
                CALL
                        SHIFTR1
                CALL                    # LIMIT DISPLAY TO 9999.9 N. MI.
                        MAXCHK
                STODL   HPER
                        4D
                CALL
                        SHIFTR1
                CALL                    # LIMIT DISPLAY TO 9999.9 N. MI.
                        MAXCHK
                STORE   HAPO
                EXIT
                CAF     V06N81          # DELVLVC
                TC      VNPOOH
                CAF     V06N42          # HAPO, HPER, VGDISP
                TC      VNPOOH
                TC      INTPRET
REVN1645        SET     CALL            # TRKMKCNT, TTOGO, +MGA
                        FINALFLG
                        VN1645
                GOTO
                        REVN1645


# *** END OF LEMP30S .103 ***
```

1.17 burn-baby-burn master ignition routine

241 ⟨*burn-baby-burn master ignition routine* 241⟩≡ (7)
⟨*Page LM0731* 242⟩
⟨*Page LM0732* 244⟩
⟨*Page LM0733* 245⟩
⟨*Page LM0734* 246⟩
⟨*Page LM0735* 248⟩
⟨*Page LM0736* 250⟩
⟨*Page LM0737* 252⟩
⟨*Page LM0738* 254⟩
⟨*Page LM0739* 256⟩
⟨*Page LM0740* 258⟩
⟨*Page LM0741* 260⟩
⟨*Page LM0742* 262⟩
⟨*Page LM0743* 264⟩
⟨*Page LM0744* 266⟩
⟨*Page LM0745* 267⟩
⟨*Page LM0746* 269⟩
⟨*Page LM0747* 271⟩
⟨*Page LM0748* 273⟩
⟨*Page LM0749* 274⟩
⟨*Page LM0750* 275⟩
⟨*Page LM0751* 277⟩

HONI SOIT QUI MAL Y PENS:
"May he be shamed who thinks badly of it"

242 ⟨*Page LM0731* 242⟩≡ (241 722)

```
# 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
# IT PERFORMS ALL FUNCTIONS IMMEDIATELY ASSOCIATED WITH APS OR DPS IGNITION:  IN PART
# BETWEEN THE PRE-IGNITION TIME CHECK -- ARE WE WITHIN 45 SECONDS OF TIG? -- AND TIG
# PROGRAMS THROTTLE UP.
#
# VARIATIONS AMONG PROGRAMS ARE ACCOMODATED BY MEANS OF TABLES CONTAINING CONSTANTS (
# WAITLIST, FOR PINBALL) AND TCF INSTRUCTIONS.  USERS PLACE THE ADRES OF THE HEAD OF
# (OF P61TABLE FOR P61LM, FOR EXAMPLE) IN ERASABLE REGISTER 'WHICH' (E4).  THE IGNITI
# WHICH TO OBTAIN OR EXECUTE THE PROPER TABLE ENTRY.  THE IGNITION ROUTINE IS INITIAT
# THROUGH BANKJUMP IF NECESSARY.  THERE IS NO RETURN.
#
# THE MASTER IGNITION ROUTINE WAS CONCEIVED AND EXECUTED, AND (NOTA BENE) IS MAINTAIN
#
#                   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)
```

244 ⟨*Page LM0732* 244⟩≡ (241 722)

```
                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)
```

245 ⟨*Page LM0733* 245⟩≡ (241 722)

```
                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)
```

246 ⟨*Page LM0734* 246⟩≡ (241 722)

```
                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.
```

248 ⟨*Page LM0735* 248⟩≡ (241 722)

```
                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
```

250 ⟨*Page LM0736* 250⟩≡ (241 722)

```
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
```

252 ⟨*Page LM0737* 252⟩≡ (241 722)

```
                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     NEG0            # 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
```

254 ⟨*Page LM0738* 254⟩≡ (241 722)

```
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
```

256 ⟨*Page LM0739* 256⟩≡ (241 722)

```
                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
```

258 ⟨*Page LM0740* 258⟩≡ (241 722)

```
                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
```

260 ⟨*Page LM0741* 260⟩≡ (241 722)

```
                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
```

262 ⟨*Page LM0742* 262⟩≡ (241 722)

```
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
```

264 ⟨*Page LM0743* 264⟩≡ (241 722)

```
                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
```

266 ⟨*Page LM0744* 266⟩≡ (241 722)

```
                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     NEG0
                DXCH    -PHASE6
                TCF     TASKOVER

CLOKJOB         EXTEND
                DCS     TIG
                DXCH    TTOGO
                EXTEND
```

267 ⟨*Page LM0745* 267⟩≡ (241 722)

```
                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 P00H
                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
```

269 ⟨*Page LM0746* 269⟩≡ (241 722)

```
 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     GOTOP00H        # WAY TO P00 (GOTOP00H 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
```

271 ⟨*Page LM0747* 271⟩≡ (241 722)

```
                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
```

273 ⟨*Page LM0748* 273⟩≡ (241 722)

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

 P40ALM          TC      ALARM           # PROGRAM SELECTION NOT CONSISTENT WITH
                 OCT     1706            # VEHICLE CONFIGURATION

 REP40ALM        CAF     V05N09          # (14)
                 TC      BANKCALL
                 CADR    GOFLASH

                 TCF     GOTOP00H        # 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     GOTOP00H        # V34E TERMINATE
                 TCF     P40A/P          # RECYCLE
 GOBACK          CA      TEMPR60
```

```
                TC      BANKJUMP        # GOODBYE.  COME AGAIN SOON.

P40A/PMD        OCT     00203
```

274 ⟨*Page LM0749* 274⟩≡ (241 722)

```
                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
```

275 ⟨*Page LM0750* 275⟩≡ (241 722)

```
# 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
#       WHICH MERELY DOES A TC TASKOVER.  IF THE SAME TASK IS SCHEDULED MORE THAN ONCE, ONLY TH
#       FIRST IS REMOVED.  IF THE TASK IS NOT SCHEDULED, KILLTASK TAKES NO ACTION AND RETURNS W
#       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
```

277 ⟨*Page LM0751* 277⟩≡ (241 722)

```
                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
```

1.18 the lunar landing

278 ⟨*the lunar landing* 278⟩≡ (7)
⟨*Page LM0785* 279⟩
⟨*Page LM0786* 281⟩
⟨*Page LM0787* 283⟩
⟨*Page LM0788* 285⟩
⟨*Page LM0789* 287a⟩
⟨*Page LM0790* 287b⟩
⟨*Page LM0791* 288⟩
⟨*Page LM0792* 289a⟩

279 ⟨*Page LM0785* 279⟩≡ (278 770)

```
                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 FLPASS0
                TS      WCHPHASE

                CA      ZERO
                TS      FLPASS0

                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:
```

281 ⟨*Page LM0786* 281⟩≡ (278 770)

```
                        0               #       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) - VI
```

```
#                           2              1                  0
#       DDUM = ----------------------------------------------------------------------------
#                                                   10
#                                                 2   (VGU - 16 VGU KIGNX/B4)
#                                                         2         0
```

283 ⟨*Page LM0787* 283⟩≡ (278 770)

```
  # disconnected from their respective variables
  # THE NUMERATOR IS SCALED IN METERS AT 2(28).  THE DENOMINATOR IS A VELOCITY IN UNITS OF 2(10)
  # THE QUOTIENT IS THUS A TIME IN UNITS OF 2(18) CENTISECONDS.  THE FINAL SHIFT RESCALES TO UNIT
  # THERE IS NO DAMPING FACTOR.  THE CONSTANTS KIGNX/B4, KIGNY/B8 AND KIGNV/B4 ARE ALL NEGATIVE I

  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
```

285 ⟨*Page LM0788* 285⟩≡ (278 770)

```
                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
```

287a ⟨*Page LM0789* 287a⟩≡ (278 770)

```
                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     GOTOP00H        # 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
```

287b ⟨*Page LM0790* 287b⟩≡ (278 770)

```
#       --------------------------------
```

288 ⟨*Page LM0791* 288⟩≡ (278 770)

```
 #        *************************
 #        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     GOTOP00H        # TERMINATE
```

```
                TCF     +2              # PROCEED
                TCF     -5              # RECYCLE

                TC      INTPRET
```

289a ⟨*Page LM0792* 289a⟩≡ (278 770)

```
                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
```

1.19 throttle control routines

289b ⟨*throttle control routines* 289b⟩≡ (7)

⟨*Page LM0793* 290⟩
⟨*Page LM0794* 291⟩
⟨*Page LM0795* 293⟩
⟨*Page LM0796* 295⟩
⟨*Page LM0797* 296a⟩

290 ⟨*Page LM0793* 290⟩≡ (289b 771)
```
                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

                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,
# MINIMUM BY ASTRONAUT OR MISSION CONTROL PROGRAMS, PROVIDES THE LOWER BOUND.  A STOP
# PROVIDES THE UPPER.

WHERETO         CA      EBANK5          # INITIALIZE L*WCR*T AND H*GHCR*T FROM
                TS      EBANK           #       PAD LOADED ERASABLES IN W-MATRIX
```

291 ⟨*Page LM0794* 291⟩≡ (289b 771)

```
                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.
#
# NOTE2         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
```

293 ⟨*Page LM0795* 293⟩≡ (289b 771)

```
                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 PROCESS 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
```

295 ⟨*Page LM0796* 295⟩≡ (289b 771)

```
                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 PULS

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
```

296a ⟨*Page LM0797* 296a⟩≡ (289b 771)

```
# CONSTANTS --

FEXTRA          =       BIT13           # FEXT +5.13309020 E+4

/AF/CNST        DEC     .13107

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

1.20 lunar landing guidance equations

296b ⟨*lunar landing guidance equations* 296b⟩≡ (7)

⟨*Page LM0798* 297⟩
⟨*Page LM0799* 298⟩
⟨*Page LM0800* 300⟩
⟨*Page LM0801* 302⟩
⟨*Page LM0802* 304⟩
⟨*Page LM0803* 306⟩
⟨*Page LM0804* 307⟩
⟨*Page LM0805* 309⟩
⟨*Page LM0806* 311⟩
⟨*Page LM0807* 313⟩
⟨*Page LM0808* 315⟩
⟨*Page LM0809* 317⟩
⟨*Page LM0810* 319⟩
⟨*Page LM0811* 321⟩
⟨*Page LM0812* 323⟩
⟨*Page LM0813* 325⟩
⟨*Page LM0814* 327⟩
⟨*Page LM0815* 328⟩
⟨*Page LM0816* 329⟩
⟨*Page LM0817* 331⟩
⟨*Page LM0818* 333⟩
⟨*Page LM0819* 335⟩
⟨*Page LM0820* 337⟩
⟨*Page LM0821* 339⟩
⟨*Page LM0822* 340⟩
⟨*Page LM0823* 341⟩
⟨*Page LM0824* 342⟩
⟨*Page LM0825* 344⟩
⟨*Page LM0826* 346⟩
⟨*Page LM0827* 347a⟩
⟨*Page LM0828* 347b⟩

297 ⟨*Page LM0798* 297⟩≡ (296b 746)

```
                  EBANK=  E2DPS
                  COUNT*  $$/F2DPS

 # ****************************************************
 # LUNAR LANDING FLIGHT SEQUENCE TABLES
 # ****************************************************

 # FLIGHT SEQUENCE TABLES ARE ARRANGED BY FUNCTION.  THEY ARE REFERENCED USING AS AN INDEX THE R
 #       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
```

298 ⟨*Page LM0799* 298⟩≡ (296b 746)

```
# 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
```

300 ⟨*Page LM0800* 300⟩≡ (296b 746)

```
#***************************************************************
# GUILDENSTERN:  AUTO-MODES MONITOR (R13)
#***************************************************************

                COUNT*  $$/R13

# HERE IS THE PHILOSOPHY OF GUILDENSTERN:       ON EVERY APPEARANCE OR DISAPPEARANCE
# DISCRETE TO SELECT P67 OR P66 RESPECTIVELY:   ON EVERY APPEARANCE OF THE ATTITUDE-H
# 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
```

302 ⟨*Page LM0801* 302⟩≡ (296b 746)

```
                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
```

304 ⟨*Page LM0802* 304⟩≡ (296b 746)

```
  +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 --
```

306 ⟨*Page LM0803* 306⟩≡ (296b 746)

```
#               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
```

307 ⟨*Page LM0804* 307⟩≡ (296b 746)

```
                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
```

309 ⟨*Page LM0805* 309⟩≡ (296b 746)

```
                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)
```

311 ⟨*Page LM0806* 311⟩≡ (296b 746)

```
#
#       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
```

313 ⟨*Page LM0807* 313⟩≡ (296b 746)

```
                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
```

315 ⟨*Page LM0808* 315⟩≡ (296b 746)

```
                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
```

317 ⟨*Page LM0809* 317⟩≡ (296b 746)

```
                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
```

319 ⟨*Page LM0810* 319⟩≡ (296b 746)

```
                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
```

321 ⟨*Page LM0811* 321⟩≡ (296b 746)

```
                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      FLPASS0         # RESET FLPASS0

                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 P
# 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 GUID
```

```
EXGSUB          TC      INTPRET         # COMPUTE TRIM VELOCITY CORRECTION TERM.
```

323 ⟨*Page LM0812* 323⟩≡ (296b 746)

```
                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
```

325 ⟨*Page LM0813* 325⟩≡ (296b 746)

```
                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
```

327 ⟨*Page LM0814* 327⟩≡ (296b 746)

```
                TC      INTPRET
                CALL
                        FINDCDUW -2
                EXIT

#               (CONTINUE TO DISPEXIT)

# ************************************************************************
# GUIDANCE LOOP DISPLAYS
# ************************************************************************

DISPEXIT        EXTEND                  # KILL GROUP 3:  DISPLAYS WILL BE
                DCA     NEG0            #       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     GOTOP00H        # TERMINATE
```

```
                TCF     P64CEED         # PROCEED       PERMIT REDESIGNATIONS
                TCF     P64DISPS        # RECYCLE
```

328 ⟨*Page LM0815* 328⟩≡ (296b 746)

```
                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
```

329 ⟨*Page LM0816* 329⟩≡ (296b 746)

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

331 ⟨*Page LM0817* 331⟩≡ (296b 746)

```
                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
```

333 ⟨*Page LM0818* 333⟩≡ (296b 746)

```
                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
```

335 ⟨*Page LM0819* 335⟩≡ (296b 746)

```
                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
```

337 ⟨*Page LM0820* 337⟩≡ (296b 746)

```
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, TERMI
```

339 ⟨*Page LM0821* 339⟩≡ (296b 746)

```
                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
```

340 ⟨*Page LM0822* 340⟩≡ (296b 746)

```
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 MEAN
                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 DA
#       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)
#
```

341 ⟨*Page LM0823* 341⟩≡ (296b 746)

```
# THE DP RESULT IS LEFT IN MPAC UPON EXIT, AND A SP COUNT OF THE ITERATIONS TO CONVERGENCE IS L
# RETURN IS NORMALLY TO LOC(TC ROOTPSRS)+3.  IF ROOTPSRS FAILS TO CONVERGE TO IN 8 PASSES, RETU
# OUTPUTS ARE NOT TO BE TRUSTED.
#
# PRECAUTION:  ROOTPSRS MAKES NO CHECKS FOR OVERFLOW OR FOR IMPROPER USAGE.  IMPROPER USAGE COU
# PRECLUDE CONVERGENCE OR REQUIRE EXCESSIVE ITERATIONS.  AS A SPECIFIC EXAMPLE, ROOTPSRS FORMS
# COEFFICIENT TABLE BY MULTIPLYING EACH A(I) BY I, WHERE I RANGES FROM 1 TO N.  IF AN ELEMENT O
# COEFFICIENT TABLE = 1 OR >1 IN MAGNITUDE, ONLY THE EXCESS IS RETAINED.  ROOTPSRS MAY CONVERGE
# ROOT NONETHELESS, BUT IT MAY TAKE AN EXCESSIVE NUMBER OF ITERATIONS.  THEREFORE THE USER SHOU
#       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 RE
#           POWER SERIES.  THIS OVERFLOW WOULD BE PRODUCED BY SUBROUTINE POWRSERS, CALLED BY RO
#           PRECLUDE EVENTUAL CONVERGENCE.
#       3.  AT PRESENT, ERASABLE LOCATIONS ARE RESERVED ONLY FOR N UP TO 5.  AN N IN EXCESS OF
#           ALL ERASABLES USED BY ROOTPSRS ARE UNSWITCHED LOCATED IN THE REGION FROM MPAC-33 OC
#       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
```

342 ⟨*Page LM0824* 342⟩≡ (296b 746)

```
                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
```

344 ⟨*Page LM0825* 344⟩≡ (296b 746)

```
                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
```

346 ⟨*Page LM0826* 346⟩≡ (296b 746)

```
                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      P00DOO
                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
# *************************************************************************************
```

347a ⟨*Page LM0827* 347a⟩≡ (296b 746)

```
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
```

347b ⟨*Page LM0828* 347b⟩≡ (296b 746)

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

1.21 p70-p71 routines

348 ⟨*p70-p71 routines* 348⟩≡ (7)
⟨*Page LM0829* 349⟩
⟨*Page LM0830* 351⟩
⟨*Page LM0831* 353⟩
⟨*Page LM0832* 355⟩
⟨*Page LM0833* 357⟩
⟨*Page LM0834* 359⟩
⟨*Page LM0835* 361⟩
⟨*Page LM0836* 363⟩
⟨*Page LM0837* 364a⟩

349 ⟨*Page LM0829* 349⟩≡ (348 754)

```
                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 NTO 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?
```

351 ⟨*Page LM0830* 351⟩≡ (348 754)

```
                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
```

353 ⟨*Page LM0831* 353⟩≡ (348 754)

```
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
```

355 ⟨*Page LM0832* 355⟩≡ (348 754)

```
                EXTEND
                DCA     NEG0
                DXCH    -PHASE1

                EXTEND
                DCA     NEG0
                DXCH    -PHASE3

                EXTEND
                DCA     NEG0
                DXCH    -PHASE6

                CAF     THREE           # SET UP 4.3SPOT FOR GOABORT
                TS      L
                COM
                DXCH    -PHASE4

# the 3 in OCT37774 could be something else, garbled
                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
```

357 ⟨*Page LM0833* 357⟩≡ (348 754)

```
                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, SET 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
```

359 ⟨*Page LM0834* 359⟩≡ (348 754)

```
                DAD*    DMP
                        ABTCOF +2,1     #       TGO(C+TGO )
                        TGO
                DAD*    DMP
                        ABTCOF +4,1     # TGO(B+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, REPLCAE 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
```

361 ⟨*Page LM0835* 361⟩≡ (348 754)

```
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
```

363 ⟨*Page LM0836* 363⟩≡ (348 754)

```
                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

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

364a ⟨*Page LM0837* 364a⟩≡ (348 754)

```
                    EBANK=  DVCNTR
ATMAGAD             2CADR   ATMAG
ORBMANAD            ADRES   ORBMANUV
```

1.22 p12 routine

364b ⟨*p12 routine* 364b⟩≡ (7)

⟨*Page LM0838* 365⟩
⟨*Page LM0839* 367⟩
⟨*Page LM0840* 369⟩
⟨*Page LM0841* 371⟩
⟨*Page LM0842* 372a⟩

365 ⟨*Page LM0838* 365⟩≡ (364b 749)

```
                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 IS NOT 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
```

367 ⟨*Page LM0839* 367⟩≡ (364b 749)

```
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(-T)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     GOTOP00H
                TCF     +2              # PROCEED
                TCF     NEWLOAD         # ENTER NEW DATA.

                CAF     P12ADRES
                TS      WHICH

                TC      PHASCHNG
                OCT     04024
```

```
        TC      INTPRET
        DLOAD   SL
                XRANGE
                5D
        DAD
```

369 ⟨*Page LM0840* 369⟩≡ (364b 749)

```
                        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
```

371 ⟨*Page LM0841* 371⟩≡ (364b 749)

```
                        (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
                        0D
              VLOAD     PUSH
```

```
                UNITZ
        RTB     PUSH
                LOADTIME
        CALL
                RP-TO-R
```

372a ⟨*Page LM0842* 372a⟩≡ (364b 749)

```
        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
```

1.23 ascent guidance

372b ⟨*ascent guidance* 372b⟩≡ (7)

⟨*Page LM0843* 373⟩
⟨*Page LM0844* 374⟩
⟨*Page LM0845* 375⟩
⟨*Page LM0846* 377⟩
⟨*Page LM0847* 379⟩
⟨*Page LM0848* 381⟩
⟨*Page LM0849* 383⟩
⟨*Page LM0850* 385⟩
⟨*Page LM0851* 387⟩
⟨*Page LM0852* 389⟩
⟨*Page LM0853* 391⟩
⟨*Page LM0854* 393⟩
⟨*Page LM0855* 394⟩
⟨*Page LM0856* 395⟩

373 ⟨*Page LM0843* 373⟩≡ (372b 718)

```
                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
```

374 ⟨*Page LM0844* 374⟩≡ (372b 718)

```
                                ASCENT
BIT3H                   OCT     4
```

375 ⟨*Page LM0845* 375⟩≡ (372b 718)

```
                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

377 ⟨*Page LM0846* 377⟩≡ (372b 718)

```
                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
```

379 ⟨*Page LM0847* 379⟩≡ (372b 718)

```
                          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
```

381 ⟨*Page LM0848* 381⟩≡ (372b 718)

```
                                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)
```

383 ⟨*Page LM0849* 383⟩≡ (372b 718)

```
                        PCONS           # (A+B(T-T0))/TBUP*2(8)
                        TBUP
                SL1     DSU
                        GEFF            # ATR*2(9)
                STODL   ATR
                        100CS
                DMP     DAD
                        YRATE
                        YCONS           # (C+D(T-T0))*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
        OOD
        FLPI
        P12RET
BON
```

385 ⟨*Page LM0850* 385⟩≡ (372b 718)

