Bulletin No. 672-21-EAS-208

preliminary

USER'S GUIDE FOR AGS
BIT-BY-BIT SIMULATOR

August 1968

Prepared Under Contract NAS 9-5384 for
NATIONAL AERONAUTICS AND SPACE ADMINISTRATION
MANNED SPACECRAFT CENTER
HOUSTON, TEXAS 77058

Prepared by K. B. Robertson

Approved by: J. E. Earl
J. E. Earl, Project Leader
AGS Verification Testing
LOCKHEED ELECTRONICS CO.

LOCKHEED ELECTRONICS COMPANY
HOUSTON AEROSPACE SYSTEMS DIVISION
HOUSTON, TEXAS 77058

NASA-MSC APPROVALS

Approved by: ______________________________________

C. F. Lively, Jr.
AGS Software Verification Manager
Systems Analysis Branch
Guidance and Control Division

Approved by: ______________________________________

A. W. Hambleton
Guidance Applications Section
Flight Mechanics Applications Branch

Approved by: ______________________________________

M. T. Cunningham, Chief
Flight Mechanics Applications Branch
Computation and Analysis Division

ACKNOWLEDGMENT

The following persons contributed significantly to the contents of this document:

J. K. McClatchy

J. A. Smith

M. E. Thompson

R. E. Tyler

CONTENTS

CONTENTS (continued)

1.0 INTRODUCTION

This user's guide defines the preparation and procedure required to simulate a Lunar Module flight using the Abort Guidance System simulator on the CDC-3800 computer. It is assumed that the user has a sound knowledge of the flight he wishes to simulate and the Abort Guidance System.

The Abort Guidance System (AGS) simulator consists of three logically distinct sections:

- PREPROCESSOR
- MAIN PROCESSOR
- PLOT PROCESSOR

The simulator Preprocessor section adds great flexibility to the AGS simulator and allows the user almost unlimited input, output and control functions. For instance, through the use of the system processor's capability, the user may interface his own special purpose routines with all other simulator routines.

That part of the system which simulates the AGS is the MAIN Processor which includes an Executive system, Interpretive Computer Simulation (ICS) section and an Environments model. Executive system functions are interface controls, editing operations and a restart capability for the AGS simulator. In the ICS section the Abort Electronics Assembly (AEA) is modeled in a bit-by-bit fashion. The Environments model allows the user the choice of a "fast" control model or a slower, more realistic simulation of the LM control system. The latter includes various transfer functions, pulse ratio modulation, jet firing logic, etc.

An extremely flexible plotting program is provided for the simulator by the PLOT Processor. This processor permits the user to plot from any of the variables computed in the simulator system or from a standard set of output parameters.

2.0 OPERATING PROCEDURE

Two distinct modes of operation are available in the AGS simulator:

- Open Loop
- Closed Loop

In the Open Loop mode the AGS simulator operates without the Environments models. This mode is requested by setting an Executive input flag (See Section 4.2). In this mode the AGS input registers contain zero pulse levels and Astronaut interfaces are bypassed. The Open Loop form of simulation helps in checking out the AGS flight program; however, this limited form of simulation is generally sufficient only in the first phases of AGS program checkout.

The Closed Loop mode is used if interaction between the AEA and Environments is desired. This is the normal mode of operation of the simulator and therefore no separate initialization parameter is required.

2.1 Simulator Deck Set-Up

The following deck set-up must be used for operation of the AGS simulator. A complete description of the CDC (Control Data Corp.) control cards may be found in the 3600/3800 DRUM SCOPE Reference Manual. Those cards enclosed in brackets are required but will vary from run to run. Complete information on the generation of these cards may be found in the sections referenced. Cards enclosed in braces are optional but may be included if the corresponding options are desired.

$^{7}_{9}$JOB

$^{7}_{9}$DEMAND,40000B, 7MT

$^{7}_{9}$EQUIP

[ALL I/O EQUIP CARDS] (See Section 2.2)

$^{7}_{9}$LOAD, 10

$^{7}_{9}$RUN

ALL PREPROC INPUT CARDS
*START
7 7
8 8

(See Section 3.0)

$^{7}_{9}$RELEASE, 10

$^{7}_{9}$COMPASS, I = 3, X = 15

$^{7}_{9}$FTN,
ANY USER SUPPLIED SUBPROGRAMS
SCOPE
$^{7}_{9}$LOAD

(See Section 2.3)

11
0
$^{7}_{9}$BANK, (0),LC.,(1),CHKPNTW

$^{7}_{9}$LOAD,35

$^{7}_{9}$LOAD,15

$^{7}_{9}$RUN

ALL MAIN PROCESSOR INPUT CARDS
$

(See Section 4.0)

$^{7}_{9}$RELEASE,35

11
0
$^{7}_{9}$BANK,(1), /POINT/

$^{7}_{9}$LOAD,22

$^{7}_{9}$RUN,

ANY PREPROC PLOT DATA CARDS

(See Section 5.1)

$$\left.\begin{array}{l} {}^{7}_{9}\text{LOAD,45} \\ \\ {}^{7}_{9}\text{RUN,} \\ \\ \text{ANY STANDARD PLOT DATA CARDS} \end{array}\right\}$$ (See Section 5.2)

2.2 Simulator Logical Units

The tables below describe the input/output references to equipment in terms of the programmed logical units (L.U.). These units may be controlled by the simulator user with CDC control cards, as indicated by the corresponding codes.

INPUT UNITS - These tapes require release control cards as indicated by their codes:

L.U.	DESCRIPTION	CODE
10	PREPROC LOAD & GO	R1
35	MAIN PROC LOAD & GO	R2
1	AEA MEMORY INPUT	R2
21	UNIVERSE DATA INPUT	R2
22	PREPROC PLOT LOAD & GO	R3
45	STD. PLOT LOAD & GO	--

OUTPUT UNITS - These units require equip and release cards as indicated by their codes:

L.U.	DESCRIPTION	CODE
30	GSE TEMELETRY	E1,R2
31	CHECKPOINT DUMP TAPE	E2,R2
5	PREPROC CALCOMP	E3,R3
49	STD. PLOT CALCOMP	E3

. WORKING UNITS - These units are generally working units and are put on the drum (no equip cards required); however, they may be controlled differently, as indicated by their codes:

L.U.	DESCRIPTION	CODE
3	PREPROC ASSEMBLY	R1
15	PREPROC OUTPUT LOAD & GO	S1,R2
4	PREPROC PLOT DUMPS	S2,R3
40	STD. PLOT DUMPS	S2
41	STD. PLOT WORK AREA	--
2	PREPROC WORK AREA	R2
6	PREPROC WORK AREA	R2

An explanation for the various codes listed above is as follows:

R1 - This unit may be released after PREPROC execution.

R2 - This unit may be released after MAIN PROC execution.

R3 - This unit may be released after PREPROC PLOT execution.

E1 - This unit may be equipped as a save tape or put on the drum if so desired.

E2 - This tape must be equipped as a save tape or a working magnetic tape unit.

E3 - This unit must be equipped as a save tape if this option was requested and density set at LO.

S1 - This unit may be equipped as a save tape if it is desired to use these PREPROC generated routines on a later run. In this case PREPROC loading and execution may be bypassed and only this tape is loaded.

S2 - This unit may be saved if it is desired to replot from these plot tapes at a later date. In this case the PLOT Processor may be run as a separate job with the desired input tapes.