```
                        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
```

387 ⟨*Page LM0851* 387⟩≡ (372b 718)

```
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
```

389 ⟨*Page LM0852* 389⟩≡ (372b 718)

```
                        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.
```

391 ⟨*Page LM0853* 391⟩≡ (372b 718)

```
                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
```

393 ⟨*Page LM0854* 393⟩≡ (372b 718)

```
 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     GOTOP00H

 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
```

394 ⟨*Page LM0855* 394⟩≡ (372b 718)

```
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
```

395 ⟨*Page LM0856* 395⟩≡ (372b 718)

```
# 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
```

1.24 servicer routine

396 ⟨*servicer routine* 396⟩≡ (7)
⟨*Page LM0857* 397⟩
⟨*Page LM0858* 398⟩
⟨*Page LM0859* 400⟩
⟨*Page LM0860* 401⟩
⟨*Page LM0861* 403⟩
⟨*Page LM0862* 405⟩
⟨*Page LM0863* 407⟩
⟨*Page LM0864* 408⟩
⟨*Page LM0865* 409⟩
⟨*Page LM0866* 410⟩
⟨*Page LM0867* 411⟩
⟨*Page LM0868* 412a⟩
⟨*Page LM0869* 412b⟩
⟨*Page LM0870* 413⟩
⟨*Page LM0871* 415⟩
⟨*Page LM0872* 416⟩
⟨*Page LM0873* 417⟩
⟨*Page LM0874* 418⟩
⟨*Page LM0875* 419⟩
⟨*Page LM0876* 420⟩
⟨*Page LM0877* 421⟩
⟨*Page LM0878* 422⟩
⟨*Page LM0879* 423⟩
⟨*Page LM0880* 424⟩
⟨*Page LM0881* 425⟩
⟨*Page LM0882* 426⟩
⟨*Page LM0883* 428⟩
⟨*Page LM0884* 429⟩
⟨*Page LM0885* 431⟩
⟨*Page LM0886* 433⟩
⟨*Page LM0887* 435⟩
⟨*Page LM0888* 437⟩
⟨*Page LM0889* 439⟩
⟨*Page LM0890* 441⟩
⟨*Page LM0891* 442⟩
⟨*Page LM0892* 443⟩
⟨*Page LM0893* 445⟩
⟨*Page LM0894* 447⟩
⟨*Page LM0895* 449⟩
⟨*Page LM0896* 451⟩
⟨*Page LM0897* 452a⟩

397 ⟨*Page LM0857* 397⟩≡ (396 765)

```
                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
```

398 ⟨*Page LM0858* 398⟩≡ (396 765)

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

400 ⟨*Page LM0859* 400⟩≡ (396 765)

```
                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
```

401 ⟨*Page LM0860* 401⟩≡ (396 765)

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

403 ⟨*Page LM0861* 403⟩≡ (396 765)

```
                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
```

405 ⟨*Page LM0862* 405⟩≡ (396 765)

```
                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
```

407 ⟨*Page LM0863* 407⟩≡ (396 765)

```
                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
```

408 ⟨*Page LM0864* 408⟩≡ (396 765)

```
                        AVETOMID
                CLEAR   EXIT
                        V37FLAG
AVERTRN         CA      OUTROUTE        # RETURN TO DESIRED POINT.
                TC      BANKJUMP

OUTGOAVE        =       AVERTRN
DVCNTR1         =       MASS1
```

409 ⟨*Page LM0865* 409⟩≡ (396 765)

```
                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
```

410 ⟨*Page LM0866* 410⟩≡ (396 765)

```
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
```

411 ⟨*Page LM0867* 411⟩≡ (396 765)

```
 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     NEG0
                 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
```

412a ⟨*Page LM0868* 412a⟩≡ (396 765)

```
                DXCH    RN
                CCS     ITEMP1
                TCF     COPYCYC +2
                TC      Q               # RETURN UNDER INHINT

EIGHTEEN        DEC     18
```

412b ⟨*Page LM0869* 412b⟩≡ (396 765)

```
# ************* 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 R
#       PIPA READINGS ARE STORED IN THE VECTOR DELV.  THE HIGH ORDER PART OF EACH COM
#       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
```

413 ⟨*Page LM0870* 413⟩≡ (396 765)

```
                   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
```

415 ⟨*Page LM0871* 415⟩≡ (396 765)

```
 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
```

416 ⟨*Page LM0872* 416⟩≡ (396 765)

```
                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
```

417 ⟨*Page LM0873* 417⟩≡ (396 765)

```
                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
```

418 ⟨*Page LM0874* 418⟩≡ (396 765)

```
# 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
```

419 ⟨*Page LM0875* 419⟩≡ (396 765)

```
                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
```

420 ⟨*Page LM0876* 420⟩≡ (396 765)

```
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
```

421 ⟨*Page LM0877* 421⟩≡ (396 765)

```
                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.
```

422 ⟨*Page LM0878* 422⟩≡ (396 765)

```
# ALTCHK COMPARES CURRENT ALTITUDE (IN HCALC) WITH A SPECIFIED ALTITUDE FROM A TABLE
# 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
# 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?
```

423 ⟨*Page LM0879* 423⟩≡ (396 765)

```
                TC      BANKCALL
                CADR    ALTCHK
                TCF     CONTSERV
                TC      UPFLAG
                ADRES   READLR          # SET READLR FLAG TO ENABLE LR READING.
                TCF     CONTSERV
```

424 ⟨*Page LM0880* 424⟩≡ (396 765)

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

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.
```

425 ⟨*Page LM0881* 425⟩≡ (396 765)

```
                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
```

426 ⟨*Page LM0882* 426⟩≡ (396 765)

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

428 ⟨*Page LM0883* 428⟩≡ (396 765)

```
                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
```

429 ⟨*Page LM0884* 429⟩≡ (396 765)

```
 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
```

431 ⟨*Page LM0885* 431⟩≡ (396 765)

```
                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
```

433 ⟨*Page LM0886* 433⟩≡ (396 765)

```
                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
```

435 ⟨*Page LM0887* 435⟩≡ (396 765)

```
                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.
```

437 ⟨*Page LM0888* 437⟩≡ (396 765)

```
                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
```

439 ⟨*Page LM0889* 439⟩≡ (396 765)

```
                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
```

441 ⟨*Page LM0890* 441⟩≡ (396 765)

```
                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
```

442 ⟨*Page LM0891* 442⟩≡ (396 765)

```
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.
```

443 ⟨*Page LM0892* 443⟩≡ (396 765)

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

445 ⟨*Page LM0893* 445⟩≡ (396 765)

```
                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
```

447 ⟨*Page LM0894* 447⟩≡ (396 765)

```
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
```

449 ⟨*Page LM0895* 449⟩≡ (396 765)

```
# 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     GOTOP00H        # 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.
```

451 ⟨*Page LM0896* 451⟩≡ (396 765)

```
                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
```

452a ⟨*Page LM0897* 452a⟩≡ (396 765)

```
        TC      LRADRET
```

1.25 landing analog displays

452b ⟨*landing analog displays* 452b⟩≡ (7)

⟨*Page LM0898* 453⟩
⟨*Page LM0899* 455⟩
⟨*Page LM0900* 457⟩
⟨*Page LM0901* 459⟩
⟨*Page LM0902* 461⟩
⟨*Page LM0903* 463⟩
⟨*Page LM0904* 465⟩
⟨*Page LM0905* 467⟩
⟨*Page LM0906* 469⟩
⟨*Page LM0907* 471⟩

453 ⟨*Page LM0898* 453⟩≡ (452b 743)

```
                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 *(-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
```

455 ⟨*Page LM0899* 455⟩≡ (452b 743)

```
                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 TO BE 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.
```

457 ⟨*Page LM0900* 457⟩≡ (452b 743)

```
                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.
```

459 ⟨*Page LM0901* 459⟩≡ (452b 743)

```
                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
```

461 ⟨*Page LM0902* 461⟩≡ (452b 743)

```
        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
```

463 ⟨*Page LM0903* 463⟩≡ (452b 743)

```
                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).
```

465 ⟨*Page LM0904* 465⟩≡ (452b 743)

```
                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
```

467 ⟨*Page LM0905* 467⟩≡ (452b 743)

```
                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
```

469 ⟨*Page LM0906* 469⟩≡ (452b 743)

```
                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      NEG0            # AVOIDS +0 DINC HARDWARE MALFUNCTION
```

471 ⟨*Page LM0907* 471⟩≡ (452b 743)

```
                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

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

1.26 findcduw-guidap interface

472 ⟨*findcduw-guidap interface* 472⟩≡ (7)
⟨*Page LM0908* 473⟩
⟨*Page LM0909* 474⟩
⟨*Page LM0910* 475⟩
⟨*Page LM0911* 476⟩
⟨*Page LM0912* 477⟩
⟨*Page LM0913* 478⟩
⟨*Page LM0914* 479⟩
⟨*Page LM0915* 480⟩
⟨*Page LM0916* 481⟩
⟨*Page LM0917* 483⟩
⟨*Page LM0918* 485⟩
⟨*Page LM0919* 486⟩
⟨*Page LM0920* 487⟩
⟨*Page LM0921* 488⟩
⟨*Page LM0922* 489⟩
⟨*Page LM0923* 490⟩
⟨*Page LM0924* 491⟩
⟨*Page LM0925* 492⟩

473 ⟨*Page LM0908* 473⟩≡ (472 731)

```
# 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 SUBSTITUDE 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.
#
```

474 ⟨Page LM0909 474⟩≡ (472 731)

```
# 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
#                       XOVINHIB        FLAG DENOTING X AXIS OVERRIDE INHIBITED.
#                       CSMDOCKD        FLAG DENOTING CSM DOCKED.
#                       STEERSW         FLAG DENOTING INSUFF THRUST FOR THRUST DIR F
#
# 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.
```

475 ⟨*Page LM0910* 475⟩≡ (472 731)

```
# 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
```

476 ⟨*Page LM0911* 476⟩≡ (472 731)

```
  # 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
```

477 ⟨*Page LM0912* 477⟩≡ (472 731)

```
 # 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
```

478 ⟨*Page LM0913* 478⟩≡ (472 731)

```
# 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
```

479 ⟨*Page LM0914* 479⟩≡ (472 731)

```
# 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
```

480 ⟨*Page LM0915* 480⟩≡ (472 731)

```
 # BRANCHES TO NOATTCNT
                CCS     FLPAUTNO
                TCF     NOATTCNT +2     # NO PNGCS AUTO

                CA      FLAGWRD5
                MASK    ENGONBIT
                EXTEND
                BZF     NOATTCNT +2     # ENGINE NOT ON
```

481 ⟨*Page LM0916* 481⟩≡ (472 731)

```
# LIMIT THE ATTITUDE ANGLE CHANGES
#
# THIS SECTION LIMITS THE ATTITUDE ANGLE CHANGES ABOUT A SET OF ORTHOGONAL VEHICLE AXES X,YPRIM
# THESE AXES COINCIDE WITH THE COMMANDED VEHICLE AXES IF AND ONLY IF CDUXD IS ZERO.  THE PRIME 
# THE COMMANDED VEHICLE SYSTEM ROTATED ABOUT THE X AXIS TO BRING THE Z AXIS INTO ALIGNMENT WITH
# AXIS.  ATTITUDE ANGLE CHANGES IN THE PRIME SYSTEM ARE RELATED TO SMALL GIMBAL ANGLE CHANGES B
#
#       [   -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
```

483 ⟨*Page LM0917* 483⟩≡ (472 731)

```
# 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 RATE
# PI/2 RAD/SEC.  THE CONSTANTS ARE BASED ON DELGMB BEING THE GIMBAL ANGLE CHANGES IN UNITS OF P
# AND 2 SECONDS BEING THE COMPUTATION PERIOD (THE PERIOD BETWEEN SUCCESSIVE PASSES THRU FINDCDU

                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
```

485 ⟨*Page LM0918* 485⟩≡ (472 731)

```
# 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
```

486 ⟨*Page LM0919* 486⟩≡ (472 731)

```
# 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
```

487 ⟨*Page LM0920* 487⟩≡ (472 731)

```
# NB2CDUSP RETURNS THE 2'S COMPLEMENT, PI, SP CDU ANGLES X,Y,Z IN MPAC,+1,+2 GIVEN THE MATRIX W
# ARE THE SEMI-UNIT NAV BASE VECTORS X,Y,X EXPRESSED IN STABLE MEMBER COORDINATES, LOCATED AT C
# 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
```

488 ⟨*Page LM0921* 488⟩≡ (472 731)

```
# 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
```

489 ⟨*Page LM0922* 489⟩≡ (472 731)

```
# ARCTRGSP RETURNS THE 2'S COMPLEMENT, PI, SP ANGLE IN THE A REGISTER GIVEN ITS SINE IN A AND I
# UNITS OF 2.  THE RESULT IS AN UNAMBIGUOUS ANGLE ANYWHERE IN THE CIRCLE, WITH A MAXIMUM ERROR
# THE ERROR IS PRODUCED BY THE SUBROUTINE SPARCSIN WHICH IS USED ONLY IN THE REGION +-45 DEGREE

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
```

490 ⟨*Page LM0923* 490⟩≡ (472 731)

```
# 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
```

491 ⟨*Page LM0924* 491⟩≡ (472 731)

```
# 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
```

492 ⟨*Page LM0925* 492⟩≡ (472 731)

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

1.27 lm down-telemetry program

493 ⟨*lm down-telemetry program* 493⟩≡ (7)
⟨*Page LM0988* 494⟩
⟨*Page LM0989* 495⟩
⟨*Page LM0990* 496⟩
⟨*Page LM0991* 498⟩
⟨*Page LM0992* 499⟩
⟨*Page LM0993* 501⟩
⟨*Page LM0994* 503⟩
⟨*Page LM0995* 504⟩
⟨*Page LM0996* 506⟩
⟨*Page LM0997* 507a⟩

494 ⟨*Page LM0988* 494⟩≡ (493 729)

```
# PROGRAM NAME -- DOWN TELEMETRY PROGRAM
# MOD NO. -- 0          TO COMPLETELY REWRITE THE DOWN TELEMETRY PROGRAM AND DOWNLINK
#                       PURPOSE OF SAVING APPROXIMATELY 150 WORDS OF CORE STORAGE.
#                       THIS CHANGE REQUIRES AN ENTIRELY NEW METHOD OF SPECIFYING DOW
#                       LISTS LOG SECTION FOR MORE DETAILS.  HOWEVER THIS CHANGE WILL
#                       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 DO
#       DNTMGOTO) TO THE BEGINNING OF THE CURRENT DOWNLIST (I.E., CURRENT CONTENTS OF
#       EFFECT OF IGNORING THE REMAINDER OF THE DOWNLIST WHICH THE DOWN-TELEMETRY PRO
#       THE RESTART (OR FRESH START) OCCURRED AND RESUME DOWN TELEMETRY FROM THE BEGI
#       DOWNLIST.
#
#       ALSO OF INTEREST IS THE FACT THAT ON A RESTART THE AGC WILL ZERO DOWNLINK CHA
#
# 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)
#               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.
```

495 ⟨*Page LM0989* 495⟩≡ (493 729)

496 ⟨*Page LM0990* 496⟩≡ (493 729)

```
# 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
#               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., O
#       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 A
#
# DOWNLIST LIST RESTRICTIONS:
# (THE FOLLOWING POINTS MAY BE LISTED ELSEWHERE BUT ARE LISTED HERE SO IT IS CLEAR TH
# DONE)
# 1.    SNAPSHOT DOWNLIST:
#       (A) CANNOT CONTAIN THE FOLLOWING ECADRS (I.E., 1DNADR'S): Q, 400, 1000, 1400,
#       (B) CAN CONTAIN ONLY 1DNADR'S
# 2.    ALL DOWNLINKED DATA (EXCEPT CHANNELS) IS PICKED UP BY A DCA SO DOWNLINK LISTS
#       EQUIVALENT OF THE FOLLOWING ECADRS (I.E., 1DNADRS): 377, 777, 1377, 1777, 237
#       (NOTE: THE TERM 'EQUIVALENT' MEANT THAT THE 1DNADR TO 6DNADR WILL BE PROCESSE
```

```
# 3.    CONTROL LISTS AND SUBLISTS CANNOT HAVE ENTRIES = OCTAL 00000 OR OCTAL 77777
```

498 ⟨*Page LM0991* 498⟩≡ (493 729)

```
# 4.    THE '1DNADR TIME2' WHICH WILL CAUSE THE DOWNLINK PROGRAM TO SET THE WORDER CO
#       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
#       'GARBAGE' INTO BITS15-12 OF EBANK.  HUGH BLAIR-SMITH WARNS US THAT BITS15-12
#       SIGNIFICANT SOMEDAY IN THE FUTURE.  IF/WHEN THAT HAPPENS, THE PROGRAM SHOULD
#       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
```

499 ⟨*Page LM0992* 499⟩≡ (493 729)

```
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
```

501 ⟨*Page LM0993* 501⟩≡ (493 729)

```
                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
# 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 STRUCTUR
# TO MINIMIZE TIME.  THE TIME OPTIMIZATION RESULTS IN RULES UNIQUE TO THE SNAPSHOT PORTION OF T
# 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.
```

503 ⟨*Page LM0994* 503⟩≡ (493 729)

```
                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
```

504 ⟨*Page LM0995* 504⟩≡ (493 729)

```
# SUBROUTINE NAME -- DNDUMP
#
# FUNCTIONAL DESCRIPTION -- TO SEND (DUMP) ALL ERASABLE STORAGE 'N' TIMES. (N=1 TO 4)
#       EACH BANK IS PRECEDED BY AN ID WORD, SYNCH BITS, ECADR AND TIME1 FOLLOWED BY
#       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
#       WILL BEGIN.
#
#       ONCE INITIATED THE DOWNLINK ERASABLE DUMP CAN BE TERMINATED (AND INTERRUPTED
#       BY THE FOLLOWING:
#
#       1.      A FRESH START
#       2.      COMPLETION OF ALL DOWNLINK DUMPS REQUESTED (ACCORDING TO BITS SET IN
#               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)) /
#               TIME (IN SECS) =  (   4    )*(    8    )*(    256     +     4   )  /
#               THUS TIME (IN SECS TO SEND DUMP OF ERASABLE 4 TIMES VIA DOWNLINK) = 8
#
# 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
#
#       DOWNLIST                                W
#         WORD  TAKEN FROM CONTENTS OF  EXAMPLE O       COMMENTS
#           1   ERASID                   0177X  0       DOWNLIST I.D. FOR DOWNLINK ER
#           2   LOWIDCOD                 77340  1       DOWNLINK SYNCH BITS.  (SAME (
#           3   DUMPLOC                  13400  1       (SEE NOTES ON DUMPLOC) 1 = 3R
#           4   TIME1                    14120  1       TIME IN CENTISECONDS
#           5   FIRST WORD OF EBANK X    03400  1       IN THIS EXAMPLE THIS WORD = (
```

```
 #          6   2ND   WORD OF EBANK X    00142  1       IN THIS EXAMPLE THIS WORD = CONTENTS OF
 #          7   3RD   WORD OF EBANK X    00142  1       IN THIS EXAMPLE THIS WORD = CONTENTS OF
 #          .
 #          .
 #          .
 #       260D   256TH WORD OF EBANK X    03777  1       IN THIS EXAMPLE THIS WORD = CONTENTS OF
 #
 # 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
 #                                                   COMPLETE ERASABLE DUMP NUMBER 1,2,3, OR 4 R
 #                                             EEE = EBANK BITS
 #                                        RRRRRRRR = RELATIVE ADDRESS WITHIN AN EBANK
```

506 ⟨*Page LM0996* 506⟩≡ (493 729)

```
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      NEG0            # 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 **
```

507a ⟨*Page LM0997* 507a⟩≡ (493 729)

```
                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
```

1.28 interpretive constant

507b ⟨*interpretive constant* 507b⟩≡ (7)
  ⟨*Page LM1100* 508⟩
  ⟨*Page LM1101* 509a⟩

508 ⟨*Page LM1100* 508⟩≡ (507b 740)

```
                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
```

509a ⟨*Page LM1101* 509a⟩≡ (507b 740)

```
# 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
```

1.29 agc block two self-check

509b ⟨*lm agc block two self check* 509b⟩≡ (7)

⟨*Page LM1284* 510⟩
⟨*Page LM1285* 512⟩
⟨*Page LM1286* 514⟩
⟨*Page LM1287* 516⟩
⟨*Page LM1288* 518⟩
⟨*Page LM1289* 520⟩
⟨*Page LM1290* 522⟩
⟨*Page LM1291* 524⟩
⟨*Page LM1292* 526⟩
⟨*Page LM1293* 527a⟩

510 ⟨*Page LM1284* 510⟩≡ (509b 711)

```
# PROGRAM DESCRIPTION                                   DATE:  20 DECEMBER 1967
# PROGRAM NAME -- SELF-CHECK                            LOG SECTION:  AGC BLOCK TWO SELF-CHEC
# 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 PRI
#       PART OF THE BACK-UP IDLE LOOP.  THE SECOND IS SHOW-BANKSUM WHICH RUNS AS A RI
#       STARTING VERB.
#
#       THE PURPOSE OF SELF-CHECK IS TO CHECK OUT VARIOUS PARTS OF THE COMPUTER AS OU
#
#       THE PURPOSE OF SHOW-BANKSUM IS TO DISPLAY THE SUM OF EACH BANK, ONE AT A TIMI
#
#       IN ALL THERE ARE 7 POSSIBLE OPTIONS IN THIS BLOCK II VERSION OF SELF-CHECK.
#       FOUND IN E-2065 BLOCK II AGC SELF-CHECK AND SHOW BANKSUM BY EDWIN D. SMALLY I
#
#       THE DIFFERENT OPTIONS ARE CONTROLLED BY PUTTING DIFFERENT NUMBERS IN THE SMOI
#       A DESCRIPTION OF WHAT PARTS OF THE COMPUTER THAT ARE CHECKED BY THE OPTIONS,
#       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 I
#       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
#       COMPUTER INTO BACKUP IDLE LOOP, - OPTIONS NUMBERS RESTART THE OPTION.
#
#       THE -0 OPTION PROCEEDS FROM THE LINE FOLLOWING THE LINE WHERE THE ERROR WAS I
#
#       SHOW-BANKSUM PROCEEDS UNTIL A TERMINATE IS KEYED IN (V 34 E).  THE COMPUTER I
#
```

```
# OUTPUT
```

512 ⟨*Page LM1285* 512⟩≡ (509b 711)

```
#       SELF-CHECK UPON DETECTING AN ERROR LOADS THE SELF-CHECK ALARM CONSTANT (01102
#       TURNS ON THE ALARM LIGHT.  THE OPERATOR MAY THEN DISPLAY THE THREE FAILREGS R
#       INFORMATION HE MAY KEY IN V 05 N 08 E, THE DSKY DISPLAY IN R1 WILL BE ADDRESS
#       IN R2 THE BBCON OF SELF-CHECK, AND IN R3 THE TOTAL NUMBER OF ERRORS DETECTED
#       INITIATED FRESH START (SLAP1).
#
#       SHOW-BANKSUM STARTING WITH BANK 0 DISPLAYS IN R1 THE BANK SUM (A +-NUMBER EQU
#       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 R
#       EXCEPTION IS A RESTART.  RESTART THAT OCCURS DURING ERASCHK RESTORES ERASABLE
#       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
```

514 ⟨*Page LM1286* 514⟩≡ (509b 711)

```
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
```

516 ⟨*Page LM1287* 516⟩≡ (509b 711)

```
                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
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).
```

518 ⟨*Page LM1288* 518⟩≡ (509b 711)

```
# SKEEP2 CONTROLS CHECKING OF NON-SWITCHABLE ERASABLE MEMORY WITH BANK NUMBERS IN EB
#
# ERASCHK TAKES APPROXIMATELY 7 SECONDS.

ERASCHK         CA      S+1
                TS      SKEEP2
0EBANK          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
```

520 ⟨*Page LM1289* 520⟩≡ (509b 711)

```
                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
```

522 ⟨*Page LM1290* 522⟩≡ (509b 711)

```
                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 ROUTINE 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
```

524 ⟨*Page LM1291* 524⟩≡ (509b 711)

```
                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
```

526 ⟨*Page LM1292* 526⟩≡ (509b 711)

```
                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
```

527a ⟨*Page LM1293* 527a⟩≡ (509b 711)

```
                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.
```

1.30 rtb op codes

527b ⟨*rtb op codes* 527b⟩≡ (7)

⟨*Page LM1397* 528⟩
⟨*Page LM1398* 530⟩
⟨*Page LM1399* 531a⟩
⟨*Page LM1400* 531b⟩
⟨*Page LM1401* 532⟩
⟨*Page LM1402* 533a⟩

528 ⟨*Page LM1397* 528⟩≡ (527b 763)

```
                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
# 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
# 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
```

530 ⟨*Page LM1398* 530⟩≡ (527b 763)

```
                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
# QUANTITY FOUND IN TEM2.  THIS MAY BE USED TO INCREMENT DESIRED IMU AND OPTICS CDU A
# (+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
```

531a ⟨*Page LM1399* 531a⟩≡ (527b 763)

```
                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
```

531b ⟨*Page LM1400* 531b⟩≡ (527b 763)

```
# 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
```

532 ⟨*Page LM1401* 532⟩≡ (527b 763)

```
# 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 YI
#
# 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 SEQU

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 I
# UP WHEN THE VECTOR BEING UNITIZED IS VERY SMALL -- IT WILL BLOW UP WHEN ALL COMPON
# IS USED AND THE UPPER ORDER HALVES OF ALL COMPONENTS ARE ZERO, THE MAGNITUDE RETUR
# BY A FACTOR OF 2(13) AND THE SQUARED MAGNITUDE RETURNED ATE 34D WILL BE TOO BIG BY

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
```

533a ⟨*Page LM1402* 533a⟩≡ (527b 763)

```
                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 ***
```

1.31 t6 rupt programs

533b ⟨*t6 rupt programs* 533b⟩≡ (7)

⟨*Page LM1403* 534⟩
⟨*Page LM1404* 536⟩
⟨*Page LM1405* 537⟩

534 ⟨*Page LM1403* 534⟩≡ (533b 769)

```
# 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 CONTR
# SINCE THE LM DAP MAINTAINS EXCLUSIVE CONTROL OVER TIME6 AND ITS INTERRUPTS, THE FO
# 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) INDICAT
#                       III.    SOMETIMES FOUND POSMAX INDICATING THAT IT IS INACTIVE
#                       IV.     SOMETIMES FOUND NEGATIVE ZERO INDICATING THAT:
#                               A.      A T6RUPT IS ABOUT TO OCCUR AT THE NEXT DINC,
#                               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
#               EVERY 5 MILLISECONDS TO PROCESS A POSSIBLE WAITING T6RUPT BEFORE IT C
#      (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           NXT6A
#               T6NEXT          T6NEXT +1                       T6NEXT          T6NEX
#               T6FURTHA        T6FURTHA +1                     T6FURTHA        T6FUR
#
```

```
# DEBRIS:       T6JOBCHK CLOBBERS A.  DOT6RUPT CLOBBERS NOTHING.

                BLOCK   02
```

536 ⟨*Page LM1404* 536⟩≡ (533b 769)

```
                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 PROCES

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
```

537 ⟨*Page LM1405* 537⟩≡ (533b 769)

```
                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.
```

1.32 dap interface subroutines

538a ⟨*dap interface subroutines* 538a⟩≡ (7)
⟨*Page LM1406* 538b⟩
⟨*Page LM1407* 539⟩
⟨*Page LM1408* 541⟩
⟨*Page LM1409* 542a⟩

538b ⟨*Page LM1406* 538b⟩≡ (538a 728)

```
                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 IS A COLLECTION OF SUBROUTINES WHICH ALLOW MISSION CONTROL PROGRAMS TO
#       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
```

539 ⟨*Page LM1407* 539⟩≡ (538a 728)

```
# 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 DAPBOOL
#       PFLITEDB -- SET DEADBAND TO 1.0 DEGREE AND ZERO THE COMMANDED ATTITUDE CHANGE AND COMMA
#
#       ALL ENTRIES SET UP A NOVAC JOB TO DO 1/ACCS SO THAT THE TJETLAW SWITCH CURVES ARE POSIT
#       REFLECT THE NEW DEADBAND.  IT SHOULD BE NOTED THAT THE DEADBAND REFERS TO THE ATTITUDE
#
# 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.
```

541 ⟨*Page LM1408* 541⟩≡ (538a 728)

```
 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
```

542a ⟨*Page LM1409* 542a⟩≡ (538a 728)

```
                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.
```

1.33 dapidler program

542b ⟨*dapidler program* 542b⟩≡ (7)

⟨*Page LM1410* 543⟩
⟨*Page LM1411* 544⟩
⟨*Page LM1412* 546⟩
⟨*Page LM1413* 548⟩
⟨*Page LM1414* 549⟩
⟨*Page LM1415* 551⟩
⟨*Page LM1416* 552⟩
⟨*Page LM1417* 553⟩
⟨*Page LM1418* 554⟩
⟨*Page LM1419* 556⟩
⟨*Page LM1420* 557a⟩

543 ⟨*Page LM1410* 543⟩≡ (542b 726)

```
# THE DAPIDLER PROGRAM IS STARTED BY FRESH START AND RESTART.  THE DAPIDLER PROGRAM IS DONE 10
# PER SECOND UNTIL THE ASTRONAUT DESIRES THE DAP TO WAKE UP, AND THE IMU AND CDUS ARE READY FOR
# 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
```

544 ⟨*Page LM1411* 544⟩≡ (542b 726)

```
# 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
```

546 ⟨*Page LM1412* 546⟩≡ (542b 726)

```
                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
```

548 ⟨*Page LM1413* 548⟩≡ (542b 726)

```
                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
```

549 ⟨*Page LM1414* 549⟩≡ (542b 726)

```
                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 PRESUMA
#       IF THE REVERSED BIT IS ONE, NEEDLER IS CALLED TO DISPLAY ATTITUDE ERRORS.  IF THE BIT I
#       ORS ARE CALCULATED AS 1) DAP FOLLOWING ERRORS, IF NEEDLFLG = 0, AND 2) TOTAL ATTITUDE E
#
#
# 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,M
#
#
# 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-
```

551 ⟨*Page LM1415* 551⟩≡ (542b 726)

```
                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.

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             # (CTHETA-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 B
```

552 ⟨*Page LM1416* 552⟩≡ (542b 726)

```
                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
```

553 ⟨*Page LM1417* 553⟩≡ (542b 726)

```
# 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
#
# THIS SUBROUTINE IS USED TO DISPLAY ATTITUDE ERRORS ON THE FDAI VIA THE DIGITAL TO ANALOG CONV
# IN THE CDUS.  CARE IS TAKEN TO METER OUT THE APPROPRIATE NUMBER OF PULSES TO THE IMU ERROR CO
# OVERFLOW, TO CONTROL THE RELAY SEQUENCING, AND TO AVOID INTERFERENCE WITH THE COARSE ALIGN LO
# THE DACS.
#
# CALLING SEQUENCE:
#       DURING THE INITIALIZATION SECTION OF THE USER'S PROGRAM, BIT3 OF RCSFLAGS SHOULD BE SET
#       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 FOLL
#               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
#       CORRESPONDS TO 42 3/16 DEGREES.  (DAC MAXIMUM CAPACITY IS 384 BITS.)  46 BITS EFFECTIVE
#
#       A CALL TO NEEDLER WILL THE 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 TH
#       REQUIRED BEFORE ANYTHING IS ACTUALLY DISPLAYED ON THE ERROR METERS.
#       NOTE:  EACH CALL TO NEEDLER MUST BE SEPARATED BY AT LEAST 50 MS. TO ASSURE PROPER RELAY
#
# ERASABLES USED:
#       AK              CDUXCMD
#       AK1             CDUYCMD
#       AK2             CDUZCMD
#       EDRIVEX         A,L,Q
#       EDRIVEY         T5TEMP
#       EDRIVEZ         DINDX
```

554 ⟨*Page LM1418* 554⟩≡ (542b 726)

```
#
# 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 AT LEAST 4 MS FOR
                TS      RCSFLAGS        # RELAY CLOSURE.
                TCF     RETNMORE
```

```
NEEDLES3        CAF     BIT6            # CHECK TO SEE IF IMU ERROR COUNTER
                EXTEND                  # IS ENABLED
                RAND    CHAN12
```

556 ⟨*Page LM1419* 556⟩≡ (542b 726)
```
                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
```

557a ⟨*Page LM1420* 557a⟩≡ (542b 726)

```
                  CS       LIMITS          # DUPLICATE CODING IN BANK 16
                  TC       Q

 RETNMORE         EXTEND                   # RETURN TO CHEKMORE
                  DCA      MORECADR
                  DTCB

                  EBANK=   AOSQ
 MORECADR         2CADR    CHEKMORE
```

1.34 p axis rcs autopilot

557b ⟨*p axis rcs autopilot* 557b⟩≡ (7)

⟨*Page LM1421* 558⟩
⟨*Page LM1422* 560⟩
⟨*Page LM1423* 562⟩
⟨*Page LM1424* 564⟩
⟨*Page LM1425* 566⟩
⟨*Page LM1426* 568⟩
⟨*Page LM1427* 570⟩
⟨*Page LM1428* 572⟩
⟨*Page LM1429* 574⟩
⟨*Page LM1430* 576⟩
⟨*Page LM1431* 578⟩
⟨*Page LM1432* 580⟩
⟨*Page LM1433* 582⟩
⟨*Page LM1434* 584⟩
⟨*Page LM1435* 586⟩
⟨*Page LM1436* 588⟩
⟨*Page LM1437* 590⟩
⟨*Page LM1438* 592⟩
⟨*Page LM1439* 593⟩
⟨*Page LM1440* 594⟩
⟨*Page LM1441* 595⟩

558 ⟨*Page LM1421* 558⟩≡ (557b 756)

```
                BANK    16
                SETLOC  DAPS1
                BANK

                EBANK=  PERROR
                COUNT*  $$/DAPP

# THE FOLLOWING T5RUPT ENTRY BEGINS THE PROGRAM WHICH CONTROLS THE P-AXIS ACTION OF T
# THE NOMINAL TIME BETWEEN THE P-AXIS RUPTS IS 100 MS IN ALL NON-IDLING MODES OF THE

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-AXI
# 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
```

560 ⟨*Page LM1422* 560⟩≡ (557b 756)

```
                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
```

562 ⟨*Page LM1423* 562⟩≡ (557b 756)

```
                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
```

564 ⟨*Page LM1424* 564⟩≡ (557b 756)

```
        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
```

566 ⟨*Page LM1425* 566⟩≡ (557b 756)

```
                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
```

568 ⟨*Page LM1426* 568⟩≡ (557b 756)

```
                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
```

570 ⟨*Page LM1427* 570⟩≡ (557b 756)

```
                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
```

572 ⟨*Page LM1428* 572⟩≡ (557b 756)

```
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
```

574 ⟨*Page LM1429* 574⟩≡ (557b 756)

```
                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
```

576 ⟨*Page LM1430* 576⟩≡ (557b 756)
```
                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
```

578 ⟨*Page LM1431* 578⟩≡ (557b 756)

```
                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-AUT
# THE DIRECT RATE MODE AFFORDS IMMEDIATE CONTROL WITHOUT OVERSHOOT.  THE PSEUDO-AUTO MODE PROVI
# RATE CONTROL AND ATTITUDE HOLD.
#
```

580 ⟨*Page LM1432* 580⟩≡ (557b 756)

```
# IN DIRECT RATE, JETS ARE FIRED WHEN STICK POSITION CHANGES BY A FIXED NUMBER OF INC
# THE 'BREAKOUT LEVEL' IS .6 D/S FOR LM-ONLY AND .3 D/S FOR CSM-DOCKED.  THIS LAW NUM
# THE 'TARGET DEADBAND', WHICH EQUALS THE BREAKOUT LEVEL.
#
# IN PSEUDO-AUTO, BODY-FIXED RATE AND ATTITUDE ERRORS ARE SUPPLIED TO TJETLAW, WHICH
# CONTROL SWITCHES FROM DIRECT RATE TO PSEUDO-AUTO IF THE TARGET DB IS ACHIEVED OR IF
# IF THE INITIAL COMMAND DOES NOT EXCEED THE BREAKOUT LEVEL, CONTROL GOES TO PSEUDO-A
#
# SINCE P-AXIS CONTROL IS SEPARATE FROM Q,R AXES CONTROL, IT IS POSSIBLE TO USE (1) I
# OR VICE VERSA.  THIS ALLOWS A DEGREE OF ATTITUDE HOLD IN UNCONTROLLED AXES.  DUE TO
# 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, M
# AND 4 D/S IN NORMAL AND FINE SCALING; HOWEVER, STICK SENSITIVITY AT ZERO COUNTS (OR
# OF 2 DEGREES FROM THE CENTERED POSITION) IS .5 OR .1 D/S PER DEGREE.  NORMAL AND FI
# CASE IS AUTOMATICALLY SET TO 1/10 THE ABOVE VALUES.  SCALING IS DETERMINED IN ROUTI
#
# 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
```

582 ⟨*Page LM1433* 582⟩≡ (557b 756)

```
                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.
```

584 ⟨*Page LM1434* 584⟩≡ (557b 756)

```
                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
```

586 ⟨*Page LM1435* 586⟩≡ (557b 756)

```
                        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
```

588 ⟨*Page LM1436* 588⟩≡ (557b 756)

```
                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
```

590 ⟨*Page LM1437* 590⟩≡ (557b 756)

```
                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)
```

592 ⟨*Page LM1438* 592⟩≡ (557b 756)

```
                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
```

593 ⟨*Page LM1439* 593⟩≡ (557b 756)

```
                  AD      -160MST6
                  EXTEND
                  BZMF    PJETSLEC -6
                  CA      SIX
                  TCF     PJETSLEC -1
 -160MST6         DEC     -256
 -FOURDEG         DEC     -.08888
```

594 ⟨*Page LM1440* 594⟩≡ (557b 756)

```
# 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
```

595 ⟨*Page LM1441* 595⟩≡ (557b 756)

```
                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
```

1.35 q-r axis rcs autopilot

596 ⟨*q-r axis rcs autopilot* 596⟩≡ (7)
⟨*Page LM1442* 597⟩
⟨*Page LM1443* 599⟩
⟨*Page LM1444* 601⟩
⟨*Page LM1445* 603⟩
⟨*Page LM1446* 605⟩
⟨*Page LM1447* 607⟩
⟨*Page LM1448* 609⟩
⟨*Page LM1449* 611⟩
⟨*Page LM1450* 612⟩
⟨*Page LM1451* 613⟩
⟨*Page LM1452* 614⟩
⟨*Page LM1453* 616⟩
⟨*Page LM1454* 618⟩
⟨*Page LM1455* 620⟩
⟨*Page LM1456* 622⟩
⟨*Page LM1457* 624⟩
⟨*Page LM1458* 626⟩
⟨*Page LM1459* 627a⟩

597 ⟨*Page LM1442* 597⟩≡ (596 758)

```
                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 SP
# WILL OVERRIDE THIS SPECIFICATION.  AN ALARM RESULTS WHEN NO POLICY IS AVAILABLE BECAUSE OF FA

SENSEGET        CA      BIT7            # INPUT BITS OVERRIDE THE INTERNAL BITS
                EXTEND                  # SENSETYP WILL NOT OPPOSE ANYTRANS
                RAND    CHAN31
```

```
        EXTEND
        BZF     +XORULGE
```

599 ⟨*Page LM1443* 599⟩≡ (596 758)

```
                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
```

601 ⟨*Page LM1444* 601⟩≡ (596 758)

```
                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
```

603 ⟨*Page LM1445* 603⟩≡ (596 758)

```
                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
```

605 ⟨*Page LM1446* 605⟩≡ (596 758)

```
                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
```

607 ⟨*Page LM1447* 607⟩≡ (596 758)

```
                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
```

609 ⟨*Page LM1448* 609⟩≡ (596 758)

```
                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
```

611 ⟨*Page LM1449* 611⟩≡ (596 758)

```
 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?
```

612 ⟨*Page LM1450* 612⟩≡ (596 758)

```
                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 R
                INHINT
                TC      IBNKCALL
                CADR    TIMEGMBL
                RELINT
                CAF     ZERO
                TS      INGTS
                TCF     RCS

                EBANK=  CDUXD
GTSCADR         2CADR   GTS
```

613 ⟨*Page LM1451* 613⟩≡ (596 758)

```
# SUBROUTINE TO COMPUTE Q,R-AXES ATTITUDE ERRORS FOR USE IN THE RCS AND GTS CONTROL LAWS AND TH

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      DELREROR        # KALCMANU INERFACE ERROR
                AD      E
                XCH     RERROR          # SAVE R-ERROR FOR EIGHT-BALL DISPLAY.
                TC      Q
```

614 ⟨*Page LM1452* 614⟩≡ (596 758)

```
# "ATTSTEER" IS THE ENTRY POINT FOR Q,R-AXES (U,V-AXES) ATTITUDE CONTROL USING THE RE

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
```

616 ⟨*Page LM1453* 616⟩≡ (596 758)

```
#               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
```

618 ⟨*Page LM1454* 618⟩≡ (596 758)

```
                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
```

620 ⟨*Page LM1455* 620⟩≡ (596 758)

```
                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
```

622 ⟨*Page LM1456* 622⟩≡ (596 758)

```
-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
```

624 ⟨*Page LM1457* 624⟩≡ (596 758)
```
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      NEG0
                TS      T6FURTHA

                TC      Q

# ROT-TOUV IS ENTERED WITH THE Q-COMPONENT OF THE QUANTITY TO BE TRANSFORMED IN A AN
# ROT-TOUV TRANSFORMS THE QUANTITY INTO THE NON-ORTHOGONAL U-V AXIS SYSTEM.  IN THE U
# PRODUCED FROM RCS JET FIRINGS.  AT THE COMPLETION OF ROT-TOUV, THE U-COMPONENT OF T
# 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
```

626 ⟨*Page LM1458* 626⟩≡ (596 758)

```
                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 RE

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
```

627a ⟨*Page LM1459* 627a⟩≡ (596 758)

```
                        BLOCK   3
                        SETLOC  FFTAG6
                        BANK

                        COUNT*  $$/DAP

MAKERUPT                EXTEND
                        EDRUPT  MAKERUPT
```

1.36 tjet law

627b ⟨*tjet law* 627b⟩≡ (7)

⟨*Page LM1460* 628⟩
⟨*Page LM1461* 630⟩
⟨*Page LM1462* 632⟩
⟨*Page LM1463* 634⟩
⟨*Page LM1464* 636⟩
⟨*Page LM1465* 638⟩
⟨*Page LM1466* 640⟩
⟨*Page LM1467* 641⟩
⟨*Page LM1468* 642⟩
⟨*Page LM1469* 644⟩

628 ⟨*Page LM1460* 628⟩≡ (627b 772)