2.3 User Supplied Program Linkage

When a special purpose program is to be loaded at object time, the user must construct the following linkage in his subroutine:

```
  SUBROUTINE ANYNAME1
  COMMON/TIME/SIMTIME(50)
  DATA (I = 0)
  IF (I.EQ.O) CALL SANDF (I,P,DT,ST)
  IF (SIMTIME.LT.ST) GO TO 1
  CALL SANDF (I,P,DT,ST)
  GO TO 2
1 CALL SCHED (DT, 8H ANYNAME1,P,1)
2

  END
```

The user's first statements should begin at label 2. This linkage is all that is necessary in order for the routine to be scheduled as any other routine under the *PROGRAMS Preprocessor Option.

2.4 Checkpoint Restart

In order to restart an earlier job, one must use the L. U. 31 output tape from that job as input to the following restart control deck:

${}^{7}_{9}$JOB

${}^{7}_{9}$DEMAND,40000B,7MT

${}^{7}_{9}$EQUIP

```
[ALL I/O EQUIP CARDS]

7
9LOAD,10

7
9RUN.

*PROGRAMS

7 7
8 8

*START

7 7
8 8

7
9COMPASS, I = 3, X = 15

11
 0
 7BANK, (0),LC.,(1),CHKPNTW
 9

7
9LOAD,35

7
9RUN

I 9 (4) =  {RESTART NUMBER (See Section 4.2)}
$
```

3.0 PREPROCESSOR INPUT

The Preprocessor (PREPROC) is a set of routines which constitute the first of the three processors that comprise the AGS Simulator. Function of the Preprocessor is to construct a set of routines which are executed by the MAIN Processor. From a systems standpoint, the sequence of events is as follows:

1) PREPROC is loaded.

2) Begin execution of PREPROC.

3) Read PREPROC data cards.

4) Write out BCD card image of the routines on LUN 3.

5) Call in the Assembler, assemble routines, and write load and go on LUN 15.

6) Load in MAIN Processor

7) Load LUN 15.

8) Begin execution of MAIN Processor.

PREPROC generates four types of programs: an input program, a variable number of output programs, a "Timelist" program, and a program containing jumps to subroutines, whose entry points have been defined by PREPROC data cards.

The gross format of all PREPROC data sets is identical. A data set is defined as an * OPTION card plus all data cards pertinent to that option and terminated by a standard SCOPE system end of file card ($^{7}_{8}\ ^{7}_{8}$). The first card of the data set contains an * in column 1 followed by the option name. *PROGRAMS, *OUTPUT, *CALCOMP, *INPUT, *TIMELIST, *START are the only legal first cards. With the exception of the *START option the order of the data sets is immaterial. *START and EOF cards are <u>always</u> last, <u>always</u> needed and serve as the PREPROC data terminator.

Three of the Preprocessor options, *PROGRAMS, *OUTPUT, and *CALCOMP, are interconnected in that the latter two options build programs which are scheduled by the former.

3.1 PROGRAMS Option

Although the program built by use of this option is the most simple, it adds great flexibility to the simulator and therefore is most difficult to explain. By intelligent use of this option the user has essentially unlimited capability to extend, modify or "foul-up" the operation of the simulator. In a sweeping statement any routine may be written, interfaced with any other simulator or system routine through common or call statements, executed during the simulation at a specific time, recalled at a periodic frequency, executed for a set length of time, or operated in a time/priority mode. In production runs this extended capability will probably not be fully utilized. For such runs, "standard" routines, like the standard print, checkpoint, plot dump and termination programs, will be controlled by this option. (See Section 6.1).

1	13	25	37	49
ENTRY POINT	START TIME	STOP TIME	DELTA TIME	PRIORITY

The format for each data card is indicated above. All data cards follow an *PROGRAMS card and are read in fixed format (A8, 4X, 4E12,8). ENTRY POINT is a BCD name (up to 8 characters) of the program to be called and starts in column 1. START TIME is the simulation time the routine is to be entered, and STOP TIME designates the time execution of the routine is to be terminated. DELTA TIME sets the update frequency of the program, and PRIORITY is a number used to reorder the execution of jobs if two or more jobs are scheduled to be executed at the same time. A low number is a high priority, 0 being "Do This Job First".

If it is desired to vary the frequency of execution of any one particular routine during the run, more than one card is required. For this operation the cards (for the particular routine) must be arranged one after the other in increasing start times among the cards following the *PROGRAMS card. An example is given below.

EXAMPLE 1

A special print routine has been defined in the OUTPUT Option called STATEVEC. It is desired that this program be executed every .02 seconds from 100. to 102. seconds and every 5 seconds from 107. seconds until normal termination, which is assumed to be 500. seconds. Two cards for STATEVEC are needed. The first schedules the high frequency print and the second, the low frequency print. NOTE, THAT FOR THIS OPERATION, THE ORDER OF THESE TWO CARDS MUST BE IN INCREASING START TIMES.

*PROGRAMS

CALCOMP	104.	400.	2.	
STATEVEC	100.	102.	02.	100.
STATEVEC	107.	500.	5.	100.
HIGHFQ	0.	500.	2.	50.
TERMIN	500.			
7 7				
8 8				

3.2 OUTPUT Option

This PREPROC option generates a user controlled number of PRINT programs that are executed in the MAIN Processor. Since Standard print routines are available in the MAIN Processor (See Sections 6.1 & 6.2), the OUTPUT option need be used only when additional parameters are to be printed. The corresponding print programs are scheduled by use of the *PROGRAMS option.

The first data card read is an *OUTPUT card followed by a program name card. The alphanumeric symbol,or name, starts in column 1 and consists of 1 to 8 characters composed of letters, numbers, or special characters (≠, <, ., v). This symbol may not begin with a numeric character and is also used in the first field of the *PROGRAMS data card which schedules this routine. Each card following the name card defines a variable, format, scale factor and variable name the user wants to print. The variable is defined by giving its location (either decimal or octal) within a common block. When successive variables are within the same common block, the common block name does not have to be updated. Data may begin in any column and must contain no imbedded blanks (except for the BCD variable name). The data card format is shown below.

COMMON BLOCK + OFFSET,BCD NAME, FORMAT, SCALE FACTOR,
CONV FACTOR

If the OFFSET is in octal, an O must follow the octal digits. BCD NAME can have as many as 8 characters and can consist of any character except a comma (,). Legal FORMAT identifiers are O for octal, E for 3800 floating point and I for decimal integer. If B is used for the SCALE FACTOR, the data will be converted from an AGS fractional number to 3800 F.P. before printing. (It should be obvious that, when a scale factor is used, the format should be E.) The CONV FACTOR consists of up to 12 numeric characters and is the number by which the variable will be multiplied before output occurs. If the SCALE FACTOR is not to be used, then two commas must separate the CONV FACTOR and the FORMAT. An example of the use of the OUTPUT option is given below.

EXAMPLE 2

```
*OUTPUT

STATEVEC

    MEMORY + 3410,LEM X,E,B23

    DFILE + 231,ALT,E,,.00016458

    CFILE + 2,FLAG SW,I

    MEMORY + 3410,OCTAL X,O

           + 3420,ATT Y,E,B23,57.296

           + 2370,F00,O

7 7
8 8
```

In the above example an output routine called STATEVEC will print the variables called out and label them with the characters appearing in the BCD name field when the program STATEVEC is executed in the MAIN Processor. However, it should be remembered that STATEVEC must be scheduled under *PROGRAMS.

If it is desired to use different print frequencies for some of the variables, the data cards may be separated into additional blocks under other program names. For instance, if the first two variables in Example 2 are to be printed at a frequency different from the remaining variables, an additional *OUTPUT card and an additional program name card are required, as shown in Example 3.

EXAMPLE 3

```
*OUTPUT

STATEVEC

    MEMORY + 3410,LEM X,E,B23

    DFILE + 231,ALT,E,,.00016458

7 7
8 8

*OUTPUT

STATEVAR
```

```
CFILE + 2,FLAG SW,I

MEMORY + 3410,OCTAL X,O

       + 3420,ATT Y,E,B23,57.296

       + 2370,FOO,O

7 7
8 8
```

Time intervals corresponding to the desired frequencies are designated on the STATEVEC and STATEVAR data cards listed under *PROGRAMS.

Should it be desired that during a run the print frequency vary for a particular routine, say STATEVEC, then additional STATEVEC data cards are required under *PROGRAMS. This is exemplified in Example 1 of the previous section.

During execution the variables are printed four across a page and the name of each variable precedes its value. Current simulation time and the routine name (e.g., STATEVEC) are also printed.

3.3 CALCOMP Option

This option generates a routine called CALCOMP which must be scheduled by the user via the *PROGRAMS option. The routine name (CALCOMP) is used on a data card following the *PROGRAMS card. Data cards read by this option are of the same format as those described under the *OUTPUT option except that the name card (second card in the *OUTPUT data set) is deleted. Also, all format options should be E since only floating point data is plotted. Since only one routine (CALCOMP) is constructed by this option, *CALCOMP may appear only once. An example follows:

EXAMPLE 4

```
*CALCOMP

    MEMORY + 550,LEM X,E,B28

           + 560,LEM Y,E,B28

           + 570,LEM Z,E,B28

    TIME   + 1,  TIME,E
```

```
DFILE + 185,OMEGA X,E
      + 186,OMEGA Y,E
      + 187,OMEGA Z,E
7 7
8 8
```

It should be restated that the routine CALCOMP must be scheduled under *PROGRAMS, as shown in Example 1.

Plotting of variables output by the CALCOMP option is controlled by input cards read by the PREPROC Plot Processor. It should also be mentioned that only [redacted] are available from the common blocks of the MAIN Processor. Also, the same plot frequency must be used for all variables selected by the CALCOMP option. This frequency is designated by the appropriate time interval on the CALCOMP data card following *PROGRAMS. Further, this frequency can be made to change at a specific time during a run by inserting a second CALCOMP data card under *PROGRAMS. In this case the first CALCOMP data card must contain data (Start Time, Stop Time, Δt) related to the first plot frequency and the second data card similar data related to the second plot frequency; these two cards must be arranged in the order of increasing start times, as indicated in Section 3.1.

It should be pointed out that only [redacted] are available for any one plot. Therefore, the user must carefully examine the plot capabilities before selecting a plot frequency (or frequencies). If the run is very long, he may want to use two frequencies, one high and the other low. The relationship below can be used to determine the plot frequencies or time period for each plot frequency.

$$\frac{T1}{\Delta t_1} + \frac{T2}{\Delta t_2} \leq 1600$$

where T_1 is the time period for plotting at the frequency related to Δt_1 and T_2 is the period for plotting at the frequency related to Δt_2. If only one frequency is desired, then the following relationship must be adhered to:

$$\frac{T}{\Delta t} \leq 1600$$

where T is the total run time.

3.4 INPUT Option

All of the card input for the MAIN Processor is accomplished by one call to a general Read routine called INDATA. INDATA is a super set of READDATA and contains some special input options developed expressly for the AGS Simulator (See Section 4.1). The PREPROC INPUT option simply extends the number of data arrays that INDATA can read into. If the "standard" input routine is all that is necessary, then this PREPROC option is unnecessary (See Section 4.2). The data cards read by the INPUT option are only for building the extra arguments in the INDATA call. The actual call to INDATA, which reads the simulation data, is done at initial condition time of the MAIN Processor (See Section 4.0).

All extra common block names start in column 1 and require one card each. The total number of arguments in the call cannot exceed 77, octal. The following example indicates how the INPUT option is used:

EXAMPLE 5

It is assumed that the "standard" input program contains four arrays (here, common block and array are synonomous) named FLAGBUF, MEMORY, DFILE, and CFILE. The input program executed in the MAIN Processor will be equivalent to the FORTRAN statement,

```
CALL INDATA (FLAGBUF, MEMORY, DFILE, CFILE)
```

If a user wishes to also read into another common block called MYDATA, three PROPROC data cards for this option are necessary. These cards follow the *INPUT card, as shown below.

```
*INPUT

MYDATA

7 7
8 8
```

The equivalent FORTRAN statement is now

```
CALL INDATA (FLAGBUF, MEMORY, DFILE, CFILE, MYDATA)
```

where MYDATA is block number 5.

It is important to note that the INPUT option is not scheduled under *PROGRAMS.

3.5 TIMELIST Option

This PREPROC option allows the user to build a program to set data into any common location at a specified time. Types of data are 3800 floating point, decimal integer, octal, or AGS type with scale factor conversion.

The first data field describes the common block name and either an octal or decimal offset. In the second data field is the time (real simulation time) at which the data is to be entered. The last field contains the data, as follows:

> 3800 Floating Point - up to 12 numeric characters; decimal point required
>
> Decimal Integer - no decimal point
>
> Octal - octal characters followed by 0
>
> AGS Data - floating point number followed by B and integer scale factor

EXAMPLE 6

```
*TIMELIST

   MEMORY + 4770,600.2,57.5B10

   DFILE  + 99,32.5,10

          + 452,900.2,7760

          + 388,10.5,599.

7 7
8 8
```

When using the scale factor option, the number to be scaled must have a decimal point, e.g., 57B10 must be written as 57.B10.

As in the case of the INPUT option, use of the TIMELIST option does not require scheduling under *PROGRAMS.

4.0 AGS MAIN PROCESSOR INPUT

The AGS MAIN Processor is a set of routines which simulate the AGS computer and its environment under executive control. MAIN Processor data input is the flight program tape and card input.

At the beginning of execution of the MAIN Processor an initialization subroutine, which first reads in the flight program memory tape and then makes a call to the INDATA input routine, is called. INDATA is a modified form of the generalized input routine READDATA. The use of this routine and the AGS modifications made to it are described in Section 4.1. A complete description of READDATA, along with documentation of several special features, may be found in the Appendix.

A standard calling sequence has been assembled in the initialization call to INDATA (See Section 4.2.1) but, as noted in PREPROC documentation, the INPUT option may be used to add arguments (common blocks) to this call. The MAIN Processor data cards will be used to insert data to these common blocks in order to initialize the simulator.

4.1 The INDATA Routine

The INDATA routine is used in the MAIN Processor of the AGS simulator to read a set of data cards to initialize the simulator and the flight program. This set of data cards is terminated by reading of a dollar sign (11 - 3 - 8 punch). Data fields on these cards begin in column 1 and are usually terminated by reading the last column (column 72) or by encountering a slash (0 - 1 punch). All characters after the slash may be used as comments. A data field usually assumes the following form (although other forms may be used as described in the Appendix):

MC (S) = N1,N2,N3,... / COMMENT

where M specifies the <u>mode</u> of the data. Modes used in the AGS simulator are

O	OCTAL
I	INTEGER
E or F	FLOATING POINT (REAL)

C is an integer (Block Number) indicating into which variable (Block Name) in the argument list that data is to be entered (See Section 4.2); if additional arguments are added by use of *INPUT, their corresponding values will begin with the number after the last one in the standard list and follow in the order of the cards following the *INPUT card.