```
# 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 I
# ATTITUDE-HOLD MODE TO CALCULATE THE JET-FIRING-TIME (TJET) REQUIRED FOR THE AXIS IN
#       -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
# UNBALANCED COUPLES ARE PREFERRED.  TJETLAW ALSO USES VARIOUS FUNCTIONS OF ACCELERAT
# COMPUTED IN THE 1/ACCONT SECTION OF 1/ACCS AND ARE STORED IN SUCH AN ORDER THAT THE
# ACCESSED BY INDEXING.
#
# THE SIGN OF THE REQUIRED ROTATION IS CARRIED THROUGH TJETLAW AS ROTSENSE AND IS FIN
# PREVIOUS TO ITS STORAGE IN THE LOCATION CORRESPONDING TO THE AXIS (TJP, TJU, OR TJV
# TJETLAW ASSUMES WILL BE USED AS INDICATED BY THE SETTING OF NUMBERT FOR THE U- OR V
# ASSUMED FOR THE P-AXIS ALTHOUGH FOUR JETS WILL BE FIRED WHEN FIREFCT IS MORE NEGATI
# (FIREFCT IS THE DISTANCE TO A SWITCH CURVE IN THE PHASE PLANE) AND A LONG FIRING IS
#
# IN ORDER TO AVOID SCALING DIFFICULTIES, SIMPLE ALGORITHMS TAGGED RUFLAW1, -2 AND -3
# 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
#
# 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
```

630 ⟨*Page LM1461* 630⟩≡ (627b 772)

```
                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 ACCESSEI
                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 ROUTII
#       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
```

632 ⟨*Page LM1462* 632⟩≡ (627b 772)

```
                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
                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
```

634 ⟨*Page LM1463* 634⟩≡ (627b 772)

```
                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.
```

636 ⟨*Page LM1464* 636⟩≡ (627b 772)

```
                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
```

638 ⟨*Page LM1465* 638⟩≡ (627b 772)

```
                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 DENOMINATO
                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
# 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
```

640 ⟨*Page LM1466* 640⟩≡ (627b 772)

```
# 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
```

```
                CAE     HH
                AD      NEG2
                EXTEND
                BZMF    FORMULA3
```

641 ⟨*Page LM1467* 641⟩≡ (627b 772)

```
# 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.
```

642 ⟨*Page LM1468* 642⟩≡ (627b 772)

```
# *** 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 RO
#       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 (
# RUFLAW2:      ERROR MORE POSITIVE THAN PI/16 RAD.  FIRE TO AN OPPOSING RATE OF 6.5
# RUFLAW3:      ERROR RATE GREATER THAN PI/32 RAD/SEC AND ERROR WITHIN BOUNDS.  COAS

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
```

644 ⟨*Page LM1469* 644⟩≡ (627b 772)

```
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.
```

1.37 kalman filter

645 ⟨*kalman filter* 645⟩≡ (7)
⟨*Page LM1470* 646⟩
⟨*Page LM1471* 647a⟩

646 ⟨*Page LM1470* 646⟩≡ (645 741)

```
                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
```

647a ⟨*Page LM1471* 647a⟩≡ (645 741)

```
                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
```

1.38 trim gimbal cntrol system

647b ⟨*trim gimbal cntrol system* 647b⟩≡ (7)

⟨*Page LM1472* 648⟩
⟨*Page LM1473* 649⟩
⟨*Page LM1474* 651⟩
⟨*Page LM1475* 653⟩
⟨*Page LM1476* 655⟩
⟨*Page LM1477* 656⟩
⟨*Page LM1478* 657⟩
⟨*Page LM1479* 659⟩
⟨*Page LM1480* 660⟩
⟨*Page LM1481* 661⟩
⟨*Page LM1482* 663⟩
⟨*Page LM1483* 665⟩
⟨*Page LM1484* 666⟩

648 ⟨*Page LM1472* 648⟩≡ (647b 774)

```
                BANK    21
                EBANK=  QDIFF
                SETLOC  DAPS4
                BANK

                COUNT*  $$/DAPGT

# CONTROL REACHES THIS POINT UNDER EITHER OF THE FOLLOWING TWO CONDITIONS ONCE THE DE
# AUTOPILOT ARE BOTH ON:
#       A) THE TRIM GIMBAL CONTROL LAW WAS ON DURING THE PREVIOUS Q,R-AXIS TIME5 INTE
#          INITIALIZATION WAS SET FOR TRIM GIMBAL CONTROL AND THIS IS THE FIRST PASS)
#       B) THE Q,R-AXES RCS AUTOPILOT DETERMINED THAT THE VEHICLE WAS ENTERING (OR HA
#          ZONE WITH A SMALL OFFSET ANGULAR ACCELERATION.
# GTS IS THE ENTRY TO THE GIMBAL TRIM SYSTEM FOR CONTROLLING ATTITUDE ERRORS AND RATE

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
#       NEGUSUM = ERROR*K + ALPHA*(DEL*OMEGA + ALPHA /(3*K)) + DEL*K   (DEL*OMEGA + A
#
#       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+DN        CAF     TWO             # SET INDEXER FOR R-AXIS CALCULATIONS.
                TCF     GOQTRIMG +1

GOQTRIMG        CAF     ZERO            # SET INDEXER FOR Q-AXIS CALCULATIONS
                TS      QRCNTR
```

649 ⟨*Page LM1473* 649⟩≡ (647b 774)

```
# 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
```

651 ⟨*Page LM1474* 651⟩≡ (647b 774)

```
                  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
```

653 ⟨*Page LM1475* 653⟩≡ (647b 774)

```
  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
```

655 ⟨*Page LM1476* 655⟩≡ (647b 774)

```
                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 NEG
# THE NEGATIVES OF THE DESIRED ACCELERATION CHANGES.  ACDT+C12 SETS Q(R)ACCDOT TO REFLECT THE N
#
# 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
```

656 ⟨*Page LM1477* 656⟩≡ (647b 774)

```
        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
```

657 ⟨*Page LM1478* 657⟩≡ (647b 774)

```
# SUBROUTINE TIMEGMBL:  MOD 0, OCTOBER 1967, CRAIG WORK
#
# TIMEGMBL COMPUTES THE DRIVE TIME NEEDED FOR THE TRIM GIMBAL TO POSITION THE DESCENT ENGINE NO
# THE OFFSET ANGULAR ACCELERATION ABOUT THE Q (OR R) AXIS.  INSTEAD OF USING AOSQ(R), TIMEGMBL
# 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 GI
# DRIVING IN THE CORRECT DIRECTION.  THE Q(R)GIMTIMR IS SET TO TERMINATE THE DRIVE AND Q(R)ACCD
# IS STORED TO REFLECT THE NEW ACCELERATION DERIVATIVE.  NEGUQ(R) WILL CONTAIN +1,+0,-1 FOR A Q
# 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 LA
#                Q(R)GIMTIMR WILL BE SET TO TIME AND TERMINATE GIMBAL DRIVE(S).
#
# DEBRIS:        A, L, Q, ITEMPS 2, 3, 6, AND 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 O
#                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
```

659 ⟨*Page LM1479* 659⟩≡ (647b 774)

```
                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
```

660 ⟨Page LM1480 660⟩≡ (647b 774)

```
                EXTEND
                BZF     TGOFFNOW                # DRIVE TIME MUST BE GREATER THAN ZE

                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,
                TS      ALLOWGTS                # DO NOT PERMIT FURTHER GTS ATTITUDE-
                                                # 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+C
                DXCH    ITEMP2                  # LEAVES ITEMPS2,3 ALONE.

                TC      IBNKCALL                # TURN OFF CHANNEL BITS, SET Q(R)ACCD
                CADR    ACDT+C12

                DXCH    ITEMP2                  # RESTORE ERASABLES FOR IBNKCALL.
                DXCH    RUPTREG3

                TC      RUPTREG2                # RETURN TO CALLER.

OCT00240        OCTAL   00240                   # DECIMAL 10/1024
```

661 ⟨*Page LM1481* 661⟩≡ (647b 774)

```
# THE FOLLOWING SECTION IS A CONTINUATION OF THE TRIM GIMBAL CONTROL FROM THE LAST GTS ENTRY.
# IS COMPUTED FOR EACH AXIS (Q,R), .707*DEL*FUNCTION(3/2) + K2THETA = NEGUSUM.  NEW DRIVES ARE
#
# THE SUBROUTINE GTSQRT ACCEPTS A DOUBLE PRECISION VALUE IN FUNCTION, FUNCTION +1 AND RETURNS A
# SQUARE ROOT OF THE FOURTEEN MOST SIGNIFICANT BITS OF THE ARGUMENT.  ALSO, THE CELL SHFTFLAG C
# EXPONENT S, SUCH THAT THE SQUARE ROOT (RETURNED IN THE A REGISTER) MUST BE SHIFTED RIGHT (MUL
# POWER (-S)) IN ORDER TO BE THE TRUE SQUARE ROOT OF THE FOURTEEN MOST SIGNIFICANT BITS OF FUNC
# SQUARE ROOT ERROR IS NOT MORE THAN 2 IN THE 14TH SIGNIFICANT BIT.  CELLS CLOBBERED ARE A, L,
# HALFARG, SCRATCH, SR, FUNCTION, FUNCTION +1.  GTSQRT IS CALLED BY TC GTSQRT AND RETURNS VIA T
# 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.
```

663 ⟨*Page LM1482* 663⟩≡ (647b 774)

```
                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
```

665 ⟨*Page LM1483* 665⟩≡ (647b 774)

```
                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         # NO 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
```

666 ⟨*Page LM1484* 666⟩≡ (647b 774)

```
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 ALGORITHM ITERATION.  THE
# SQUARE ROOT IS PRESUMED TO BE IN THE A REGISTER AND ONE-HALF THE SQUARE IS TAKEN FR
# TO THE SQUARE ROOT IS RETURNED IN THE A REGISTER.  DEBRIS:  A, L, SR, SCRATCH.  ROO
# 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 OVERFL

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
```

1.39 aostask and aosjob

667 ⟨*aostask and aosjob* 667⟩≡ (7)
⟨*Page LM1485* 668⟩
⟨*Page LM1486* 670⟩
⟨*Page LM1487* 672⟩
⟨*Page LM1488* 674⟩
⟨*Page LM1489* 676⟩
⟨*Page LM1490* 678⟩
⟨*Page LM1491* 679⟩
⟨*Page LM1492* 680⟩
⟨*Page LM1493* 681⟩
⟨*Page LM1494* 683⟩
⟨*Page LM1495* 685⟩
⟨*Page LM1496* 686⟩
⟨*Page LM1497* 687⟩
⟨*Page LM1498* 689⟩
⟨*Page LM1499* 691⟩
⟨*Page LM1500* 693⟩
⟨*Page LM1501* 695⟩
⟨*Page LM1502* 697⟩
⟨*Page LM1503* 699⟩
⟨*Page LM1504* 701⟩
⟨*Page LM1505* 703⟩
⟨*Page LM1506* 704⟩

668 ⟨*Page LM1485* 668⟩≡ (667 714)

```
# 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 A
#       CHANGE IN THE MASS OF THE VEHICLE, IN THE DEADBAND SELECTED, IN THE VEHICLE C
#       DOCKED), AND DURING A FRESH START OR A RESTART, 1/ACCS IS CALLED TO COMMUNICA
#
#       THE INPUTS TO 1/ACCS ARE MASS, ACCELERATION (ABDELV), DEADBAND (DB), OFFSET A
#       STAGE VERIFY BIT (CHAN30, BIT2), DOCKED BIT (DAPBOOLS, BIT13), DRIFT BIT (DAP
#       BIT14), AND SURFACE FLAG (FLAGWRDB, BIT8), AND CH5MASK.
#
#       1/ACCS COMPUTES THE JET ACCELERATIONS (1JACC, 1JACCQ, 1JACCR) AS FUNCTIONS OF
#       FORMED BY RESOLVING 1JACCQ AND 1JACCR.  IN THE DESCENT CASE, THE DESCENT ENGI
#       COMPUTED AS A FUNCTION OF MASS.  THE RATE OF CHANGE OF ACCELERATION DUE TO RO
#       ACCDOTR) IS ALSO COMPUTED IN THE DESCENT CASE.
#
#       AFTER THE ABOVE COMPUTATIONS, THE PROGRAM 1/ACCONT COMPUTES THE RECIPROCAL NE
#       AND V AXES (2 JETS FOR P-AXIS, BOTH 1 AND 2 JETS FOR U AND V AXES), AND THE E
#       THE P, U, AND V AXES.  THE ACCELERATION FUNCTIONS (ACCFCTZ1 AND ACCFCTZ5) ARE
#       FIRE AND COAST DEADBANDS AND AXISDIST ARE COMPUTED FOR EACH AXIS.  FLAT AND Z
#       MINIMUM IMPULSE ZONE, ARE COMPUTED.  1/ACCONT ALSO SETS ACCSWU AND ACCSWV, WH
#       IS NOT SUFFICIENT TO PRODUCE MINIMUM ACCELERATION.  AT THE COMPLETION OF 1/AC
#
# 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 C
#
# RESTRICTIONS:
#       1/ACCS MUST BE CALLED BY BANKCALL
```

```
#       EBANK IS SET TO 6, BUT NOT RESTORED.
```

670 ⟨*Page LM1486* 670⟩≡ (667 714)

```
                BANK    20
                SETLOC  DAPS3
                BANK

                COUNT*  $$/DAPAO

                EBANK=  AOSQ

# ENTRY IS THROUGH 1/ACCJOB OR 1/ACCSIT WHEN 1/ACCS IS TO BE DONE AS A SEPARATE NOVAC
# IT IS POSSIBLE FOR MORE THAN ONE OF THESE JOBS TO BE SET UP CONCURRENTLY.  HOWEVER,
# 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 DOWNLIS
                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
#       ENSURE THAT THE LEM MASS VALUE IS WITHIN THE ACCEPTABLE RANGE
```

```
        INHINT
        CAE     FLGWRD10        # DETERMINE WHETHER STAGED.
        MASK    APSFLBIT
        EXTEND
        BZF     DPSFLITE
```

672 ⟨*Page LM1487* 672⟩≡ (667 714)

```
                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 + CSM
                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
```

674 ⟨*Page LM1488* 674⟩≡ (667 714)

```
                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 COMP
# 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
```

676 ⟨*Page LM1489* 676⟩≡ (667 714)

```
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
# IMPLEMENTED IN BOTH THE Y-X PLANE AND THE Z-X PLANE IS -- D(ALPHA)/DT = TL/I*D(DELT
#       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
# THE RATIO OF ACCELERATION FROM PIPAS TO ACCELERATION OF GRAVITY IS THE SAME IN METRIC OR ENGI
# 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.
```

678 ⟨*Page LM1490* 678⟩≡ (667 714)

```
                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.
```

679 ⟨*Page LM1491* 679⟩≡ (667 714)

```
                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
```

680 ⟨*Page LM1492* 680⟩≡ (667 714)

```
# SUBROUTINE:   DVOVSUB
# AUTHOR:       C. WORK, MOD 0, 12 JUNE 68
# PURPOSE:      THIS SUBROUTINE PROVIDES A SINGLE-PRECISION MACHINE LANGUAGE DIVISION
#               (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
#               THE CALLING PROGRAM IS PRESUMED TO BE A JOB IN THE F BANK WHICH CONTA
#               THE DIVISOR FOR THIS ROUTINE MAY BE IN EITHER FIXED OR ERASABLE STORA
#               ASSUMED BETWEEN THE TWO HALVES OF THE DIVIDEND.  (THIS IS CERTAIN IF
#               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 DESIGNA
#               DIVISOR MAY BE IN THE DVOVSUB FBANK,FIXED-FIXED FBANK,EBANK 6, OR UNS
# OUTPUT:       QUOTIENT AND REMAINDER, OR POSMAX (NEGMAX), WHICHEVER IS APPROPRIATE
# DEBRIS:       SCRATCHX,SCRATCHY,SCRATCHZ,A,L  (NOTE: SCRATCHX,Y,Z ARE EQUATED TO MP
# 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)
```

681 ⟨*Page LM1493* 681⟩≡ (667 714)

```
                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 +
#
# 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
# 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
```

683 ⟨*Page LM1494* 683⟩≡ (667 714)

```
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
```

685 ⟨*Page LM1495* 685⟩≡ (667 714)

```
                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
```

686 ⟨*Page LM1496* 686⟩≡ (667 714)

```
                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
```

687 ⟨*Page LM1497* 687⟩≡ (667 714)

```
                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 ASOU 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
```

689 ⟨*Page LM1498* 689⟩≡ (667 714)

```
                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
```

691 ⟨*Page LM1499* 691⟩≡ (667 714)

```
                        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
```

693 ⟨*Page LM1500* 693⟩≡ (667 714)

```
                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
```

695 ⟨*Page LM1501* 695⟩≡ (667 714)

```
                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
```

697 ⟨*Page LM1502* 697⟩≡ (667 714)

```
                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
```

699 ⟨*Page LM1503* 699⟩≡ (667 714)

```
                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
```

701 ⟨*Page LM1504* 701⟩≡ (667 714)

```
 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
```

703 ⟨*Page LM1505* 703⟩≡ (667 714)

```
                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
```

704 ⟨*Page LM1506* 704⟩≡ (667 714)

```
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
```

1.40 sps back up rcs control

705 ⟨*sps back up rcs control* 705⟩≡ (7)
  ⟨*Page LM1507* 706⟩
  ⟨*Page LM1508* 707⟩
  ⟨*Page LM1509* 709⟩
  ⟨*Page LM1510* 710⟩

706 ⟨*Page LM1507* 706⟩≡ (705 768)

```
# 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 A
#       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 LIM
#
#       2. RATE DEAD BAND TEST
#               IF JETS ARE FIRING NEGATIVELY AND RATE IS GREATER THAN TARGET RATE, K
#                       JETS ON AND GO TO INHIBITION LOGIC.  OTHERWISE, CONTINUE.
#               IF JETS ARE FIRING POSITIVELY AND RATE IS   LESS  THAN TARGET RATE, K
#                       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, CONTIN
#
#       4. COAST ZONE TEST
#               IF STATE (E,EDOT) IS BELOW LINE E + 4 X EDOT > -1.4 DEG AND EDOT IS L
#                       POSITIVE AND CONTINUE.  OTHERWISE, SET JET FIRING TIME TO ZER
#               IF STATE IS ABOVE LINE E + 4 X EDOT > +1.4 DEG AND EDOT IS GREATER TH
#                       AND CONTINUE.  OTHERWISE, SET JET FIRING TIME TO ZERO AND CON
#
#       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 I
#                          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 P
#               *NOTE: INHIBITION COUNTERS CAN BE SET TO 4 OR 10 FOR THE P AND UV AXE
#               RESPECTIVELY, IN SPSRCS.  THEY ARE DECREMENTED BY ONE AT THE BEGINNIN
```

707 ⟨*Page LM1508* 707⟩≡ (705 768)

```
#               EACH DAP PASS.
#
#       THE MINIMUM PULSE WIDTH OF THIS CONTROLLER IS DETERMINED BY THE REPETITION RATE AT WHIC
#       AND IS NOMINALLY 100 MS FOR ALL AXES IN DRIFTING FLIGHT.  DURING POWERED FLIGHT THE MIN
#       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
```

709 ⟨*Page LM1509* 709⟩≡ (705 768)

```
                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
```

710 ⟨*Page LM1510* 710⟩≡ (705 768)

```
                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 ***
```

Chapter 2

Original Files

2.1 AGC BLOCK TWO SELF-CHECK

711 ⟨*src/Luminary099/AGC-BLOCK-TWO-SELF-CHECK.agc* 711⟩≡

```
# Copyright:    Public domain.
# Filename:     AGC_BLOCK_TWO_SELF_CHECK.agc
# Purpose:      Part of the source code for Luminary 1A build 099.
#               It is part of the source code for the Lunar Module's (LM)
#               Apollo Guidance Computer (AGC), for Apollo 11.
# Assembler:    yaYUL
# Contact:      Onno Hommes <ohommes@cmu.edu>.
# Website:      www.ibiblio.org/apollo.
# Pages:        1284-1293
# Mod history:  2009-05-27 OH   Transcribed from page images.
#
# This source code has been transcribed or otherwise adapted from
# digitized images of a hardcopy from the MIT Museum.  The digitization
# was performed by Paul Fjeld, and arranged for by Deborah Douglas of
# the Museum.  Many thanks to both.  The images (with suitable reduction
# in storage size and consequent reduction in image quality as well) are
# available online at www.ibiblio.org/apollo.  If for some reason you
# find that the images are illegible, contact me at info@sandroid.org
# about getting access to the (much) higher-quality images which Paul
# actually created.
#
# Notations on the hardcopy document read, in part:
#
#       Assemble revision 001 of AGC program LMY99 by NASA 2021112-61
#       16:27 JULY 14, 1969

# Page 1284
```

```
⟨Page LM1284 510⟩
# Page 1285
⟨Page LM1285 512⟩
# Page 1286
⟨Page LM1286 514⟩
# Page 1287
⟨Page LM1287 516⟩
# Page 1288
⟨Page LM1288 518⟩
# Page 1289
⟨Page LM1289 520⟩
# Page 1290
⟨Page LM1290 522⟩
# Page 1291
⟨Page LM1291 524⟩
# Page 1292
⟨Page LM1292 526⟩
# Page 1293
⟨Page LM1293 527a⟩
```

This code is written to file `src/Luminary099/AGC-BLOCK-TWO-SELF-CHECK.agc`.

2.2 AGS INITIALIZATION

713 ⟨*src/Luminary099/AGS-INITIALIZATION.agc* 713⟩≡