S is a positive non zero linear subscript indicating the element of the array associated with it; an O following an element number will flag the displacement as octal rather than decimal.

N1, N2, ... is the numeric field, used in the AGS simulator as described in the next several paragraphs.

DECIMAL FIELDS (I, E, or F)

A decimal field has the following components, the first of which is mandatory:

Principal Part - a decimal number with or without a decimal point. The decimal point, if any, may appear at the beginning of, end of, or within the principal part. If omitted, it is assumed to be located at the right-hand end of the principal part. Omitted for mode I.

Decimal Scale Factor - (optional) consists of the letter E or D followed by a signed or unsigned decimal integer. The value preceding E or D will be multiplied by 10 to this power. The decimal scale factor must follow the principal part. (NOTE: D and E are completely interchangeable in this case). Restricted to mode E or F.

Conversion Factor - (optional) consists of the letter M separating two decimal numbers in a field. The product of the two values will be stored as one element. Restricted to mode E or F.

Binary Scale Factor - (optional) consists of the character B followed by a signed or unsigned decimal integer. The number following B represents the power of 2 which, when 2 to this power is divided into the decimal value preceding the B, will produce an 18 bit fixed point number of AEA specifications. Negative values are represented as two's complement. Restricted to mode E or F.

EXAMPLE 7

```
E7 (85) = 1188.3, 1E3, -.987D-4  /VX, VY, VZ

F6 (185) = 1.0  /FAST CONTROL

F4 (51) = 2.76 M57.2957  /PITCH DEGREES

F2 (3070) = 3.0B2, -.5DB2, .25B-1, -.25B-1
```

E2 (230) = 5.23E5B23

E2 (270) = 3.9874B4M2.54B17

OCTAL FIELDS (I or O)

An octal field may be input in either of the following manners for the AGS simulator:

Integer Mode - consists of a string of not more than 16 octal digits (0-7), with or without a sign, followed by the letter O or K. A positive decimal integer N placed after the O or K will cause N zeros to be appended to the right of a positive octal number. The effect of a minus sign preceding the number is to "one's complement" the entire word after zeros are appended. (Note: O and K are completely interchangeable in this case).

Octal Mode - operates as described above except the O or K need not be used after the number. Obviously, all zeros must be appended to the octal number as input.

EXAMPLE 8

I 11 (6) = 357013, 176K13 /BITWORDS

0 11 (1) = 1042, 377, 1143, 3000 /EXEMPT

4.2 Initialization Blocks

The following common blocks have been assembled in the standard initialization call to INDATA:

BLOCK NUMBER	BLOCK NAME
1	FLAGBUF
2	MEMORY
3	BLK1
4	VEH1
5	VEH3
6	CONT
7	UNIVR
8	TARGT
9	TIME
10	NOERROR
11	DEDABUF
12	SWCOM

The sets of maps on the next several pages describe the MAIN Processor input to these blocks for a nominal simulator run.

```
C
C * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C
C
C                     INSTRUCTION TRACE CONTROL
C
C
C * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C
C- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
C
C* TRACE CONTROL CELLS *
C
CELL            BLOCK               MNEMONIC
C
C 1             FLAGBUF             TON
C                 1
C  TO
C 10
C
C 11            FLAGBUF             TOFF
C                 1
C  TO
C 20
C
C 21            FLAGBUF             LOC1
C                 1
C  TO
C 30
C
C 31            FLAGBUF             LOC2
C                 1
C  TO
C 40
C
C
C- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
C
C- TRACE VARIABLE DESCRIPTION -
C
C NOTE -
C     FOUR ENTRIES ARE REQUIRED FOR EACH TRACE REQUEST.
C     THE BLOCK IS DIVIDED INTO FOUR ARRAYS OF TEN WORDS. THE CORRESPONDING
C     POSITIONS WITHIN THE ARRAYS CONTROL THE TEN POSSIBLE TRACE REQUESTS.
C
C   TON         BEGINNING TIMES FOR TRACE. FLOATING POINT SECONDS.
C
C   TOFF        ENDING TIMES FOR TRACE.
C
C   LOC1        BEGINNING LOCATION FOR TRACE. OCTAL AEA ADDRESS
C
C   LOC2        ENDING LOCATIONS FOR TRACE.
C
C * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C
C                     ASTRONAUT  INPUT
C
C * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C
```

```
C- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
C* ASTRONAUT  INITIALIZATION CELLS *
C
CELL               BLOCK              MNEMONIC
C
C 1                BLK1               TIME
C                   3
C TO
C 75
C
C 101              BLK1               CODE
C                   3
C  TO
C 175
C
C 201              BLK1               DATA1
C                   3
C  TO
C 275
C
C 301              BLK1               DATA2
C                   3
C  TO
C 375
C
C
C- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
C
C- ASTRONAUT VARIABLE DESCRIPTION -
C
C
C NOTE -
C        THIS BLOCK IS DIVIDED INTO FOUR ARRAYS OF 100 WORDS,OF WHICH THE
C        USER MAY ACCESS 75. AN ASTRONAUT FUNCTION IS REQUESTED BY SETTING
C        THE CORRESPONDING VARIABLES WITHIN EACH ARRAY.
C
C   TIME         TIME IN FLOATING POINT SECONDS FOR THIS FUNCTION.
C
C   CODE         INTEGER CODE FOR REQUESTED FUNCTION. (SEE LIST BELOW)
C
C   DATA1        OCTAL ADDRESS FOR USE WITH CODES 1 AND 2. (SEE BELOW)
C
C   DATA2       FLOATING POINT DATA WORD FOR REQUESTED FUNCTION. (SEE BELOW)
C
C
C  FUNCTION        CODE      DATA1               DATA2
C
C  DEDA INPUT        1       ADDRESS             DATA
C                                                ADJUST DECIMAL POINT TO PLACE
C                                                THE FIVE DIGITS THAT ARE TO
C                                                BE ENTERED INTO THE DEDA
C                                                IMMEDIATELY TO THE LEFT OF
C                                                THE DECIMAL POINT.
C
C  DEDA OUTPUT       2       ADDRESS             TIME DURATION OF DISPLAY IN
C                                                SECONDS. NO ENTRY (ZERO)
C                                                WILL RESULT IN A 40 MS
C                                                DURATION.
C
```

```
C   GUIDANCE           3      -------             0. - AGS
C   CONTROL SWITCH                                1. - PGNS
C
C   MODE CONTROL       4      -------             0. - OFF
C   SWITCH                                        1. - AUTOMATIC
C                                                 2. - ATTITUDE HOLD
C
C   ABORT STAGE        5      -------             0. - RESET
C   BUTTON                                        1. - ABORT STAGE
C
C   ABORT BUTTON       6      -------             0. - RESET
C                                                 1. - ABORT
C
C   DOWNLINK UPDATE    7      -------             -------
C
C  NOTE -
C      ALLOW A MINIMUM OF 6 SEC. AFTER A DOWNLINK BEFORE ANOTHER UPDATE,
C      AND 4 SEC. BEFORE ANY OTHER ASTRO FUNCTION.
C
C
C   LM UPDATE VIA      8      -------             -------
C   DEDA
C
C  NOTE -
C      ALLOW A MINIMUM OF 18 SEC. AFTER AN UPDATE BEFORE SCHEDULING
C      ANOTHER ASTRO FUNCTION.
C
C
C   CSM UPDATE VIA     9      -------             -------
C   DEDA
C
C  NOTE -
C      ALLOW A MINIMUM OF 18 SEC. AFTER AN UPDATE BEFORE SCHEDULING
C      ANOTHER ASTRO FUNCTION.
C
C
C   ULLAGE             10     -------             -------
C
C   AXIS BY AXIS
C   RCS THRUSTING      11     -------             0. - ALL 3 AXES
C                                                 1. - X AXIS ONLY
C                                                 2. - Y AXIS ONLY
C                                                 3. - Z AXIS ONLY
C
C   INACTIVE           12
C
C  TPI SEARCH          13     NOMINAL TIME OF     NOMINAL VALUE OF THLOS
C                             ULLAGE IN FLT. PT.  FLT. PT. DEGREES.
C                             SECONDS
C
C
C   RENDEZVOUS         14     -------             0. - MAKE 5 MARKS
C   RADAR                                         X. - MAKE X MARKS
C
C   ENGINE ON/OFF      15     -------             0. - ENGINE OFF
C                                                 1. - ENGINE ON
C
C   DESCENT ENGINE     16     -------             0. - NOT ARMED
C   ARM                                           1. - ARMED
```

```
C
C   ASCENT ENGINE     17          -------                        0. - NOT ARMED
C   ARM                                                          1. - ARMED
C
C   DESCENT ENGINE    18          -------                        PERCENT OF THROTTLE
C                                                                DESIRED.
C                                                                (0. TO 100.)
C
C   BANDWIDTH SWTTCH 19           -------                        0. - MAX.
C                                                                1. - MIN.
C
C   JET SELECT        20          -------                        0. - 2 JETS
C   SWITCH                                                       1. - 4 JETS
C
C
C * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C
C
C                        ENVIRONMENTS INITIALIZATION
C
C
C * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C      LOCATIONS  WHICH  MUST  BE  INITIALIZED  FOR  A  NOMINAL
C      LM  RUN.  IN  SUCH  A  RUN  THE  MAIN  ENGINE  IS  ASSUMED
C      OFF .
C      ALL ENVIRONMENTS INPUT VALUES ARE FLOATING POINT UNLESS OTHERWISE STATED
C
C- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
C
C* LM INITIALIZATION CELLS *
C
C
CELL            BLOCK              MNEMONIC
C
C 406           VEH1               STAGE
C                4
C
C 209           VEH3               ENGARM
C                5
C
C 654           VEH1               WV(1)
C                4
C
C 655           VEH1               WV(2)
C                4
C
C 656           VEH1               WV(3)
C                4
C
C 176           VEH3               DFUELL
C                5
C
C  17           VEH3               AFUELL
C                5
C
C 185           CONT               FASTSW
C                6
C
C                                                                *
```

```
C NOTE -
C     STABLE  MEMBER  IS  INITIALIZED  BY  SPECIFYING  EITHER  TSI
C     OR  EULERY, Z, X.
C 954             VEH1               EULERY
C                  4
C
C 955             VEH1               EULERZ
C                  4
C
C 956             VEH1               EULERX
C                  4
C
C 957             VEH1               TSI(1)
C                  4
C  TO                                 TO
C 965                                TSI(9)
C
C NOTE -
C     VEHICLE  ATTITUDE  IS ESTABLISHED  BY  INITIALIZING  P,Y,R,
C     RELATIVE  TO  LOCAL  VERTICAL  OR  BY  INITIALIZING  THE ATTITUDE
C     RELATIVE  TO  THE  STABLE  MEMBER. IN THE  LATTER  CASE  EITHER
C                                                    *
C     THE  GIMBAL  ANGLES (GAI, GAM, GAO )  OR  TVS  ARE  INITIALIZED.
C  51             VEH1               PITCH
C                  4
C
C 592             VEH1               YAW
C                  4
C
C 288             VEH1               ROLL
C                  4
C
C 951             VEH1               GAI
C                  4
C
C 952             VEH1               GAM
C                  4
C
C 953             VEH1               GAO
C                  4
C
C 966             VEH1               TVS(1)
C                  4
C  TO                                 TO
C 974                                TVS(9).
C
C  13             VEH1               ASAERR
C                  4
C  56             VEH1               PRNTCT
C                  4
C 204             VEH3               DTPRIN
C                  5
C
C
C- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
C
C- LM VARIABLE DESCRIPTIONS -
C
C   STAGE     0. =  ASCENT          1. =  DESCENT
```

```
C              2. =  CSM ATTACHED
C
C
C    ENGARM     0. = OFF                  1. = ON
C
C
C    PITCH,YAW,ROLL   ARE   SUCCESSIVE   ROTATIONS  ABOUT  VEHICLE
C    Y , Z , X   AXES  (X,Y,Z  RIGHT HANDED SYSTEM )  RELATIVE  TO
C    INITIAL  ORIENTATION OF  VEHICLE  ALONG  LOCAL  VERTICAL
C    I.E.  XBODY =YBODY X ZBODY
C                  -    -
C          YBODY = V X R
C          ZBODY  ALONG -POSITION VECTOR TO SPACECRAFT FROM  MAJOR  GRAV-
C          ITATING BODY
C    UNITS   DEGREES
C
C
C    WV(1) ,(2), (3)  ARE  ANGULAR  RATES  OF VEHICLE  AROUND  XBODY,
C    YBODY , ZBODY    UNITS   DEGREES/SECOND
C
C
C    DFUELL       INITIAL  FRACTION (0. TO 1.)  OF  DESCENT   FUEL  LOAD
C
C
C    AFUELL       INITIAL  FRACTION (0. TO 1.)  OF  ASCENT    FUEL  LOAD
C
C
C    FASTSW     0.  =  SLOW  CONTROL  1.  =  FAST  CONTROL
C
C
C
C     GAI,GAM,GAO  ARE INNER, MIDDLE AND OUTER GIMBAL ANGLES
C     INPUT UNITS  DEGREES
C
C     EULERY,Z,X  ARE  EULER ANGLE ROTATIONS OF THE STABLE  MEMBER
C     RELATIVE  TO  REFERENCE  INERTIAL
C        INPUT  UNITS  DEGREES
C
C
C     TSI   IS  TRANSFORMATION  MATRIX  FROM  REFERENCE  INERTIAL  TO
C     STABLE  MEMBER  COORDINATES
C
C     TVS   IS  TRANS. MATRIX  FROM  STABLE  MEMBER TO BODY COORDINATES
C
C
C     ASAERR   IS   FLAG  INDICATING  THAT  ERRORS  ARE  TO BE
C     INCLUDED  IN   ACCELEROMETER  AND  GYRO   MODELS.  ON = 1.
C     PRNTCT   LM  CONTROLLED  PRINT  SWITCH. OPTIONS  ARE THOSE  LIST-
C              ED  IN  LM  DOCUMENT.
      DTPRIN   CONTROLS  DELTA T   BETWEEN  LM  PRINTS
               = 0.  PRINT  EVERY   INTEGRATION  (WITH PROPER PRNTCT
               SETTING )
               UNITS -  SECONDS
C
C
C
C    ADDITIONAL   CELLS  WHICH  MAY  BE  INITIALIZED
C   13          VEH1                ASAERR        FLAG SET TO  1.  WHEN  GYRO
C                4                                AND ACC. BIASES ARE DESIRED
```

```
C 257          VEH1           RCSBIT(1),  THESE 4 CELLS  CONTROL 4
C              4              2,3,4       QUAD ERRORS(SEE LM MANUAL)
C 478          VEH1           THBITS      SEE  MANUAL
C              4
C 638          VEH1           BJPY        Y NOZZLE TRIM  ANGLE(DES-
C              4                          CENT) IN  DEGREES INIT.