```
# Copyright:    Public domain.
# Filename:     AGS_INITIALIZATION.agc
# Purpose:      Part of the source code for Luminary 1A build 099.
#               It is part of the source code for the Lunar Module's (LM)
#               Apollo Guidance Computer (AGC), for Apollo 11.
# Assembler:    yaYUL
# Contact:      Hartmuth Gutsche <hgutsche@xplornet.com>.
# Website:      www.ibiblio.org/apollo.
# Pages:        206-210
# Mod history:  2009-05-19 HG   Transcribed from page images.
#
# This source code has been transcribed or otherwise adapted from
# digitized images of a hardcopy from the MIT Museum.  The digitization
# was performed by Paul Fjeld, and arranged for by Deborah Douglas of
# the Museum.  Many thanks to both.  The images (with suitable reduction
# in storage size and consequent reduction in image quality as well) are
# available online at www.ibiblio.org/apollo.  If for some reason you
# find that the images are illegible, contact me at info@sandroid.org
# about getting access to the (much) higher-quality images which Paul
# actually created.
#
# Notations on the hardcopy document read, in part:
#
#       Assemble revision 001 of AGC program LMY99 by NASA 2021112-61
#       16:27 JULY 14, 1969

# Page 206
⟨Page LM0206 71⟩
# Page 207
⟨Page LM0207 73⟩
# Page 208
⟨Page LM0208 75⟩
# Page 209
⟨Page LM0209 77⟩
# Page 210
⟨Page LM0210 78a⟩
```

This code is written to file `src/Luminary099/AGS-INITIALIZATION.agc`.

2.3 AOSTASK AND AOSJOB

714 ⟨*src/Luminary099/AOSTASK-AND-AOSJOB.agc* 714⟩≡

```
# Copyright:    Public domain.
# Filename:     AOSTASK_AND_AOSJOB.agc
# Purpose:      Part of the source code for Luminary 1A build 099.
#               It is part of the source code for the Lunar Module's (LM)
#               Apollo Guidance Computer (AGC), for Apollo 11.
# Assembler:    yaYUL
# Contact:      Ron Burkey <info@sandroid.org>.
# Website:      www.ibiblio.org/apollo.
# Pages:        1485-1506
# Mod history:  2009-05-27 RSB  Adapted from the corresponding
#                               Luminary131 file, using page
#                               images from Luminary 1A.
#               2009-06-05 RSB  Corrected a memory-bank error type.
#               2009-06-07 RSB  Corrected a typo.
#
# This source code has been transcribed or otherwise adapted from
# digitized images of a hardcopy from the MIT Museum.  The digitization
# was performed by Paul Fjeld, and arranged for by Deborah Douglas of
# the Museum.  Many thanks to both.  The images (with suitable reduction
# in storage size and consequent reduction in image quality as well) are
# available online at www.ibiblio.org/apollo.  If for some reason you
# find that the images are illegible, contact me at info@sandroid.org
# about getting access to the (much) higher-quality images which Paul
# actually created.
#
# Notations on the hardcopy document read, in part:
#
#       Assemble revision 001 of AGC program LMY99 by NASA 2021112-61
#       16:27 JULY 14, 1969

# Page 1485
⟨Page LM1485 668⟩
# Page 1486
⟨Page LM1486 670⟩
# Page 1487
⟨Page LM1487 672⟩
# Page 1488
⟨Page LM1488 674⟩
# Page 1489
⟨Page LM1489 676⟩
# Page 1490
⟨Page LM1490 678⟩
# Page 1491
```

⟨*Page LM1491* 679⟩
```
# Page 1492
```
⟨*Page LM1492* 680⟩
```
# Page 1493
```
⟨*Page LM1493* 681⟩
```
# Page 1494
```
⟨*Page LM1494* 683⟩
```
# Page 1495
```
⟨*Page LM1495* 685⟩
```
# Page 1496
```
⟨*Page LM1496* 686⟩
```
# Page 1497
```
⟨*Page LM1497* 687⟩
```
# Page 1498
```
⟨*Page LM1498* 689⟩
```
# Page 1499
```
⟨*Page LM1499* 691⟩
```
# Page 1500
```
⟨*Page LM1500* 693⟩
```
# Page 1501
```
⟨*Page LM1501* 695⟩
```
# Page 1502
```
⟨*Page LM1502* 697⟩
```
# Page 1503
```
⟨*Page LM1503* 699⟩
```
# Page 1504
```
⟨*Page LM1504* 701⟩
```
# Page 1505
```
⟨*Page LM1505* 703⟩
```
# Page 1506
```
⟨*Page LM1506* 704⟩

This code is written to file `src/Luminary099/AOSTASK-AND-AOSJOB.agc`.

2.4 AOTMARK

716 ⟨*src/Luminary099/AOTMARK.agc* 716⟩≡

```
# Copyright:    Public domain.
# Filename:     AOTMARK.agc
# Purpose:      Part of the source code for Luminary 1A build 099.
#               It is part of the source code for the Lunar Module's (LM)
#               Apollo Guidance Computer (AGC), for Apollo 11.
# Assembler:    yaYUL
# Contact:      Ron Burkey <info@sandroid.org>.
# Website:      www.ibiblio.org/apollo.
# Pages:        244-261
# Mod history:  2009-05-10 SN   (Sergio Navarro).  Started adapting
#                               from the Luminary131/ file of the same
#                               name, using Luminary099 page images.
#
# This source code has been transcribed or otherwise adapted from
# digitized images of a hardcopy from the MIT Museum.  The digitization
# was performed by Paul Fjeld, and arranged for by Deborah Douglas of
# the Museum.  Many thanks to both.  The images (with suitable reduction
# in storage size and consequent reduction in image quality as well) are
# available online at www.ibiblio.org/apollo.  If for some reason you
# find that the images are illegible, contact me at info@sandroid.org
# about getting access to the (much) higher-quality images which Paul
# actually created.
#
# Notations on the hardcopy document read, in part:
#
#       Assemble revision 001 of AGC program LMY99 by NASA 2021112-61
#       16:27 JULY 14, 1969

# Page 244
⟨Page LM0244 79⟩
# Page 245
⟨Page LM0245 80⟩
# Page 246
⟨Page LM0246 81⟩
# Page 247
⟨Page LM0247 82⟩
# Page 248
⟨Page LM0248 83⟩
# Page 249
⟨Page LM0249 84⟩
# Page 250
⟨Page LM0250 86⟩
# Page 251
```

⟨*Page LM0251* 87⟩
```
# Page 252
```
⟨*Page LM0252* 88⟩
```
# Page 253
```
⟨*Page LM0253* 90⟩
```
# Page 254
```
⟨*Page LM0254* 92⟩
```
# Page 255
```
⟨*Page LM0255* 94⟩
```
# Page 256
```
⟨*Page LM0256* 95⟩
```
# Page 257
```
⟨*Page LM0257* 96⟩
```
# Page 258
```
⟨*Page LM0258* 97⟩
```
# Page 259
```
⟨*Page LM0259* 98⟩
```
# Page 260
```
⟨*Page LM0260* 99⟩
```
# Page 261
```
⟨*Page LM0261* 100⟩

This code is written to file `src/Luminary099/AOTMARK.agc`.

2.5 ASCENT GUIDANCE

718 ⟨*src/Luminary099/ASCENT-GUIDANCE.agc* 718⟩≡

```
# Copyright:    Public domain.
# Filename:     ASCENT_GUIDANCE.agc
# Purpose:      Part of the source code for Luminary 1A build 099.
#               It is part of the source code for the Lunar Module's (LM)
#               Apollo Guidance Computer (AGC), for Apollo 11.
# Assembler:    yaYUL
# Contact:      Hartmuth Gutsche <hgutsche@xplornet.com>.
# Website:      www.ibiblio.org/apollo.
# Pages:        843-856
# Mod history:  2009-05-23 HG   Transcribed from page images.
#               2009-06-05 RSB  Fixed a couple of typos.
#               2009-06-07 RSB  Corrected a typo.
#
# This source code has been transcribed or otherwise adapted from
# digitized images of a hardcopy from the MIT Museum.  The digitization
# was performed by Paul Fjeld, and arranged for by Deborah Douglas of
# the Museum.  Many thanks to both.  The images (with suitable reduction
# in storage size and consequent reduction in image quality as well) are
# available online at www.ibiblio.org/apollo.  If for some reason you
# find that the images are illegible, contact me at info@sandroid.org
# about getting access to the (much) higher-quality images which Paul
# actually created.
#
# Notations on the hardcopy document read, in part:
#
#       Assemble revision 001 of AGC program LMY99 by NASA 2021112-61
#       16:27 JULY 14, 1969

# Page 843
⟨Page LM0843 373⟩
# Page 844
⟨Page LM0844 374⟩
# Page 845
⟨Page LM0845 375⟩
# Page 846
⟨Page LM0846 377⟩
# Page 847
⟨Page LM0847 379⟩
# Page 848
⟨Page LM0848 381⟩
# Page 849
⟨Page LM0849 383⟩
# Page 850
```

⟨*Page LM0850* 385⟩
```
# Page 851
```
⟨*Page LM0851* 387⟩
```
# Page 852
```
⟨*Page LM0852* 389⟩
```
# Page 853
```
⟨*Page LM0853* 391⟩
```
# Page 854
```
⟨*Page LM0854* 393⟩
```
# Page 855
```
⟨*Page LM0855* 394⟩
```
# Page 856
```
⟨*Page LM0856* 395⟩

This code is written to file `src/Luminary099/ASCENT-GUIDANCE.agc`.

2.6 ATTITUDE MANEUVER ROUTINE

720 ⟨*src/Luminary099/ATTITUDE-MANEUVER-ROUTINE.agc* 720⟩≡

```
# Copyright:    Public domain.
# Filename:     ATTITUDE_MANEUVER_ROUTINE.agc
# Purpose:      Part of the source code for Luminary 1A build 099.
#               It is part of the source code for the Lunar Module's (LM)
#               Apollo Guidance Computer (AGC), for Apollo 11.
# Assembler:    yaYUL
# Contact:      Ron Burkey <info@sandroid.org>.
# Website:      www.ibiblio.org/apollo.
# Pages:        342-363
# Mod history:  2009-05-16 RSB  Adapted from the corresponding
#                               Luminary131 file, using page
#                               images from Luminary 1A.
#
# This source code has been transcribed or otherwise adapted from
# digitized images of a hardcopy from the MIT Museum.  The digitization
# was performed by Paul Fjeld, and arranged for by Deborah Douglas of
# the Museum.  Many thanks to both.  The images (with suitable reduction
# in storage size and consequent reduction in image quality as well) are
# available online at www.ibiblio.org/apollo.  If for some reason you
# find that the images are illegible, contact me at info@sandroid.org
# about getting access to the (much) higher-quality images which Paul
# actually created.
#
# Notations on the hardcopy document read, in part:
#
#       Assemble revision 001 of AGC program LMY99 by NASA 2021112-61
#       16:27 JULY 14, 1969

# Page 342
⟨Page LM0342 113⟩
# Page 343
⟨Page LM0343 115⟩
# Page 344
⟨Page LM0344 117⟩
# Page 345
⟨Page LM0345 119⟩
# Page 346
⟨Page LM0346 121⟩
# Page 347
⟨Page LM0347 123⟩
# Page 348
⟨Page LM0348 125⟩
# Page 349
```

⟨*Page LM0349* 126⟩
```
# Page 350
```
⟨*Page LM0350* 127⟩
```
# Page 351
```
⟨*Page LM0351* 128⟩
```
# Page 352
```
⟨*Page LM0352* 130⟩
```
# Page 353
```
⟨*Page LM0353* 132⟩
```
# Page 354
```
⟨*Page LM0354* 134⟩
```
# Page 355
```
⟨*Page LM0355* 135⟩
```
# Page 356
```
⟨*Page LM0356* 136⟩
```
# Page 357
```
⟨*Page LM0357* 138⟩
```
# Page 358
```
⟨*Page LM0358* 140⟩
```
# Page 359
```
⟨*Page LM0359* 142⟩
```
# Page 360
```
⟨*Page LM0360* 144⟩
```
# Page 361
```
⟨*Page LM0361* 146⟩
```
# Page 362
```
⟨*Page LM0362* 148⟩
```
# Page 363
```
⟨*Page LM0363* 149a⟩

This code is written to file `src/Luminary099/ATTITUDE-MANEUVER-ROUTINE.agc`.

2.7 BURN BABY BURN–MASTER IGNITION ROUTINE

722 ⟨*src/Luminary099/BURN-BABY-BURN–MASTER-IGNITION-ROUTINE.agc* 722⟩≡

```
# Copyright:    Public domain.
# Filename:     BURN_BABY_BURN--MASTER_IGNITION_ROUTINE.agc
# Purpose:      Part of the source code for Luminary 1A build 099.
#               It is part of the source code for the Lunar Module's (LM)
#               Apollo Guidance Computer (AGC), for Apollo 11.
# Assembler:    yaYUL
# Contact:      Ron Burkey <info@sandroid.org>.
# Website:      www.ibiblio.org/apollo.
# Pages:        731-751
# Mod history:  2009-05-19 RSB  Adapted from the corresponding
#                               Luminary131 file, using page
#                               images from Luminary 1A.
#               2009-06-07 RSB  Corrected 3 typos.
#               2009-07-23 RSB  Added Onno's notes on the naming
#                               of this function, which he got from
#                               Don Eyles.
#
# This source code has been transcribed or otherwise adapted from
# digitized images of a hardcopy from the MIT Museum.  The digitization
# was performed by Paul Fjeld, and arranged for by Deborah Douglas of
# the Museum.  Many thanks to both.  The images (with suitable reduction
# in storage size and consequent reduction in image quality as well) are
# available online at www.ibiblio.org/apollo.  If for some reason you
# find that the images are illegible, contact me at info@sandroid.org
# about getting access to the (much) higher-quality images which Paul
# actually created.
#
# Notations on the hardcopy document read, in part:
#
#       Assemble revision 001 of AGC program LMY99 by NASA 2021112-61
#       16:27 JULY 14, 1969

# Page 731
## At the get-together of the AGC developers celebrating the 40th anniversary
## of the first moonwalk, Don Eyles (one of the authors of this routine along
## with Peter Adler) has related to us a little interesting history behind the
## naming of the routine.
##
## It traces back to 1965 and the Los Angeles riots, and was inspired
## by disc jockey extraordinaire and radio station owner Magnificent Montague.
## Magnificent Montague used the phrase "Burn, baby! BURN!" when spinning the
```

```
## hottest new records. Magnificent Montague was the charismatic voice of
## soul music in Chicago, New York, and Los Angeles from the mid-1950s to
## the mid-1960s.
⟨Page LM0731 242⟩
# Page 732
⟨Page LM0732 244⟩
# Page 733
⟨Page LM0733 245⟩
# Page 734
⟨Page LM0734 246⟩
# Page 735
⟨Page LM0735 248⟩
# Page 736
⟨Page LM0736 250⟩
# Page 737
⟨Page LM0737 252⟩
# Page 738
⟨Page LM0738 254⟩
# Page 739
⟨Page LM0739 256⟩
# Page 740
⟨Page LM0740 258⟩
# Page 741
⟨Page LM0741 260⟩
# Page 742
⟨Page LM0742 262⟩
# Page 743
⟨Page LM0743 264⟩
# Page 744
⟨Page LM0744 266⟩
# Page 745
⟨Page LM0745 267⟩
# Page 746
⟨Page LM0746 269⟩
# Page 747
⟨Page LM0747 271⟩
# Page 748
⟨Page LM0748 273⟩
# Page 749
⟨Page LM0749 274⟩
# Page 750
⟨Page LM0750 275⟩
# Page 751
⟨Page LM0751 277⟩
```

This code is written to file `src/Luminary099/BURN-BABY-BURN--MASTER-IGNITION-ROUTINE.agc`.

2.8 CONTROLLED CONSTANTS

724 ⟨*src/Luminary099/CONTROLLED-CONSTANTS.agc* 724⟩≡

```
# Copyright:    Public domain.
# Filename:     CONTROLLED_CONSTANTS.agc
# Purpose:      Part of the source code for Luminary 1A build 099.
#               It is part of the source code for the Lunar Module's (LM)
#               Apollo Guidance Computer (AGC), for Apollo 11.
#
# Assembler:    yaYUL
# Contact:      Jim Lawton <jim.lawton@gmail.com>
# Website:      www.ibiblio.org/apollo.
# Pages:        038-053
# Mod history:  2009-05-16      JVL     Transcribed from page images.
#
# This source code has been transcribed or otherwise adapted from digitized
# images of a hardcopy from the MIT Museum.  The digitization was performed
# by Paul Fjeld, and arranged for by Deborah Douglas of the Museum.  Many
# thanks to both.  The images (with suitable reduction in storage size and
# consequent reduction in image quality as well) are available online at
# www.ibiblio.org/apollo.  If for some reason you find that the images are
# illegible, contact me at info@sandroid.org about getting access to the
# (much) higher-quality images which Paul actually created.
#
# Notations on the hardcopy document read, in part:
#
#    Assemble revision 001 of AGC program LMY99 by NASA 2021112-061
#    16:27 JULY 14, 1969

# Page 38
⟨Page LM0038 9⟩
# Page 39
⟨Page LM0039 10⟩
# Page 40
⟨Page LM0040 11⟩
# Page 41
⟨Page LM0041 12⟩
# Page 42
⟨Page LM0042 13⟩
# Page 43
⟨Page LM0043 14⟩
# Page 44
⟨Page LM0044 15⟩
# Page 45
⟨Page LM0045 16⟩
# Page 46
```

⟨*Page LM0046* 17⟩
```
# Page 47
```
⟨*Page LM0047* 18⟩
```
# Page 48
```
⟨*Page LM0048* 19⟩
```
# Page 49
```
⟨*Page LM0049* 20⟩
```
# Page 50
```
⟨*Page LM0050* 21⟩
```
# Page 51
```
⟨*Page LM0051* 22⟩
```
# Page 52
```
⟨*Page LM0052* 23a⟩
```
# Page 53
```
⟨*Page LM0053* 23b⟩

This code is written to file `src/Luminary099/CONTROLLED-CONSTANTS.agc`.

2.9 DAPIDLER PROGRAM

726 ⟨*src/Luminary099/DAPIDLER-PROGRAM.agc* 726⟩≡

```
# Copyright:    Public domain.
# Filename:     DAPIDLER_PROGRAM.agc
# Purpose:      Part of the source code for Luminary 1A build 099.
#               It is part of the source code for the Lunar Module's (LM)
#               Apollo Guidance Computer (AGC), for Apollo 11.
# Assembler:    yaYUL
# Contact:      Ron Burkey <info@sandroid.org>.
# Website:      www.ibiblio.org/apollo.
# Pages:        1410-1420
# Mod history:  2009-05-10 SN   (Sergio Navarro).  Started adapting
#                               from the Luminary131/ file of the same
#                               name, using Luminary099 page images.
#
# This source code has been transcribed or otherwise adapted from
# digitized images of a hardcopy from the MIT Museum.  The digitization
# was performed by Paul Fjeld, and arranged for by Deborah Douglas of
# the Museum.  Many thanks to both.  The images (with suitable reduction
# in storage size and consequent reduction in image quality as well) are
# available online at www.ibiblio.org/apollo.  If for some reason you
# find that the images are illegible, contact me at info@sandroid.org
# about getting access to the (much) higher-quality images which Paul
# actually created.
#
# Notations on the hardcopy document read, in part:
#
#       Assemble revision 001 of AGC program LMY99 by NASA 2021112-61
#       16:27 JULY 14, 1969

# Page 1410
⟨Page LM1410 543⟩
# Page 1411
⟨Page LM1411 544⟩
# Page 1412
⟨Page LM1412 546⟩
# Page 1413
⟨Page LM1413 548⟩
# Page 1414
⟨Page LM1414 549⟩
# Page 1415
⟨Page LM1415 551⟩
# Page 1416
⟨Page LM1416 552⟩
# Page 1417
```

⟨*Page LM1417* 553⟩
```
# Page 1418
```
⟨*Page LM1418* 554⟩
```
# Page 1419
```
⟨*Page LM1419* 556⟩
```
# Page 1420
```
⟨*Page LM1420* 557a⟩

This code is written to file `src/Luminary099/DAPIDLER-PROGRAM.agc`.

2.10 DAP INTERFACE SUBROUTINES

728 ⟨*src/Luminary099/DAP-INTERFACE-SUBROUTINES.agc* 728⟩≡

```
# Copyright:    Public domain.
# Filename:     DAP_INTERFACE_SUBROUTINES.agc
# Purpose:      Part of the source code for Luminary 1A build 099.
#               It is part of the source code for the Lunar Module's (LM)
#               Apollo Guidance Computer (AGC), for Apollo 11.
# Assembler:    yaYUL
# Contact:      Ron Burkey <info@sandroid.org>.
# Website:      www.ibiblio.org/apollo.
# Pages:        1406-1409
# Mod history:  2009-05-10 SN   (Sergio Navarro).  Started adapting
#                               from the Luminary131/ file of the same
#                               name, using Luminary099 page images.
#
# This source code has been transcribed or otherwise adapted from
# digitized images of a hardcopy from the MIT Museum.  The digitization
# was performed by Paul Fjeld, and arranged for by Deborah Douglas of
# the Museum.  Many thanks to both.  The images (with suitable reduction
# in storage size and consequent reduction in image quality as well) are
# available online at www.ibiblio.org/apollo.  If for some reason you
# find that the images are illegible, contact me at info@sandroid.org
# about getting access to the (much) higher-quality images which Paul
# actually created.
#
# Notations on the hardcopy document read, in part:
#
#       Assemble revision 001 of AGC program LMY99 by NASA 2021112-61
#       16:27 JULY 14, 1969

# Page 1406
⟨Page LM1406 538b⟩
# Page 1407
⟨Page LM1407 539⟩
# Page 1408
⟨Page LM1408 541⟩
# Page 1409
⟨Page LM1409 542a⟩
```

This code is written to file `src/Luminary099/DAP-INTERFACE-SUBROUTINES.agc`.

2.11 DOWN TELEMETRY PROGRAM

729 ⟨*src/Luminary099/DOWN–TELEMETRY-PROGRAM.agc* 729⟩≡

```
# Copyright:    Public domain.
# Filename:     DOWN_TELEMETRY_PROGRAM.agc
# Purpose:      Part of the source code for Luminary 1A build 099.
#               It is part of the source code for the Lunar Module's (LM)
#               Apollo Guidance Computer (AGC), for Apollo 11.
# Assembler:    yaYUL
# Contact:      Ron Burkey <info@sandroid.org>.
# Website:      www.ibiblio.org/apollo.
# Pages:        988-997
# Mod history:  2009-05-24 RSB  Adapted from the corresponding
#                               Luminary131 file, using page
#                               images from Luminary 1A.
#
# This source code has been transcribed or otherwise adapted from
# digitized images of a hardcopy from the MIT Museum.  The digitization
# was performed by Paul Fjeld, and arranged for by Deborah Douglas of
# the Museum.  Many thanks to both.  The images (with suitable reduction
# in storage size and consequent reduction in image quality as well) are
# available online at www.ibiblio.org/apollo.  If for some reason you
# find that the images are illegible, contact me at info@sandroid.org
# about getting access to the (much) higher-quality images which Paul
# actually created.
#
# Notations on the hardcopy document read, in part:
#
#       Assemble revision 001 of AGC program LMY99 by NASA 2021112-61
#       16:27 JULY 14, 1969

# Page 988
⟨Page LM0988 494⟩
# Page 989 (empty page)⟨Page LM0989 495⟩
# Page 990
⟨Page LM0990 496⟩
# Page 991
⟨Page LM0991 498⟩
# Page 992
⟨Page LM0992 499⟩
# Page 993
⟨Page LM0993 501⟩
# Page 994
⟨Page LM0994 503⟩
# Page 995
⟨Page LM0995 504⟩
```

```
# Page 996
```
⟨*Page LM0996* 506⟩

```
# Page 997
```
⟨*Page LM0997* 507a⟩

This code is written to file `src/Luminary099/DOWN--TELEMETRY-PROGRAM.agc`.

2.12 FINDCDUW–GUIDAP INTERFACE

731 ⟨*src/Luminary099/FINDCDUW–GUIDAP-INTERFACE.agc* 731⟩≡

```
# Copyright:    Public domain.
# Filename:     FINDCDUW--GUIDAP_INTERFACE.agc
# Purpose:      Part of the source code for Luminary 1A build 099.
#               It is part of the source code for the Lunar Module's (LM)
#               Apollo Guidance Computer (AGC), for Apollo 11.
# Assembler:    yaYUL
# Contact:      Hartmuth Gutsche <hgutsche@xplornet.com>.
# Website:      www.ibiblio.org/apollo.
# Pages:        908-925
# Mod history:  2009-05-28 HG   Transcribed from page images.
#
# This source code has been transcribed or otherwise adapted from
# digitized images of a hardcopy from the MIT Museum.  The digitization
# was performed by Paul Fjeld, and arranged for by Deborah Douglas of
# the Museum.  Many thanks to both.  The images (with suitable reduction
# in storage size and consequent reduction in image quality as well) are
# available online at www.ibiblio.org/apollo.  If for some reason you
# find that the images are illegible, contact me at info@sandroid.org
# about getting access to the (much) higher-quality images which Paul
# actually created.
#
# Notations on the hardcopy document read, in part:
#
#       Assemble revision 001 of AGC program LMY99 by NASA 2021112-61
#       16:27 JULY 14, 1969

# Page 908
⟨Page LM0908 473⟩
# Page 909
⟨Page LM0909 474⟩
# Page 910
⟨Page LM0910 475⟩
# Page 911
⟨Page LM0911 476⟩
# Page 912
⟨Page LM0912 477⟩
# Page 913
⟨Page LM0913 478⟩
# Page 914
⟨Page LM0914 479⟩
# Page 915
⟨Page LM0915 480⟩
# Page 916
```

```
⟨Page LM0916 481⟩
# Page 917
⟨Page LM0917 483⟩
# Page 918
⟨Page LM0918 485⟩
# Page 919
⟨Page LM0919 486⟩
# Page 920
⟨Page LM0920 487⟩
# Page 921
⟨Page LM0921 488⟩
# Page 922
⟨Page LM0922 489⟩
# Page 923
⟨Page LM0923 490⟩
# Page 924
⟨Page LM0924 491⟩
# Page 925
⟨Page LM0925 492⟩
```

This code is written to file `src/Luminary099/FINDCDUW--GUIDAP-INTERFACE.agc`.

2.13 FLAGWORD ASSIGNMENTS

733 ⟨*src/Luminary099/FLAGWORD-ASSIGNMENTS.agc* 733⟩≡

```
# Copyright:    Public domain.
# Filename:     FLAGWORD_ASSIGNMENTS.agc
# Purpose:      Part of the source code for Luminary 1A build 099.
#               It is part of the source code for the Lunar Module's (LM)
#               Apollo Guidance Computer (AGC), for Apollo 11.
# Assembler:    yaYUL
# Contact:      Onno Hommes <ohommes@cmu.edu>.
# Website:      www.ibiblio.org/apollo.
# Pages:        0061-0089
# Mod history:  2009-05-15 OH   Transcribed from page images.
#               2009-05-17 RSB  Extended to (blank) p. 89.
#
# This source code has been transcribed or otherwise adapted from
# digitized images of a hardcopy from the MIT Museum.  The digitization
# was performed by Paul Fjeld, and arranged for by Deborah Douglas of
# the Museum.  Many thanks to both.  The images (with suitable reduction
# in storage size and consequent reduction in image quality as well) are
# available online at www.ibiblio.org/apollo.  If for some reason you
# find that the images are illegible, contact me at info@sandroid.org
# about getting access to the (much) higher-quality images which Paul
# actually created.
#
# Notations on the hardcopy document read, in part:
#
#       Assemble revision 001 of AGC program LMY99 by NASA 2021112-61
#       16:27 JULY 14, 1969


# Page 61
⟨Page LM0061 32⟩
# Page 62
⟨Page LM0062 33⟩
# Page63
⟨Page LM0063 35⟩
# Page 64
⟨Page LM0064 37⟩
# Page 65
⟨Page LM0065 39⟩
# Page 66
⟨Page LM0066 40⟩
# Page 67
⟨Page LM0067 41⟩
# Page 68
```

```
⟨Page LM0068 42⟩
# Page 69
⟨Page LM0069 43⟩
# Page 70
⟨Page LM0070 44⟩
# Page 71
⟨Page LM0071 45⟩
# Page 72
⟨Page LM0072 46⟩
# Page 73
⟨Page LM0073 48⟩
# Page 74
⟨Page LM0074 49⟩
# Page 75
⟨Page LM0075 50⟩
# Page 76
⟨Page LM0076 51⟩
# Page 77
⟨Page LM0077 52⟩
# Page 78
⟨Page LM0078 53⟩
# Page 79
⟨Page LM0079 54⟩
# Page 80
⟨Page LM0080 55⟩
# Page 81
⟨Page LM0081 56⟩
# Page 82
⟨Page LM0082 57⟩
# Page 83
⟨Page LM0083 59⟩
# Page 84
⟨Page LM0084 60⟩
# Page 85
⟨Page LM0085 61⟩
# Page 86
⟨Page LM0086 62⟩
# Page 87
⟨Page LM0087 63⟩
# Page 88
⟨Page LM0088 64a⟩
# Page 89 (nothing on this page)
⟨Page LM0089 64b⟩
```

This code is written to file `src/Luminary099/FLAGWORD-ASSIGNMENTS.agc`.

2.14 IMU PERFORMANCE TEST 2

735 ⟨*src/Luminary099/IMU-PERFORMANCE-TEST-2.agc* 735⟩≡

```
# Copyright:    Public domain.
# Filename:     IMU_PERFORMANCE_TEST_2.agc
# Purpose:      Part of the source code for Luminary 1A build 099.
#               It is part of the source code for the Lunar Module's (LM)
#               Apollo Guidance Computer (AGC), for Apollo 11.
# Assembler:    yaYUL
# Contact:      Ron Burkey <info@sandroid.org>.
# Website:      www.ibiblio.org/apollo.
# Pages:        373-381
# Mod history:  2009-05-17 RSB  Adapted from the corresponding
#                               Luminary131 file, using page
#                               images from Luminary 1A.
#
# This source code has been transcribed or otherwise adapted from
# digitized images of a hardcopy from the MIT Museum.  The digitization
# was performed by Paul Fjeld, and arranged for by Deborah Douglas of
# the Museum.  Many thanks to both.  The images (with suitable reduction
# in storage size and consequent reduction in image quality as well) are
# available online at www.ibiblio.org/apollo.  If for some reason you
# find that the images are illegible, contact me at info@sandroid.org
# about getting access to the (much) higher-quality images which Paul
# actually created.
#
# Notations on the hardcopy document read, in part:
#
#       Assemble revision 001 of AGC program LMY99 by NASA 2021112-61
#       16:27 JULY 14, 1969

# Page 373
⟨Page LM0373 150⟩
# Page 374
⟨Page LM0374 152⟩
# Page 375
⟨Page LM0375 154⟩
# Page 376
⟨Page LM0376 156⟩
# Page 377
⟨Page LM0377 157⟩
# Page 378
⟨Page LM0378 158⟩
# Page 379
⟨Page LM0379 159⟩
# Page 380
```

⟨*Page LM0380* 160⟩

```
# Page 381
```

⟨*Page LM0381* 162⟩

This code is written to file `src/Luminary099/IMU-PERFORMANCE-TEST-2.agc`.

2.15 IMU PERFORMANCE TESTS 4

737 ⟨*src/Luminary099/IMU-PERFORMANCE-TESTS-4.agc* 737⟩≡

```
# Copyright:    Public domain.
# Filename:     IMU_PERFORMANCE_TESTS_4.agc
# Purpose:      Part of the source code for Luminary 1A build 099.
#               It is part of the source code for the Lunar Module's (LM)
#               Apollo Guidance Computer (AGC), for Apollo 11.
# Assembler:    yaYUL
# Contact:      Ron Burkey <info@sandroid.org>.
# Website:      www.ibiblio.org/apollo.
# Pages:        382-389
# Mod history:  2009-05-17 RSB  Adapted from the corresponding
#                               Luminary131 file, using page
#                               images from Luminary 1A.
#
# This source code has been transcribed or otherwise adapted from
# digitized images of a hardcopy from the MIT Museum.  The digitization
# was performed by Paul Fjeld, and arranged for by Deborah Douglas of
# the Museum.  Many thanks to both.  The images (with suitable reduction
# in storage size and consequent reduction in image quality as well) are
# available online at www.ibiblio.org/apollo.  If for some reason you
# find that the images are illegible, contact me at info@sandroid.org
# about getting access to the (much) higher-quality images which Paul
# actually created.
#
# Notations on the hardcopy document read, in part:
#
#       Assemble revision 001 of AGC program LMY99 by NASA 2021112-61
#       16:27 JULY 14, 1969

# Page 382
⟨Page LM0382 163b⟩
# Page 383
⟨Page LM0383 164⟩
# Page 384
⟨Page LM0384 165⟩
# Page 385
⟨Page LM0385 166⟩
# Page 386
⟨Page LM0386 168⟩
# Page 387
⟨Page LM0387 170⟩
# Page 388
⟨Page LM0388 172⟩
# Page 389
```

⟨*Page LM0389* 173a⟩

This code is written to file `src/Luminary099/IMU-PERFORMANCE-TESTS-4.agc`.

2.16 INPUT OUTPUT CHANNEL BIT DESCRIPTIONS

739 ⟨*src/Luminary099/INPUT-OUTPUT-CHANNEL-BIT-DESCRIPTIONS.agc* 739⟩≡

```
# Copyright:    Public domain.
# Filename:     INPUT_OUTPUT_CHANNEL_BIT_DESCRIPTIONS.agc
# Purpose:      Part of the source code for Luminary 1A build 099.
#               It is part of the source code for the Lunar Module's (LM)
#               Apollo Guidance Computer (AGC), for Apollo 11.
# Assembler:    yaYUL
# Contact:      Onno Hommes <ohommes@cmu.edu>.
# Website:      www.ibiblio.org/apollo.
# Pages:        0054-0060
# Mod history:  2009-05-14 OH   Transcribed from page images.
#
# This source code has been transcribed or otherwise adapted from
# digitized images of a hardcopy from the MIT Museum.  The digitization
# was performed by Paul Fjeld, and arranged for by Deborah Douglas of
# the Museum.  Many thanks to both.  The images (with suitable reduction
# in storage size and consequent reduction in image quality as well) are
# available online at www.ibiblio.org/apollo.  If for some reason you
# find that the images are illegible, contact me at info@sandroid.org
# about getting access to the (much) higher-quality images which Paul
# actually created.
#
# Notations on the hardcopy document read, in part:
#
#       Assemble revision 001 of AGC program LMY99 by NASA 2021112-61
#       16:27 JULY 14, 1969

# Page 54
⟨Page LM0054 25⟩
# Page 55
⟨Page LM0055 26⟩
# Page 56
⟨Page LM0056 27⟩
# Page 57
⟨Page LM0057 28⟩
# Page 58
⟨Page LM0058 29⟩
# Page 59
⟨Page LM0059 30a⟩
# Page 60
⟨Page LM0060 30b⟩
```

This code is written to file `src/Luminary099/INPUT-OUTPUT-CHANNEL-BIT-DESCRIPTIONS.agc`.

2.17 INTERPRETIVE CONSTANT

740 ⟨*src/Luminary099/INTERPRETIVE-CONSTANT.agc* 740⟩≡

```
# Copyright:    Public domain.
# Filename:     INTERPRETIVE_CONSTANT.agc
# Purpose:      Part of the source code for Luminary 1A build 099.
#               It is part of the source code for the Lunar Module's (LM)
#               Apollo Guidance Computer (AGC), for Apollo 11.
# Assembler:    yaYUL
# Contact:      Ron Burkey <info@sandroid.org>.
# Website:      www.ibiblio.org/apollo.
# Pages:        1100-1101
# Mod history:  2009-05-25 RSB  Adapted from the corresponding
#                               Luminary131 file, using page
#                               images from Luminary 1A.
#
# This source code has been transcribed or otherwise adapted from
# digitized images of a hardcopy from the MIT Museum.  The digitization
# was performed by Paul Fjeld, and arranged for by Deborah Douglas of
# the Museum.  Many thanks to both.  The images (with suitable reduction
# in storage size and consequent reduction in image quality as well) are
# available online at www.ibiblio.org/apollo.  If for some reason you
# find that the images are illegible, contact me at info@sandroid.org
# about getting access to the (much) higher-quality images which Paul
# actually created.
#
# Notations on the hardcopy document read, in part:
#
#       Assemble revision 001 of AGC program LMY99 by NASA 2021112-61
#       16:27 JULY 14, 1969

# Page 1100
⟨Page LM1100 508⟩
# Page 1101
⟨Page LM1101 509a⟩
```

This code is written to file `src/Luminary099/INTERPRETIVE-CONSTANT.agc`.

2.18 KALMAN FILTER

741 ⟨*src/Luminary099/KALMAN-FILTER.agc* 741⟩≡

```
# Copyright:    Public domain.
# Filename:     KALMAN_FILTER.agc
# Purpose:      Part of the source code for Luminary 1A build 099.
#               It is part of the source code for the Lunar Module's (LM)
#               Apollo Guidance Computer (AGC), for Apollo 11.
# Assembler:    yaYUL
# Contact:      Ron Burkey <info@sandroid.org>.
# Website:      www.ibiblio.org/apollo.
# Pages:        1470-1471
# Mod history:  2009-05-27 RSB  Adapted from the corresponding
#                               Luminary131 file, using page
#                               images from Luminary 1A.
#
# This source code has been transcribed or otherwise adapted from
# digitized images of a hardcopy from the MIT Museum.  The digitization
# was performed by Paul Fjeld, and arranged for by Deborah Douglas of
# the Museum.  Many thanks to both.  The images (with suitable reduction
# in storage size and consequent reduction in image quality as well) are
# available online at www.ibiblio.org/apollo.  If for some reason you
# find that the images are illegible, contact me at info@sandroid.org
# about getting access to the (much) higher-quality images which Paul
# actually created.
#
# Notations on the hardcopy document read, in part:
#
#       Assemble revision 001 of AGC program LMY99 by NASA 2021112-61
#       16:27 JULY 14, 1969

# Page 1470
```
⟨*Page LM1470* 646⟩
```
# Page 1471
```
⟨*Page LM1471* 647a⟩

This code is written to file `src/Luminary099/KALMAN-FILTER.agc`.

2.19 LAMBERT AIMPOINT GUIDANCE

742 ⟨*src/Luminary099/LAMBERT-AIMPOINT-GUIDANCE.agc* 742⟩≡

```
# Copyright:    Public domain.
# Filename:     LAMBERT_AIMPOINT_GUIDANCE.agc
# Purpose:      Part of the source code for Luminary 1A build 099.
#               It is part of the source code for the Lunar Module's (LM)
#               Apollo Guidance Computer (AGC), for Apollo 11.
# Assembler:    yaYUL
# Contact:      Ron Burkey <info@sandroid.org>.
# Website:      www.ibiblio.org/apollo.
# Pages:        651-653
# Mod history:  2009-05-18 RSB  Transcribed from Luminary 099
#                               page images.
#               2009-06-05 RSB  Corrected 4 typos.
#               2009-06-07 RSB  Fixed a typo.
#
# This source code has been transcribed or otherwise adapted from
# digitized images of a hardcopy from the MIT Museum.  The digitization
# was performed by Paul Fjeld, and arranged for by Deborah Douglas of
# the Museum.  Many thanks to both.  The images (with suitable reduction
# in storage size and consequent reduction in image quality as well) are
# available online at www.ibiblio.org/apollo.  If for some reason you
# find that the images are illegible, contact me at info@sandroid.org
# about getting access to the (much) higher-quality images which Paul
# actually created.
#
# Notations on the hardcopy document read, in part:
#
#       Assemble revision 001 of AGC program LMY99 by NASA 2021112-61
#       16:27 JULY 14, 1969

# Page 651
⟨Page LM0651 237⟩
# Page 652
⟨Page LM0652 239⟩
# Page 653
⟨Page LM0653 240⟩
```

This code is written to file `src/Luminary099/LAMBERT-AIMPOINT-GUIDANCE.agc`.

2.20 LANDING ANALOG DISPLAYS

743 ⟨*src/Luminary099/LANDING-ANALOG-DISPLAYS.agc* 743⟩≡

```
# Copyright:    Public domain.
# Filename:     LANDING_ANALOG_DISPLAYS.agc
# Purpose:      Part of the source code for Luminary, build 099. It
#               is part of the source code for the Lunar Module's
#               (LM) Apollo Guidance Computer (AGC), Apollo 11.
# Assembler:    yaYUL
# Reference:    pp. 898-907
# Contact:      Ron Burkey <info@sandroid.org>,
#               Fabrizio Bernardini <fabrizio@spacecraft.it>
# Website:      http://www.ibiblio.org/apollo.
# Mod history:  05/06/09 FB     Transcription Batch 4 Assignment.
#
# The contents of the "Luminary099" files, in general, are transcribed
# from scanned documents.
#
#       Assemble revision 001 of AGC program Luminary099 by NASA
#       2021112-061.  July 14, 1969.
#
#       Prepared by
#                       Massachusetts Institute of Technology
#                       75 Cambridge Parkway
#                       Cambridge, Massachusetts
#
#       under NASA contract NAS 9-4065.
#
# Refer directly to the online document mentioned above for further
# information.  Please report any errors to info@sandroid.org.