C 639          VEH1           BJPZ        Z NOZZLE TRIM
C              4
C   2          VEH3           AEBITS      SEE  LM  MANUAL
C              5
C 164          VEH3           DEBITS      SEE  LM  MANUAL
C              5
C 165          VEH3           DEBUG       SEE  LM  MANUAL
C              5
C 335          VEH1           SLOSH       SET  TO 1.  FOR  SLOSH
C              4                          DURING  THRUSTING
C
C
C- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
C
C* UNIVERSE INITIALIZATION CELLS *
C
C
CELL           BLOCK          MNEMONIC
C
C  56          UNIVR          EPOCH
C              7
C
C  57          UNIVR          ORIGIN
C              7
C
C  80          UNIVR          R(X)
C              7
C
C  81          UNIVR          R(Y)
C              7
C
C  82          UNIVR          R(Z)
C              7
C
C  85          UNIVR          V(X)
C              7
C
C  86          UNIVR          V(Y)
C              7
C
C  87          UNIVR          V(Z)
C              7
C
C  91          UNIVR          PRNTSW
C              7
C
C
C 106          UNIVR          PERT
C              7
C
C 172          UNIVR          YEAR
C              7
C
```

```
C
C- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
C
C- UNIVERSE VARIABLE DESCRIPTIONS *
C
C
C
C     EPOCH          TIME  MEASURED  IN  DAYS  FROM  MIDNIGHT  BEFORE
C                    JULY  1  (GREENWICH TIME ) TO SIMULATOR TIME=0.
C
C
C     YEAR  SPECIFIES  EPHEMERIS  DATA  PERIOD  STARTING  JULY  1.
C     1967-1968 YEAR = 1968.    1968-1969  YEAR = 1969.
C
C     ORIGIN         ORIGIN OF REFERENCE  INERTIAL  COORDINATE SYSTEM
C                    1. = CENTER OF EARTH
C                    2. = CENTER OF MOON
C                    ORIGIN SHOULD CORRESPOND TO PRINCIPAL GRAVITATING
C                    BODY (EARTH  OR  MOON )
C
C     -
C     R              R(X),R(Y),R(Z)  VEHICLE POSITION IN REFERENCE INER-
C                    TIAL COORDINATES RELATIVE TO SPECIFIED ORIGIN.
C     INPUT  UNITS - FEET
C
C     -
C     V              V(X),V(Y),V(Z)  VEHICLE VELOCITY IN SAME COORDINATES
C                    AS ABOVE
C     INPUT  UNITS - FEET/SECOND
C
C     PRNTSW         0. =  VIRTUALLY NO  UNIVRS  CONTROLLED  PRINT (EXCEPT
C                          INITIALLY )
C                    1. =  TIME AND STATE VECTOR (IN METRIC UNITS)
C                          EACH TRAJECTORY INTEGRATION CALL , WHICH NORM-
C                          ALLY OCCURS AT LEAST EVERY 2 SECONDS.
C                    2. =  SAME AS 1. PLUS  PRINTING OF TIME AND MODE
C                          AT EACH UNIVRS CALL.
C
C     PERT           0. =  SPHERICAL GRAVITY MODELS FOR TRAJECTORY INTEG-
C                          RATION.
C                    NONZERO (USUALLY SET TO 1.) = PERTURBATIONS TO G-
C                    FIELD DUE  TO SUN  AND OTHER(NON-ORIGIN) BODY
C                    AND ALSO  NONSPHERICAL FIELD OF PRINCIPAL BODY.
C
C
C
C
C
C
C- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
C
C* TARGET INITIALIZATION CELLS *
C
C
C
CELL            BLOCK            MNEMONIC
C
C  58           TARGT            RT(X)
C                 8
```

```
C
C  59              TARGT                   RT(Y)
C                    8
C
C  60              TARGT                   RT(Z)
C                    8
C
C  96              TARGT                   VT(X)
C                    8
C
C  97              TARGT                   VT(Y)
C                    8
C
C  98              TARGT                   VT(Z)
C                    8
C
C
C  21              TARGT                   DTPRNT
C                    8
C
C
C
C- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
C
C- TARGET VARIABLE DESCRIPTION -
C
C
C     DTPRNT          TIME INTERVAL BETWEEN  STATE VECTOR PRINT
C                     UNITS  SECOND
C     -
C     RT              TARGET(PASSIVE) VEHICLE POSITION  VECTOR (SAME ORIGIN
C                     AS UNIVRS)
C     INPUT  UNITS - FEET
C
C     -                                                          -
C     VT              TARGET VEHICLE  VELOCITY (SAME COORDINATES AS RT)
C     INPUT  UNITS - FEET/SECOND
C
C
C
C- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
C
C* LEM/ASTRONAUT SWITCH INITIALIZATION *
C
C
CELL               BLOCK                   MNEMONIC
C  11              SWCOM                   SW(11)      ABORT BUTTON
C                   12                                 1.=SET 0.=RESET
C  12              SWCOM                   SW(12)      ABORT STAGE BUTTON
C                   12                                 1.=SET 0.=RESET
C  13              SWCOM                   SW(13)      BALANCE COUPLE SWITCH
C                   12                                 0.=JETS 1,5,9,13 DISABLED
C                                                      1.=JETS 1,5,9,13 ENABLED
C   1              SWCOM                   SW(1)       GUIDANCE  CONTROL  SWITCH
C                   12                                 0.= AGS  1.= LGC
C   2              SWCOM                   SW(2)       MODE CONTROL SWITCH
C                   12                                 0.=OFF 1.=AUTO 2.=ATT.HOLD
C   3              SWCOM                   SW(3)       + X TRANS.
C                   12                                 1.=ON 0.= OFF
```

```
C    4              SWCOM                SW(4)           D.E. ARM
C                     12                                 1.=ARMED 0.=NOT ARMED
C    5              SWCOM                SW(5)           A.E. ARM
C                     12                                 1.=ARMED 0.=NOT ARMED
C    6              SWCOM                SW(6)           D.B. SWITCH(SLOW MODEL)
C                     12                                 0.=WIDE 1.=NARROW
C    7              SWCOM                SW(7)           214 JET X TRANS.SWITCH
C                     12                                 0.= 2 JET 1.= 4 JET
C    8              SWCOM                SW(8)           ACA OUT-OF-DETENT
C                     12                                 0.= IN-DETENT 1.=OUT-OF-DETENT
C    9              SWCOM                SW(9)           MANUAL START
C                     12                                 1.=SET 0.=RESET
C   10              SWCOM                SW(10)          MANUAL STOP
C                     12                                 1.=SET 0.=RESET
C
C
C * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C
C                        EXECUTIVE INITIALIZATION
C
C * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C
C- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
C
C* EXECUTIVE INITIALIZATION CELLS *
C
CELL                BLOCK                MNEMONIC
C
C 1                 TIME                 TIME
C                     9
C
C
C 4                 TIME                 RS
C                     9
C
C 5                 TIME                 TIMEFLG
C                     9
C
C
C 13                TIME                 IOPEN
C                     9
C
C- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
C
C- EXECUTIVE VARIABLE DESCRIPTION -
C
C   TIME            SIMULATOR INITIAL TIME IN FLOATING POINT SECONDS.
C                   TIME FOR AEA MUST ALSO BE INITIALIZED IN MEMORY BLOCK.
C
C   RS              FOR USE WITH CHECKPOINT RESTARTS. SET TO INTEGER NUMBER
C                   OF CHECKPOINT TO BE RESTARTED AT.
C
C
C   TIMEFLG         ICS TIME CHECK FLAG. CHECKS TO SEE IF TOTAL TIME OF
C                   EXECUTION OF A MINOR CYCLE IS LESS THAN OR EQUAL TO
C                   20 MS. SET TO -1.0 FOR THIS CHECK, ZERO (OR NO ENTRY)
C                   FOR NO CHECK.
C
C
```

```
C   IOPEN   . OPEN LOOP RUN. INTEGER, 1 IF OPEN LOOP
C                                   0 FOR NORMAL CLOSED LOOP
C
C
C * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C
C               EXEMPTING AEA LOCATIONS FROM ERROR MESSAGES
C
C * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C
C- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
C
C* EXEMPT INITIALIZATION CELLS *
C
CELL          BLOCK               MNEMONIC
C
C 1           NOERROR             OBLOCK
C               10
C  TO
C 200
C
C- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
C
C- EXEMPT VARIABLE DESCRIPTION -
C
C   OBLOCK    THIS ARRAY CONTAINS THE OCTAL AEA ADDRESSES OF LOCATIONS TO
C             BE EXEMPTED FROM OVERFLOW ERROR MESSAGES.
C
C
C * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C
C               INPUT DISCRETE WORD INITIALIZATION
C
C * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C
C- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
C
C* BITWORD INITIALIZATION CELLS *
C
CELL          BLOCK               MNEMONIC
C
C 6           DEDABUF             BITWORD1
C               11
C
C 7           DEDABUF             BITWORD2
C               11
C
C- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
C
C- BITWORD DESCRIPTION -
C
C NOTE -
C      THE BITWORDS WILL BE INITIALLY IN OFF STATES IF THERE IS NO INPUT.
C      THE WORD MUST BE INPUT IN A LEFT JUSTIFIED OCTAL 3800 WORD FORMAT.
C
C   DIS BIT NO.   3800 BIT NO.    DIS WD1             DIS WD2
C
C        1            46          DOWNLINK TEL STOP   GSE DISCRETE 1
C        2            45          OUTPUT TEL STOP     GSE DISCRETE 2
```

```
C        3              44            FOLLOW UP              GSE DISCRETE 3
C        4              43            AUTOMATIC              DEDA CLEAR
C        5              42            DESCENT ENG ON         DEDA HOLD
C        6              41            ASCENT ENG ON          DEDA ENTER
C        7              40            ABORT                  DEDA READOUT
C        8              39            ABORT STAGE
C
C
C     THESE WORDS USE INVERSE LOGIC - A 1 IN A BIT POSITION DENOTES OFF.
C
C    BITWORD1   INPUT DISCRETE WORD1  INITIAL SET VALUE = 377000000000000
C
C    BITWORD2   INPUT DISCRETE WORD2  INITIAL SET VALUE = 376000000000000
C
C
C******************************************************************************
C
C
C     ADDITIONAL  COMMON BLOCKS WHICH MAY NEED TO BE INITIALIZED
C     THESE BLOCKS  ARE NOT IN THE STANDARD CALL TO INDATA  AND
C     MUST APPEAR IN AN  *INPUT REGUEST
C
C
C******************************************************************************
C
C- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
C
C*  TO START EXECUTION,  IN THE AGS, AT SOME POSITION OTHER THAN 4050  *
C
CELL            BLOCK                  MNEMONIC
C
C 6             REGCOM                 PREG
C
C- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
C
C - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
C
C NOTE -
C     THE NUMBER INTERED INTO PREG SHOULD BE THE OCTAL ADDRESS OF SOME
C     INSTRUCTION IN THE AGS FLIGHT PROGRAM WHERE EXECUTION IS TO BEGIN
C
C - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
C
C * DEBUGING AID *
C
C  TO GET THE NAME AND TIME OF EACH ROUTINE CALLED BY THE EXECUTIVE PROGRAM
C
CELL           BLOCK                  MNEMONIC
C
C 1            NOPTJOB                NOPTSW
C - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
C
C  NOTE-
C
C      THE  SWITCH IS INITALLY SET OFF (SET TO 0)
C      TO GET THE PRINT INTER  A.1
C
C- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
```

5.0 PLOT PROCESSOR

The AGS plot package has been designed as a third processor for the AGS simulator and operates in conjunction with the Preprocessor (PREPROC) and MAIN Processor. The PLOT Processor consists of two separate CALCOMP plot routines. Either of these two plot routines (PREPROC or STANDARD) may be run as a separate job if the dump tapes from a previous simulator run are used as input.

5.1 PREPROC Plot Input

The PREPROC plot routine works in conjunction with the CALCOMP option, described in Section 3.3. Object of this routine is to allow the user a completely flexible plot output function. If this output is desired, a $^{7}_{9}$ LOAD, 22 and $^{7}_{9}$ RUN card, followed by the data sets described below, is necessary.

A Plot Data Set is defined as the total set of data cards necessary to produce one CALCOMP graph. The graph may have up to three variables plotted against one independent variable. There are two types of data cards within a data set. One card, called the Header Card, contains twelve fields and controls the type of plot, scaling, etc; the other card is a Title Card and contains the BCD information the user wants for labeling the graph and the name used on the X and Y-axes.

THE HEADER CARD

One Header Card is required for each plot; a total of three is required when three plots are displayed on one graph. The format for Header Cards is (2A8, 6I2, 4E12.8). The fields, in sequential order, are as follows:

Card Columns	Name	Description
1-8	XNAME	The BCD name of the variable to be plotted on the X-axis. This name must agree with one of the names used in the *CALCOMP data cards.
9-16	YNAME	BCD name of the variable to be plotted on the Y-axis. This must agree with a variable name used in the *CALCOMP data cards.

Card Columns	Name	Description
17-18	NH2	A fixed point 80 or 0 depending on whether another Y-axis is to be generated. The first Header Card must contain an 80 since at least one Y-axis must always be generated.
19-20	IX	A fixed point number giving the desired length of the X-axis in inches.
21-22	IY	A fixed point number giving the length of the Y-axis in inches. If both IX and IY are 0, the plot will be 10" x 10".
23-24	JJ	<u>JJ</u> <u>Plot</u> 0 Line 1 Symbol at each data point N Symbol at each Nth data point -N Only symbol is plotted (no line)
25-26	NP	Number of plots on graph. Fixed point (1, 2, or 3).
27-28	M	<u>M</u> <u>Plot Symbol</u> 0 Routine selects a different symbol for each function on a given frame 1 Octagon 2 Triangle 3 + 4 X 5 Diamond 6 Up arrow 7 Half closed X 8 Z 9 Y 10 Square 11 * 12 Closed X 13 Prime 14 Star
29-40	A1	X-axis minimum. Floating point; if the manual scaling option is used, no point falling outside the desired limits will be plotted.