# Page 898
⟨Page LM0898 453⟩
# Page 899
⟨Page LM0899 455⟩
# Page 900
⟨Page LM0900 457⟩
# Page 901
⟨Page LM0901 459⟩
# Page 902
⟨Page LM0902 461⟩
# Page 903
⟨Page LM0903 463⟩
# Page 904
⟨Page LM0904 465⟩
# Page 905
```

```
⟨Page LM0905 467⟩
# Page 906
⟨Page LM0906 469⟩
# Page 907
⟨Page LM0907 471⟩
```

This code is written to file `src/Luminary099/LANDING-ANALOG-DISPLAYS.agc`.

2.21 LEM GEOMETRY

745 ⟨*src/Luminary099/LEM-GEOMETRY.agc* 745⟩≡

```
# Copyright:    Public domain.
# Filename:     LEM_GEOMETRY.agc
# Purpose:      Part of the source code for Luminary 1A build 099.
#               It is part of the source code for the Lunar Module's (LM)
#               Apollo Guidance Computer (AGC), for Apollo 11.
# Assembler:    yaYUL
# Contact:      Ron Burkey <info@sandroid.org>.
# Website:      www.ibiblio.org/apollo.
# Pages:        320-325
# Mod history:  2009-05-16 RSB  Adapted from the corresponding
#                               Luminary131 file, using page
#                               images from Luminary 1A.
#
# This source code has been transcribed or otherwise adapted from
# digitized images of a hardcopy from the MIT Museum.  The digitization
# was performed by Paul Fjeld, and arranged for by Deborah Douglas of
# the Museum.  Many thanks to both.  The images (with suitable reduction
# in storage size and consequent reduction in image quality as well) are
# available online at www.ibiblio.org/apollo.  If for some reason you
# find that the images are illegible, contact me at info@sandroid.org
# about getting access to the (much) higher-quality images which Paul
# actually created.
#
# Notations on the hardcopy document read, in part:
#
#       Assemble revision 001 of AGC program LMY99 by NASA 2021112-61
#       16:27 JULY 14, 1969

# Page 320
⟨page LM0320 102⟩
# Page 321
⟨page LM0321 103⟩
# Page 322
⟨page LM0322 104⟩
# Page 323
⟨page LM0323 105⟩
# Page 324
⟨page LM0324 106⟩
# Page 325
⟨page LM0325 107a⟩
```

This code is written to file `src/Luminary099/LEM-GEOMETRY.agc`.

2.22 LUNAR LANDING GUIDANCE EQUATIONS

746 ⟨*src/Luminary099/LUNAR-LANDING-GUIDANCE-EQUATIONS.agc* 746⟩≡

```
# Copyright:    Public domain.
# Filename:     LUNAR_LANDING_GUIDANCE_EQUATIONS.agc
# Purpose:      Part of the source code for Luminary 1A build 099.
#               It is part of the source code for the Lunar Module's (LM)
#               Apollo Guidance Computer (AGC), for Apollo 11.
# Assembler:    yaYUL
# Contact:      HARTMUTH GUTSCHE <hgutsche@xplornet.com>.
# Website:      www.ibiblio.org/apollo.
# Pages:        798-828
# Mod history:  2009-05-23 HG   Transcribed from page images.
#               2009-06-05 RSB  Fixed a goofy thing that was apparently
#                               legal in GAP but not in yaYUL.  Eliminated
#                               a couple of lines of code that shouldn't
#                               have survived from Luminary 131 to here.
#               2009-06-07 RSB  Fixed a typo.
#
# This source code has been transcribed or otherwise adapted from
# digitized images of a hardcopy from the MIT Museum.  The digitization
# was performed by Paul Fjeld, and arranged for by Deborah Douglas of
# the Museum.  Many thanks to both.  The images (with suitable reduction
# in storage size and consequent reduction in image quality as well) are
# available online at www.ibiblio.org/apollo.  If for some reason you
# find that the images are illegible, contact me at info@sandroid.org
# about getting access to the (much) higher-quality images which Paul
# actually created.
#
# Notations on the hardcopy document read, in part:
#
#       Assemble revision 001 of AGC program LMY99 by NASA 2021112-61
#       16:27 JULY 14, 1969

# Page 798
⟨Page LM0798 297⟩
# Page 799
⟨Page LM0799 298⟩
# Page 800
⟨Page LM0800 300⟩
# Page 801
⟨Page LM0801 302⟩
# Page 802
⟨Page LM0802 304⟩
```

```
# Page 803
```
⟨*Page LM0803* 306⟩
```
# Page 804
```
⟨*Page LM0804* 307⟩
```
# Page 805
```
⟨*Page LM0805* 309⟩
```
# Page 806 actually starts one line earlier but that would separate the markers from their vari
```
⟨*Page LM0806* 311⟩
```
# Page 807
```
⟨*Page LM0807* 313⟩
```
# Page 808
```
⟨*Page LM0808* 315⟩
```
# Page 809
```
⟨*Page LM0809* 317⟩
```
# Page 810
```
⟨*Page LM0810* 319⟩
```
# Page 811
```
⟨*Page LM0811* 321⟩
```
# Page 812
```
⟨*Page LM0812* 323⟩
```
# Page 813
```
⟨*Page LM0813* 325⟩
```
# Page 814
```
⟨*Page LM0814* 327⟩
```
# Page 815
```
⟨*Page LM0815* 328⟩
```
# Page 816
```
⟨*Page LM0816* 329⟩
```
# Page 817
```
⟨*Page LM0817* 331⟩
```
# Page 818
```
⟨*Page LM0818* 333⟩
```
# Page 819
```
⟨*Page LM0819* 335⟩
```
# Page 820
```
⟨*Page LM0820* 337⟩
```
# Page 821
```
⟨*Page LM0821* 339⟩
```
# Page 822
```
⟨*Page LM0822* 340⟩
```
# Page 823
```
⟨*Page LM0823* 341⟩
```
# Page 824
```
⟨*Page LM0824* 342⟩
```
# Page 825
```
⟨*Page LM0825* 344⟩

```
# Page 826
```
⟨*Page LM0826* 346⟩
```
# Page 827
```
⟨*Page LM0827* 347a⟩
```
# Page 828
```
⟨*Page LM0828* 347b⟩

This code is written to file `src/Luminary099/LUNAR-LANDING-GUIDANCE-EQUATIONS.agc`.

2.23 P12

749 ⟨*src/Luminary099/P12.agc* 749⟩≡

```
# Copyright:    Public domain.
# Filename:     P12.agc
# Purpose:      Part of the source code for Luminary 1A build 099.
#               It is part of the source code for the Lunar Module's (LM)
#               Apollo Guidance Computer (AGC), for Apollo 11.
# Assembler:    yaYUL
# Contact:      Hartmuth Gutsche <hgutsche@xplornet.com>.
# Website:      www.ibiblio.org/apollo.
# Pages:        838-842
# Mod history:  2009-05-23 HG   Transcribed from page images.
#
# This source code has been transcribed or otherwise adapted from
# digitized images of a hardcopy from the MIT Museum.  The digitization
# was performed by Paul Fjeld, and arranged for by Deborah Douglas of
# the Museum.  Many thanks to both.  The images (with suitable reduction
# in storage size and consequent reduction in image quality as well) are
# available online at www.ibiblio.org/apollo.  If for some reason you
# find that the images are illegible, contact me at info@sandroid.org
# about getting access to the (much) higher-quality images which Paul
# actually created.
#
# Notations on the hardcopy document read, in part:
#
#       Assemble revision 001 of AGC program LMY99 by NASA 2021112-61
#       16:27 JULY 14, 1969

# Page 838
⟨Page LM0838 365⟩
# Page 839
⟨Page LM0839 367⟩
# Page 840
⟨Page LM0840 369⟩
# Page 841
⟨Page LM0841 371⟩
# Page 842
⟨Page LM0842 372a⟩
```

This code is written to file `src/Luminary099/P12.agc`.

2.24 P30 P37

750 ⟨*src/Luminary099/P30–P37.agc* 750⟩≡

```
# Copyright:    Public domain.
# Filename:     P30_P37.agc
# Purpose:      Part of the source code for Luminary 1A build 099.
#               It is part of the source code for the Lunar Module's (LM)
#               Apollo Guidance Computer (AGC), for Apollo 11.
# Assembler:    yaYUL
# Contact:      Ron Burkey <info@sandroid.org>.
# Website:      www.ibiblio.org/apollo.
# Pages:        614-617
# Mod history:  2009-05-17 RSB  Adapted from the corresponding
#                               Luminary131 file, using page
#                               images from Luminary 1A.
#               2009-06-05 RSB  Removed 4 lines of code that shouldn't
#                               have survived from Luminary 131.
#
# This source code has been transcribed or otherwise adapted from
# digitized images of a hardcopy from the MIT Museum.  The digitization
# was performed by Paul Fjeld, and arranged for by Deborah Douglas of
# the Museum.  Many thanks to both.  The images (with suitable reduction
# in storage size and consequent reduction in image quality as well) are
# available online at www.ibiblio.org/apollo.  If for some reason you
# find that the images are illegible, contact me at info@sandroid.org
# about getting access to the (much) higher-quality images which Paul
# actually created.
#
# Notations on the hardcopy document read, in part:
#
#       Assemble revision 001 of AGC program LMY99 by NASA 2021112-61
#       16:27 JULY 14, 1969

# Page 614
⟨Page LM0614 182⟩
# Page 615
⟨Page LM0615 183⟩
# Page 616
⟨Page LM0616 184⟩
# Page 617
⟨Page LM0617 186⟩
```

This code is written to file `src/Luminary099/P30--P37.agc`.

2.25 P32-P35 P72-P75

751 ⟨*src/Luminary099/P32-P35-P72-P75.agc* 751⟩≡

```
# Copyright:    Public domain.
# Filename:     P32-P35_P72-P75.agc
# Purpose:      Part of the source code for Luminary 1A build 099.
#               It is part of the source code for the Lunar Module's (LM)
#               Apollo Guidance Computer (AGC), for Apollo 11.
# Assembler:    yaYUL
# Contact:      Ron Burkey <info@sandroid.org>.
# Website:      www.ibiblio.org/apollo.
# Pages:        618-650
# Mod history:  2009-05-18 RSB  Adapted from the Luminary 131 file of the
#                               same name, as corrected from Luminary 099
#                               page images.
#
# This source code has been transcribed or otherwise adapted from
# digitized images of a hardcopy from the MIT Museum.  The digitization
# was performed by Paul Fjeld, and arranged for by Deborah Douglas of
# the Museum.  Many thanks to both.  The images (with suitable reduction
# in storage size and consequent reduction in image quality as well) are
# available online at www.ibiblio.org/apollo.  If for some reason you
# find that the images are illegible, contact me at info@sandroid.org
# about getting access to the (much) higher-quality images which Paul
# actually created.
#
# Notations on the hardcopy document read, in part:
#
#       Assemble revision 001 of AGC program LMY99 by NASA 2021112-61
#       16:27 JULY 14, 1969

# Page 618
⟨Page LM0618 188⟩
# Page 619
⟨Page LM0619 190⟩
# Page 620
⟨Page LM0620 192⟩
# Page 621
⟨Page LM0621 194⟩
# Page 622
⟨Page LM0622 196⟩
# Page 623
⟨Page LM0623 197⟩
# Page 624
⟨Page LM0624 198⟩
# Page 625
```

```
⟨Page LM0625 200⟩
# Page 626
⟨Page LM0626 202⟩
# Page 627
⟨Page LM0627 204⟩
# Page 628
⟨Page LM0628 206⟩
# Page 629
⟨Page LM0629 207⟩
# Page 630
⟨Page LM0630 208⟩
# Page 631
⟨Page LM0631 209⟩
# Page 632
⟨Page LM0632 210⟩
# Page 633
⟨Page LM0633 212⟩
# Page 634
⟨Page LM0634 214⟩
# Page 635
⟨Page LM0635 216⟩
# Page 636
⟨Page LM0636 218⟩
# Page 637
⟨Page LM0637 220⟩
# Page 638
⟨Page LM0638 222⟩
# Page 639
⟨Page LM0639 224⟩
# Page 640
⟨Page LM0640 226⟩
# Page 641
⟨Page LM0641 228⟩
# Page 642
⟨Page LM0642 229⟩
# Page 643
⟨Page LM0643 230a⟩
# Page 644
⟨Page LM0644 230b⟩
# Page 645
⟨Page LM0645 231a⟩
# Page 646
⟨Page LM0646 231b⟩
# Page 647
⟨Page LM0647 232⟩
# Page 648
```

⟨*Page LM0648* 234⟩
```
# Page 649
```
⟨*Page LM0649* 235a⟩
```
# Page 650
```
⟨*Page LM0650* 235b⟩

This code is written to file `src/Luminary099/P32-P35-P72-P75.agc`.

2.26 P70-P71

754 ⟨*src/Luminary099/P70-P71.agc* 754⟩≡

```
# Copyright:    Public domain.
# Filename:     P70-P71.agc
# Purpose:      Part of the source code for Luminary 1A build 099.
#               It is part of the source code for the Lunar Module's (LM)
#               Apollo Guidance Computer (AGC), for Apollo 11.
# Assembler:    yaYUL
# Contact:      Hartmuth Gutsche <hgutsche@xplornet.com>.
# Website:      www.ibiblio.org/apollo.
# Pages:        829-837
# Mod history:  2009-05-23 HG   Transcribed from page images.
#               2009-06-05 RSB  Fixed a typo.
#
# This source code has been transcribed or otherwise adapted from
# digitized images of a hardcopy from the MIT Museum.  The digitization
# was performed by Paul Fjeld, and arranged for by Deborah Douglas of
# the Museum.  Many thanks to both.  The images (with suitable reduction
# in storage size and consequent reduction in image quality as well) are
# available online at www.ibiblio.org/apollo.  If for some reason you
# find that the images are illegible, contact me at info@sandroid.org
# about getting access to the (much) higher-quality images which Paul
# actually created.
#
# Notations on the hardcopy document read, in part:
#
#       Assemble revision 001 of AGC program LMY99 by NASA 2021112-61
#       16:27 JULY 14, 1969

# Page 829
⟨Page LM0829 349⟩
# Page 830
⟨Page LM0830 351⟩
# Page 831
⟨Page LM0831 353⟩
# Page 832
⟨Page LM0832 355⟩
# Page 833
⟨Page LM0833 357⟩
# Page 834
⟨Page LM0834 359⟩
# Page 835
⟨Page LM0835 361⟩
# Page 836
⟨Page LM0836 363⟩
```

```
# Page 837
```

⟨*Page LM0837* 364a⟩

This code is written to file `src/Luminary099/P70-P71.agc`.

2.27 P-AXIS RCS AUTOPILOT

756 ⟨*src/Luminary099/P-AXIS-RCS-AUTOPILOT.agc* 756⟩≡

```
# Copyright:    Public domain.
# Filename:     P-AXIS_RCS_AUTOPILOT.agc
# Purpose:      Part of the source code for Luminary 1A build 099.
#               It is part of the source code for the Lunar Module's (LM)
#               Apollo Guidance Computer (AGC), for Apollo 11.
# Assembler:    yaYUL
# Contact:      Ron Burkey <info@sandroid.org>.
# Website:      www.ibiblio.org/apollo.
# Pages:        1421-1441
# Mod history:  2009-05-27 RSB  Adapted from the corresponding
#                               Luminary131 file, using page
#                               images from Luminary 1A.
#               2009-06-05 RSB  Corrected a relative jump from
#                               +8 to +8D.
#               2009-06-07 RSB  Corrected a typo.
#
# This source code has been transcribed or otherwise adapted from
# digitized images of a hardcopy from the MIT Museum.  The digitization
# was performed by Paul Fjeld, and arranged for by Deborah Douglas of
# the Museum.  Many thanks to both.  The images (with suitable reduction
# in storage size and consequent reduction in image quality as well) are
# available online at www.ibiblio.org/apollo.  If for some reason you
# find that the images are illegible, contact me at info@sandroid.org
# about getting access to the (much) higher-quality images which Paul
# actually created.
#
# Notations on the hardcopy document read, in part:
#
#       Assemble revision 001 of AGC program LMY99 by NASA 2021112-61
#       16:27 JULY 14, 1969

# Page 1421
⟨Page LM1421 558⟩
# Page 1422
⟨Page LM1422 560⟩
# Page 1423
⟨Page LM1423 562⟩
# Page 1424
⟨Page LM1424 564⟩
# Page 1425
⟨Page LM1425 566⟩
# Page 1426
⟨Page LM1426 568⟩
```

```
# Page 1427
⟨Page LM1427 570⟩
# Page 1428
⟨Page LM1428 572⟩
# Page 1429
⟨Page LM1429 574⟩
# Page 1430
⟨Page LM1430 576⟩
# Page 1431
⟨Page LM1431 578⟩
# Page 1432
⟨Page LM1432 580⟩
# Page 1433
⟨Page LM1433 582⟩
# Page 1434
⟨Page LM1434 584⟩
# Page 1435
⟨Page LM1435 586⟩
# Page 1436
⟨Page LM1436 588⟩
# Page 1437
⟨Page LM1437 590⟩
# Page 1438
⟨Page LM1438 592⟩
# Page 1439
⟨Page LM1439 593⟩
# Page 1440
⟨Page LM1440 594⟩
# Page 1441
⟨Page LM1441 595⟩
```

This code is written to file `src/Luminary099/P-AXIS-RCS-AUTOPILOT.agc`.

2.28 Q R-AXIS RCS AUTOPILOT

758 ⟨*src/Luminary099/Q-R-AXIS-RCS-AUTOPILOT.agc* 758⟩≡
```
# Copyright:    Public domain.
# Filename:     Q_R-AXIS_RCS_AUTOPILOT.agc
# Purpose:      Part of the source code for Luminary 1A build 099.
#               It is part of the source code for the Lunar Module's (LM)
#               Apollo Guidance Computer (AGC), for Apollo 11.
# Assembler:    yaYUL
# Contact:      Ron Burkey <info@sandroid.org>.
# Website:      www.ibiblio.org/apollo.
# Pages:        1442-1459
# Mod history:  2009-05-27 RSB  Adapted from the corresponding
#                               Luminary131 file, using page
#                               images from Luminary 1A.
#               2009-06-07 RSB  Corrected "DEC 96.0" to "DEC 96", since
#                               the former is not compatible with yaYUL.
#
# This source code has been transcribed or otherwise adapted from
# digitized images of a hardcopy from the MIT Museum.  The digitization
# was performed by Paul Fjeld, and arranged for by Deborah Douglas of
# the Museum.  Many thanks to both.  The images (with suitable reduction
# in storage size and consequent reduction in image quality as well) are
# available online at www.ibiblio.org/apollo.  If for some reason you
# find that the images are illegible, contact me at info@sandroid.org
# about getting access to the (much) higher-quality images which Paul
# actually created.
#
# Notations on the hardcopy document read, in part:
#
#       Assemble revision 001 of AGC program LMY99 by NASA 2021112-61
#       16:27 JULY 14, 1969

# Page 1442
⟨Page LM1442 597⟩
# Page 1443
⟨Page LM1443 599⟩
# Page 1444
⟨Page LM1444 601⟩
# Page 1445
⟨Page LM1445 603⟩
# Page 1446
⟨Page LM1446 605⟩
# Page 1447
⟨Page LM1447 607⟩
# Page 1448
```

⟨*Page LM1448* 609⟩
```
# Page 1449
```
⟨*Page LM1449* 611⟩
```
# Page 1450
```
⟨*Page LM1450* 612⟩
```
# Page 1451
```
⟨*Page LM1451* 613⟩
```
# Page 1452
```
⟨*Page LM1452* 614⟩
```
# Page 1453
```
⟨*Page LM1453* 616⟩
```
# Page 1454
```
⟨*Page LM1454* 618⟩
```
# Page 1455
```
⟨*Page LM1455* 620⟩
```
# Page 1456
```
⟨*Page LM1456* 622⟩
```
# Page 1457
```
⟨*Page LM1457* 624⟩
```
# Page 1458
```
⟨*Page LM1458* 626⟩
```
# Page 1459
```
⟨*Page LM1459* 627a⟩

This code is written to file `src/Luminary099/Q-R-AXIS-RCS-AUTOPILOT.agc`.

2.29 R63

760 ⟨*src/Luminary099/R63.agc* 760⟩≡

```
# Copyright:    Public domain.
# Filename:     R63.agc
# Purpose:      Part of the source code for Luminary 1A build 099.
#               It is part of the source code for the Lunar Module's (LM)
#               Apollo Guidance Computer (AGC), for Apollo 11.
# Assembler:    yaYUL
# Contact:      Ron Burkey <info@sandroid.org>.
# Website:      www.ibiblio.org/apollo.
# Pages:        338-341
# Mod history:  2009-05-16 RSB  Adapted from the corresponding
#                               Luminary131 file, using page
#                               images from Luminary 1A.
#
# This source code has been transcribed or otherwise adapted from
# digitized images of a hardcopy from the MIT Museum.  The digitization
# was performed by Paul Fjeld, and arranged for by Deborah Douglas of
# the Museum.  Many thanks to both.  The images (with suitable reduction
# in storage size and consequent reduction in image quality as well) are
# available online at www.ibiblio.org/apollo.  If for some reason you
# find that the images are illegible, contact me at info@sandroid.org
# about getting access to the (much) higher-quality images which Paul
# actually created.
#
# Notations on the hardcopy document read, in part:
#
#       Assemble revision 001 of AGC program LMY99 by NASA 2021112-61
#       16:27 JULY 14, 1969

# Page 338
⟨Page LM0338 108⟩
# Page 339
⟨Page LM0339 110⟩
# Page 340
⟨Page LM0340 111a⟩
# Page 341
⟨Page LM0341 111b⟩
```

This code is written to file `src/Luminary099/R63.agc`.

2.30 RADAR LEADIN ROUTINES

761 ⟨*src/Luminary099/RADAR-LEADIN-ROUTINES.agc* 761⟩≡

```
# Copyright:    Public domain.
# Filename:     RADAR_LEADIN_ROUTINES.agc
# Purpose:      Part of the source code for Luminary 1A build 099.
#               It is part of the source code for the Lunar Module's (LM)
#               Apollo Guidance Computer (AGC), for Apollo 11.
# Assembler:    yaYUL
# Contact:      Ron Burkey <info@sandroid.org>.
# Website:      www.ibiblio.org/apollo.
# Pages:        490-491
# Mod history:  2009-05-17 RSB  Adapted from the corresponding
#                               Luminary131 file, using page
#                               images from Luminary 1A.
#
# This source code has been transcribed or otherwise adapted from
# digitized images of a hardcopy from the MIT Museum.  The digitization
# was performed by Paul Fjeld, and arranged for by Deborah Douglas of
# the Museum.  Many thanks to both.  The images (with suitable reduction
# in storage size and consequent reduction in image quality as well) are
# available online at www.ibiblio.org/apollo.  If for some reason you
# find that the images are illegible, contact me at info@sandroid.org
# about getting access to the (much) higher-quality images which Paul
# actually created.
#
# Notations on the hardcopy document read, in part:
#
#       Assemble revision 001 of AGC program LMY99 by NASA 2021112-61
#       16:27 JULY 14, 1969

# Page 490
```
⟨*Page LM0490* 180⟩
```
# Page 491
```
⟨*Page LM0491* 181a⟩

This code is written to file `src/Luminary099/RADAR-LEADIN-ROUTINES.agc`.

2.31 RCS FAILURE MONITOR

762 ⟨*src/Luminary099/RCS-FAILURE-MONITOR.agc* 762⟩≡

```
# Copyright:    Public domain.
# Filename:     RCS_FAILURE_MONITOR.agc
# Purpose:      Part of the source code for Luminary 1A build 099.
#               It is part of the source code for the Lunar Module's (LM)
#               Apollo Guidance Computer (AGC), for Apollo 11.
# Assembler:    yaYUL
# Contact:      Hartmuth Gutsche <hgutsche@xplornet.com>.
# Website:      www.ibiblio.org/apollo.
# Pages:        190-192
# Mod history:  2009-05-19 HG   Transcribed from page images.
#
# This source code has been transcribed or otherwise adapted from
# digitized images of a hardcopy from the MIT Museum.  The digitization
# was performed by Paul Fjeld, and arranged for by Deborah Douglas of
# the Museum.  Many thanks to both.  The images (with suitable reduction
# in storage size and consequent reduction in image quality as well) are
# available online at www.ibiblio.org/apollo.  If for some reason you
# find that the images are illegible, contact me at info@sandroid.org
# about getting access to the (much) higher-quality images which Paul
# actually created.
#
# Notations on the hardcopy document read, in part:
#
#       Assemble revision 001 of AGC program LMY99 by NASA 2021112-61
#       16:27 JULY 14, 1969

# Page 190
⟨Page LM0190 65⟩
# Page 191
⟨Page LM0191 67⟩
# Page 192
⟨Page LM0192 69⟩
```

This code is written to file `src/Luminary099/RCS-FAILURE-MONITOR.agc`.

2.32 RTB OP CODES

763 ⟨*src/Luminary099/RTB-OP-CODES.agc* 763⟩≡

```
# Copyright:    Public domain.
# Filename:     RTB_OP_CODES.agc
# Purpose:      Part of the source code for Luminary 1A build 099.
#               It is part of the source code for the Lunar Module's (LM)
#               Apollo Guidance Computer (AGC), for Apollo 11.
# Assembler:    yaYUL
# Contact:      Ron Burkey <info@sandroid.org>.
# Website:      www.ibiblio.org/apollo.
# Pages:        1397-1401
# Mod history:  2009-05-10 SN   (Sergio Navarro).  Started adapting
#                               from the Luminary131/ file of the same
#                               name, using Luminary099 page images.
#
# This source code has been transcribed or otherwise adapted from
# digitized images of a hardcopy from the MIT Museum.  The digitization
# was performed by Paul Fjeld, and arranged for by Deborah Douglas of
# the Museum.  Many thanks to both.  The images (with suitable reduction
# in storage size and consequent reduction in image quality as well) are
# available online at www.ibiblio.org/apollo.  If for some reason you
# find that the images are illegible, contact me at info@sandroid.org
# about getting access to the (much) higher-quality images which Paul
# actually created.
#
# Notations on the hardcopy document read, in part:
#
#       Assemble revision 001 of AGC program LMY99 by NASA 2021112-61
#       16:27 JULY 14, 1969

# Page 1397
⟨Page LM1397 528⟩
# Page 1398
⟨Page LM1398 530⟩
# Page 1399
⟨Page LM1399 531a⟩
# Page 1400
⟨Page LM1400 531b⟩
# Page 1401
⟨Page LM1401 532⟩
# Page 1402
⟨Page LM1402 533a⟩
```

This code is written to file `src/Luminary099/RTB-OP-CODES.agc`.

2.33 S-BAND ANTENNA FOR LM

764 ⟨*src/Luminary099/S-BAND-ANTENNA-FOR-LM.agc* 764⟩≡

```
# Copyright:    Public domain.
# Filename:     S-BAND_ANTENNA_FOR_LM.agc
# Purpose:      Part of the source code for Luminary 1A build 099.
#               It is part of the source code for the Lunar Module's (LM)
#               Apollo Guidance Computer (AGC), for Apollo 11.
# Assembler:    yaYUL
# Contact:      Ron Burkey <info@sandroid.org>.
# Website:      www.ibiblio.org/apollo.
# Pages:        486-489
# Mod history:  2009-05-17 RSB  Adapted from the corresponding
#                               Luminary131 file, using page
#                               images from Luminary 1A.
#               2009-06-07 RSB  Corrected a misprint.
#
# This source code has been transcribed or otherwise adapted from
# digitized images of a hardcopy from the MIT Museum.  The digitization
# was performed by Paul Fjeld, and arranged for by Deborah Douglas of
# the Museum.  Many thanks to both.  The images (with suitable reduction
# in storage size and consequent reduction in image quality as well) are
# available online at www.ibiblio.org/apollo.  If for some reason you
# find that the images are illegible, contact me at info@sandroid.org
# about getting access to the (much) higher-quality images which Paul
# actually created.
#
# Notations on the hardcopy document read, in part:
#
#       Assemble revision 001 of AGC program LMY99 by NASA 2021112-61
#       16:27 JULY 14, 1969

# Page 486
⟨Page LM0486 174⟩
# Page 487
⟨Page LM0487 176⟩
# Page 488
⟨Page LM0488 178⟩
# Page 489
⟨Page LM0489 179a⟩
```

This code is written to file `src/Luminary099/S-BAND-ANTENNA-FOR-LM.agc`.

2.34 SERVICER

765 ⟨*src/Luminary099/SERVICER.agc* 765⟩≡

```
# Copyright:    Public domain.
# Filename:     SERVICER.agc
# Purpose:      Part of the source code for Luminary, build 099. It
#               is part of the source code for the Lunar Module's
#               (LM) Apollo Guidance Computer (AGC), Apollo 11.
# Assembler:    yaYUL
# Reference:    pp. 857-897
# Contact:      Ron Burkey <info@sandroid.org>,
#               Fabrizio Bernardini <fabrizio@spacecraft.it>
# Website:      http://www.ibiblio.org/apollo.
# Mod history:  2009-06-01 FB   Transcription Batch 4 Assignment.
#               2009-06-05 RSB  Fixed a couple of typos, plus a goofy relative
#                               label reference from the original source.
#
# The contents of the "Luminary099" files, in general, are transcribed
# from scanned documents.
#
#       Assemble revision 001 of AGC program Luminary099 by NASA
#       2021112-061.  July 14, 1969.
#
#       Prepared by
#                       Massachusetts Institute of Technology
#                       75 Cambridge Parkway
#                       Cambridge, Massachusetts
#
#       under NASA contract NAS 9-4065.
#
# Refer directly to the online document mentioned above for further
# information.  Please report any errors to info@sandroid.org.

# Page 857
⟨Page LM0857 397⟩
# Page 858
⟨Page LM0858 398⟩
# Page 859
⟨Page LM0859 400⟩
# Page 860
⟨Page LM0860 401⟩
# Page 861
⟨Page LM0861 403⟩
# Page 862
⟨Page LM0862 405⟩
# Page 863
```

⟨*Page LM0863* 407⟩
Page 864
⟨*Page LM0864* 408⟩
Page 865
⟨*Page LM0865* 409⟩
Page 866
⟨*Page LM0866* 410⟩
Page 867
⟨*Page LM0867* 411⟩
Page 868
⟨*Page LM0868* 412a⟩
Page 869
⟨*Page LM0869* 412b⟩
Page 870
⟨*Page LM0870* 413⟩
Page 871
⟨*Page LM0871* 415⟩
Page 872
⟨*Page LM0872* 416⟩
Page 873
⟨*Page LM0873* 417⟩
Page 874
⟨*Page LM0874* 418⟩
Page 875
⟨*Page LM0875* 419⟩
Page 876
⟨*Page LM0876* 420⟩
Page 877
⟨*Page LM0877* 421⟩
Page 878
⟨*Page LM0878* 422⟩
Page 879
⟨*Page LM0879* 423⟩
Page 880
⟨*Page LM0880* 424⟩
Page 881
⟨*Page LM0881* 425⟩
Page 882
⟨*Page LM0882* 426⟩
Page 883
⟨*Page LM0883* 428⟩
Page 884
⟨*Page LM0884* 429⟩
Page 885
⟨*Page LM0885* 431⟩
Page 886

⟨*Page LM0886* 433⟩
```
# Page 887
```
⟨*Page LM0887* 435⟩
```
# Page 888
```
⟨*Page LM0888* 437⟩
```
# Page 889
```
⟨*Page LM0889* 439⟩
```
# Page 890
```
⟨*Page LM0890* 441⟩
```
# Page 891
```
⟨*Page LM0891* 442⟩
```
# Page 892
```
⟨*Page LM0892* 443⟩
```
# Page 893
```
⟨*Page LM0893* 445⟩
```
# Page 894
```
⟨*Page LM0894* 447⟩
```
# Page 895
```
⟨*Page LM0895* 449⟩
```
# Page 896
```
⟨*Page LM0896* 451⟩
```
# Page 897
```
⟨*Page LM0897* 452a⟩

This code is written to file `src/Luminary099/SERVICER.agc`.

2.35 SPS BACK-UP RCS CONTROL

768 ⟨*src/Luminary099/SPS-BACK-UP-RCS-CONTROL.agc* 768⟩≡

```
# Copyright:    Public domain.
# Filename:     SPS_BACK-UP_RCS_CONTROL.agc
# Purpose:      Part of the source code for Luminary 1A build 099.
#               It is part of the source code for the Lunar Module's (LM)
#               Apollo Guidance Computer (AGC), for Apollo 11.
# Assembler:    yaYUL
# Contact:      Ron Burkey <info@sandroid.org>.
# Website:      www.ibiblio.org/apollo.
# Pages:        1507-1510
# Mod history:  2009-05-27 RSB  Adapted from the corresponding
#                               Luminary131 file, using page
#                               images from Luminary 1A.
#
# This source code has been transcribed or otherwise adapted from
# digitized images of a hardcopy from the MIT Museum.  The digitization
# was performed by Paul Fjeld, and arranged for by Deborah Douglas of
# the Museum.  Many thanks to both.  The images (with suitable reduction
# in storage size and consequent reduction in image quality as well) are
# available online at www.ibiblio.org/apollo.  If for some reason you
# find that the images are illegible, contact me at info@sandroid.org
# about getting access to the (much) higher-quality images which Paul
# actually created.
#
# Notations on the hardcopy document read, in part:
#
#       Assemble revision 001 of AGC program LMY99 by NASA 2021112-61
#       16:27 JULY 14, 1969

# Page 1507
⟨Page LM1507 706⟩
# Page 1508
⟨Page LM1508 707⟩
# Page 1509
⟨Page LM1509 709⟩
# Page 1510
⟨Page LM1510 710⟩
```

This code is written to file `src/Luminary099/SPS-BACK-UP-RCS-CONTROL.agc`.

2.36 T6-RUPT PROGRAMS

769 ⟨*src/Luminary099/T6-RUPT-PROGRAMS.agc* 769⟩≡

```
# Copyright:    Public domain.
# Filename:     T6-RUPT_PROGRAMS.agc
# Purpose:      Part of the source code for Luminary 1A build 099.
#               It is part of the source code for the Lunar Module's (LM)
#               Apollo Guidance Computer (AGC), for Apollo 11.
# Assembler:    yaYUL
# Contact:      Ron Burkey <info@sandroid.org>.
# Website:      www.ibiblio.org/apollo.
# Pages:        1403-1405
# Mod history:  2009-05-10 SN   (Sergio Navarro).  Started adapting
#                               from the Luminary131/ file of the same
#                               name, using Luminary099 page images.
#
# This source code has been transcribed or otherwise adapted from
# digitized images of a hardcopy from the MIT Museum.  The digitization
# was performed by Paul Fjeld, and arranged for by Deborah Douglas of
# the Museum.  Many thanks to both.  The images (with suitable reduction
# in storage size and consequent reduction in image quality as well) are
# available online at www.ibiblio.org/apollo.  If for some reason you
# find that the images are illegible, contact me at info@sandroid.org
# about getting access to the (much) higher-quality images which Paul
# actually created.
#
# Notations on the hardcopy document read, in part:
#
#       Assemble revision 001 of AGC program LMY99 by NASA 2021112-61
#       16:27 JULY 14, 1969

# Page 1403
⟨Page LM1403 534⟩
# Page 1404
⟨Page LM1404 536⟩
# Page 1405
⟨Page LM1405 537⟩
```

This code is written to file `src/Luminary099/T6-RUPT-PROGRAMS.agc`.

2.37 THE LUNAR LANDING

770 ⟨*src/Luminary099/THE-LUNAR-LANDING.agc* 770⟩≡

```
# Copyright:    Public domain.
# Filename:     THE_LUNAR_LANDING.agc
# Purpose:      Part of the source code for Luminary 1A build 099.
#               It is part of the source code for the Lunar Module's (LM)
#               Apollo Guidance Computer (AGC), for Apollo 11.
# Assembler:    yaYUL
# Contact:      Hartmuth Gutsche<hgutsche@xplornet.com>.
# Website:      www.ibiblio.org/apollo.
# Pages:        785-792
# Mod history:  2009-05-20 HG   Transcribed from page images.
#
# This source code has been transcribed or otherwise adapted from
# digitized images of a hardcopy from the MIT Museum.  The digitization
# was performed by Paul Fjeld, and arranged for by Deborah Douglas of
# the Museum.  Many thanks to both.  The images (with suitable reduction
# in storage size and consequent reduction in image quality as well) are
# available online at www.ibiblio.org/apollo.  If for some reason you
# find that the images are illegible, contact me at info@sandroid.org
# about getting access to the (much) higher-quality images which Paul
# actually created.
#
# Notations on the hardcopy document read, in part:
#
#       Assemble revision 001 of AGC program LMY99 by NASA 2021112-61
#       16:27 JULY 14, 1969

# Page 785
⟨Page LM0785 279⟩
# Page 786
⟨Page LM0786 281⟩
# Page 787 new page is actually one line earlier but this would put the indices on a
⟨Page LM0787 283⟩
# Page 788
⟨Page LM0788 285⟩
# Page 789
⟨Page LM0789 287a⟩
# Page 790
⟨Page LM0790 287b⟩
# Page 791
⟨Page LM0791 288⟩
# Page 792
⟨Page LM0792 289a⟩
```

This code is written to file `src/Luminary099/THE-LUNAR-LANDING.agc`.

2.38 THROTTLE CONTROL ROUTINES

771 ⟨*src/Luminary099/THROTTLE-CONTROL-ROUTINES.agc* 771⟩≡

```
# Copyright:    Public domain.
# Filename:     THROTTLE_CONTROL_ROUTINES.agc
# Purpose:      Part of the source code for Luminary 1A build 099.
#               It is part of the source code for the Lunar Module's (LM)
#               Apollo Guidance Computer (AGC), for Apollo 11.
# Assembler:    yaYUL
# Contact:      HARTMUTH GUTSCHE <hgutsche@xplornet.com>.
# Website:      www.ibiblio.org/apollo.
# Pages:        793-797
# Mod history:  2009-05-20 HG   Transcribed from page images.
#
# This source code has been transcribed or otherwise adapted from
# digitized images of a hardcopy from the MIT Museum.  The digitization
# was performed by Paul Fjeld, and arranged for by Deborah Douglas of
# the Museum.  Many thanks to both.  The images (with suitable reduction
# in storage size and consequent reduction in image quality as well) are
# available online at www.ibiblio.org/apollo.  If for some reason you
# find that the images are illegible, contact me at info@sandroid.org
# about getting access to the (much) higher-quality images which Paul
# actually created.
#
# Notations on the hardcopy document read, in part:
#
#       Assemble revision 001 of AGC program LMY99 by NASA 2021112-61
#       16:27 JULY 14, 1969

# Page 793
⟨Page LM0793 290⟩
# Page 794
⟨Page LM0794 291⟩
# Page 795
⟨Page LM0795 293⟩
# Page 796
⟨Page LM0796 295⟩
# Page 797
⟨Page LM0797 296a⟩
```

This code is written to file `src/Luminary099/THROTTLE-CONTROL-ROUTINES.agc`.

2.39 TJET LAW

772 ⟨*src/Luminary099/TJET-LAW.agc* 772⟩≡
```
# Copyright:    Public domain.
# Filename:     TJET_LAW.agc
# Purpose:      Part of the source code for Luminary 1A build 099.
#               It is part of the source code for the Lunar Module's (LM)
#               Apollo Guidance Computer (AGC), for Apollo 11.
# Assembler:    yaYUL
# Contact:      Ron Burkey <info@sandroid.org>.
# Website:      www.ibiblio.org/apollo.
# Pages:        1460-1469
# Mod history:  2009-05-27 RSB  Adapted from the corresponding
#                               Luminary131 file, using page
#                               images from Luminary 1A.
#               2009-06-06 RSB  Eliminated a stray instruction that had crept
#                               in somehow.
#
# This source code has been transcribed or otherwise adapted from
# digitized images of a hardcopy from the MIT Museum.  The digitization
# was performed by Paul Fjeld, and arranged for by Deborah Douglas of
# the Museum.  Many thanks to both.  The images (with suitable reduction
# in storage size and consequent reduction in image quality as well) are
# available online at www.ibiblio.org/apollo.  If for some reason you
# find that the images are illegible, contact me at info@sandroid.org
# about getting access to the (much) higher-quality images which Paul
# actually created.
#
# Notations on the hardcopy document read, in part:
#
#       Assemble revision 001 of AGC program LMY99 by NASA 2021112-61
#       16:27 JULY 14, 1969

# Page 1460
⟨Page LM1460 628⟩
# Page 1461
⟨Page LM1461 630⟩
# Page 1462
⟨Page LM1462 632⟩
# Page 1463
⟨Page LM1463 634⟩
# Page 1464
⟨Page LM1464 636⟩
# Page 1465
⟨Page LM1465 638⟩
# Page 1466
```

⟨*Page LM1466* 640⟩
```
# Page 1467
```
⟨*Page LM1467* 641⟩
```
# Page 1468
```
⟨*Page LM1468* 642⟩
```
# Page 1469
```
⟨*Page LM1469* 644⟩

This code is written to file `src/Luminary099/TJET-LAW.agc`.

2.40 TRIM GIMBAL CNTROL SYSTEM

774 ⟨*src/Luminary099/TRIM-GIMBAL-CNTROL-SYSTEM.agc* 774⟩≡

```
# Copyright:    Public domain.
# Filename:     TRIM_GIMBAL_CNTROL_SYSTEM.agc
# Purpose:      Part of the source code for Luminary 1A build 099.
#               It is part of the source code for the Lunar Module's (LM)
#               Apollo Guidance Computer (AGC), for Apollo 11.
# Assembler:    yaYUL
# Contact:      Ron Burkey <info@sandroid.org>.
# Website:      www.ibiblio.org/apollo.
# Pages:        1472-1485
# Mod history:  2009-05-27 RSB  Adapted from the corresponding
#                               Luminary131 file, using page
#                               images from Luminary 1A.
#
# This source code has been transcribed or otherwise adapted from
# digitized images of a hardcopy from the MIT Museum.  The digitization
# was performed by Paul Fjeld, and arranged for by Deborah Douglas of
# the Museum.  Many thanks to both.  The images (with suitable reduction
# in storage size and consequent reduction in image quality as well) are
# available online at www.ibiblio.org/apollo.  If for some reason you
# find that the images are illegible, contact me at info@sandroid.org
# about getting access to the (much) higher-quality images which Paul
# actually created.
#
# Notations on the hardcopy document read, in part:
#
#       Assemble revision 001 of AGC program LMY99 by NASA 2021112-61
#       16:27 JULY 14, 1969

# Page 1472
⟨Page LM1472 648⟩
# Page 1473
⟨Page LM1473 649⟩
# Page 1474
⟨Page LM1474 651⟩
# Page 1475
⟨Page LM1475 653⟩
# Page 1476
⟨Page LM1476 655⟩
# Page 1477
⟨Page LM1477 656⟩
# Page 1478
⟨Page LM1478 657⟩
# Page 1479
```

⟨*Page LM1479* 659⟩
```
# Page 1480
```
⟨*Page LM1480* 660⟩
```
# Page 1481
```
⟨*Page LM1481* 661⟩
```
# Page 1482
```
⟨*Page LM1482* 663⟩
```
# Page 1483
```
⟨*Page LM1483* 665⟩
```
# Page 1484
```
⟨*Page LM1484* 666⟩

This code is written to file `src/Luminary099/TRIM-GIMBAL-CNTROL-SYSTEM.agc`.

Chapter 3

Notes, Bibliography and Indexes

3.1 Chunks

⟨*Page LM0819* 335⟩
⟨*Page LM0820* 337⟩
⟨*Page LM0821* 339⟩
⟨*Page LM0822* 340⟩
⟨*Page LM0823* 341⟩
⟨*Page LM0824* 342⟩
⟨*Page LM0825* 344⟩
⟨*Page LM0826* 346⟩
⟨*Page LM0827* 347a⟩
⟨*Page LM0828* 347b⟩
⟨*Page LM0829* 349⟩
⟨*Page LM0830* 351⟩
⟨*Page LM0831* 353⟩
⟨*Page LM0832* 355⟩
⟨*Page LM0833* 357⟩
⟨*Page LM0834* 359⟩
⟨*Page LM0835* 361⟩
⟨*Page LM0836* 363⟩
⟨*Page LM0837* 364a⟩
⟨*Page LM0838* 365⟩
⟨*Page LM0839* 367⟩
⟨*Page LM0840* 369⟩
⟨*Page LM0841* 371⟩
⟨*Page LM0842* 372a⟩
⟨*Page LM0843* 373⟩
⟨*Page LM0844* 374⟩
⟨*Page LM0845* 375⟩
⟨*Page LM0846* 377⟩
⟨*Page LM0847* 379⟩
⟨*Page LM0848* 381⟩
⟨*Page LM0849* 383⟩
⟨*Page LM0850* 385⟩
⟨*Page LM0851* 387⟩
⟨*Page LM0852* 389⟩
⟨*Page LM0853* 391⟩
⟨*Page LM0854* 393⟩
⟨*Page LM0855* 394⟩
⟨*Page LM0856* 395⟩
⟨*Page LM0857* 397⟩
⟨*Page LM0858* 398⟩
⟨*Page LM0859* 400⟩
⟨*Page LM0860* 401⟩
⟨*Page LM0861* 403⟩
⟨*Page LM0862* 405⟩
⟨*Page LM0863* 407⟩
⟨*Page LM0864* 408⟩

3.2 Index