Card Columns	Name	Description
41-52	A2	X-axis maximum. Rest same as A1.
53-64	B1	Y-axis minimum. Rest same as A1.
65-76	B2	Y-axis maximum. Rest same as A1.

TITLE CARDS

At least three title cards are needed for each graph. If multiple plots are requested (NP), and if a new Y-axis is desired, an additional Y-axis title card is needed. The title cards are read in (10A8) format and a complete card is used for each title. The first card following the Header Card is the graph title, the second is the X-axis label and the third is the Y-axis label.

SCALING OPTIONS

There are two scaling options, Automatic Scaling and Manual Scaling. If the former is to be used, the user enters zero for A1, A2, B1 and B2. Otherwise, he chooses his own limits and sets A1, A2, B1 and B2 accordingly.

SINGLE PLOT

For a single plot (one plot, X vs. Y) the following cards are necessary:

1 Header Card (NP = 1, NH2 = 80)

1 Plot Title Card

1 X-Axis Title Card

1 Y-Axis Title Card

MULTIPLE PLOTS

If two plots and a single Y-axis are desired, the cards indicated below are required:

1st Header Card (NP = 2, NH2 = 80)

1 Plot Title Card

1 X-Axis Title Card

1 Y-Axis Title Card

2nd Header Card (NP = 2, NH2 = 0)

For this case no additional title cards are necessary following the second Header Card.

When multiple plots <u>and</u> multiple Y-axes are desired, additional Y-axis cards are necessary. For example, if the user needs three Y-axes, he must use the following cards:

1st Header Card (NP = 3, NH2 = 80)

1 Plot Title Card

1 X-Axis Title Card

1st Y-Axis Title Card

2nd Header Card (NP = 3, NH2 = 80)

2nd Y-Axis Title Card

3rd Header Card (NP = 3, NH2 = 80)

3rd Y-Axis Title Card

EXAMPLE 9

```
TIME   LEM L 80 0 0 5 1 5  0.        0.        0.        0.
                      LEM X POSITION VS. TIME
                           TIME IN SECONDS
                           X POSITION IN FEET
LEM X  LEM Y 80 0 0 5 1 5  -5.E+6    +5.E+6    -5.E+6    +5.E+6
                             ORBIT PLOT
                           X POSITION IN FEET
                           Y POSITION IN FEET
TIME OMEGA Y 80 0 0 0 3 0  0.        0.        0.        0.
    BODY RATE X = LINE, Y = OCTAGON, Z = TRIANGLE
                             TIME IN SECONDS
                      BODY RATE IN RADIANS/SECOND
TIME OMEGA Y 0 0 0-5 3 1   0.        0.        0.        0.
TIME OMEGA Z 0 0 0-5 3 2   0.        0.        0.        0.
7 7
8 8
```

A $\substack{7\\8}$ $\substack{7\\8}$ EOF terminates the complete plot data. As can be seen, N data sets can be generated, where N = 8.

5.2 Standard Plot Input

Variables are plotted by the standard plot routine from a set of parameters output under user control (*PROGRAMS) by the MAIN Processor subroutine PLOTAP. If this output is desired, a $\substack{7\\9}$ LOAD, 45 and a $\substack{7\\9}$ RUN card, followed by the data set described below, is necessary.

5.2.1 Standard Plot Data Description

The standard plot routine plots from a set list (See Section 5.2.2) of variables and produces a set of CALCOMP graphs. Each graph may have up to three variables plotted against one independent variable. A job title card, one plot title card for each graph, the sets of variable request cards, and a job end card are required for standard plot output. Description of these cards and the associated data sets follow :

JOB TITLE CARD

One job title card is necessary for a set of graphs. It is the first card in the data deck.

Card Columns	Name	Description
1	IAST	Asterisk in the first column of the card.
2-73	ITITLE	Job title to be printed on each page of printed output and at the bottom of each plot.
75-79	IPRNT	Print flag; if tabular print desired, insert word "PRINT".

PLOT TITLE CARD

One plot title card is required for each graph. This card is the first card in each plot sequence.

Card Columns	Name	Description
1	IDS	Dollar sign in first column.
2-73	GTITLE	Grid title to be applied to this plot.

VARIABLE REQUEST CARD

One of these cards must appear for each variable in a particular plot. The first card is reserved for the independent variable and the ones that follow are for the dependent variables. At least two, but no more than four, of these cards must be present for each plot.

Card Columns	Name	Description
1-8	NAME	Variable name as taken from list of standard variables. Left justified.
10-29	TITLE	NAME to be applied to axis on which variable is presented.
35-49	MIN	Minimum value for variable; if blank, it will be computed. (E15.8).
55-69	MAX	Maximum value for variable; if blank, it will be computed. (E15.8).
79	IPL	Used to determine end of variable requests for particular plot. 1 - another variable follows 0 - last request is this sequence.

JOB END CARD

This card terminates data card input. It must be the last card in the data deck.

Card Columns	Name	Description
1	IDS	Dollar sign in column 1
2-4	END	Word "END"

Example 10 shows how the cards are to be arranged for standard plotting.

EXAMPLE 10

```
*  CASE 2.7 A  SECOND TPI                                                        PRIN
$CSM POSITION COORDINATES IN AGS COORDINATE SYSTEM
TIME      MISSION TIME (SEC)
RX        X COMPONENT (FEET)
RY        Y COMPONENT (FEET)
RZ        Z COMPONENT (FEET)
$CSM VELOCITY COORDINATES IN AGS COORDINATE SYSTEM
TIME      MISSION TIME (SEC)
VX        X COMPONENT (FT/SEC)
VY        Y COMPONENT (FT/SEC)
VZ        Z COMPONENT (FT/SEC)
$CSM POSITION COORDINATES IN INERTIAL COORDINATE SYSTEM
TIME      MISSION TIME (SEC)
RXI       X COMPONENT (FEET)
RYI       Y COMPONENT (FEET)
RZI       Z COMPONENT (FEET)
$CSM VELOCITY COORDINATES IN INERTIAL COORDINATE SYSTEM
TIME      MISSION TIME (SEC)
VXI       X COMPONENT (FT/SEC)
VYI       Y COMPONENT (FT/SEC)
VZI       Z COMPONENT (FT/SEC)
$LM POSITION COORDINATES IN AGS COORDINATE SYSTEM
TIME      MISSION TIME (SEC)
LRX       X COMPONENT (FEET)
LRY       Y COMPONENT (FEET)
LRZ       Z COMPONENT (FEET)
$LM VELOCITY COORDINATES IN AGS COORDINATE SYSTEM
TIME      MISSION TIME (SEC)
LVX       X COMPONENT (FT/SEC)
LVY       Y COMPONENT (FT/SEC)
LVZ       Z COMPONENT (FT/SEC)
$LM POSITION COORDINATES IN INERTIAL COORDINATE SYSTEM
TIME      MISSION TIME (SEC)
LRXI      X COMPONENT (FEET)
LRYI      Y COMPONENT (FEET)
LRZI      Z COMPONENT (FEET)
$LM VELOCITY COORDINATES IN INERTIAL COORDINATE SYSTEM
TIME      MISSION TIME (SEC)
LVXI      X COMPONENT (FT/SEC)
LVYI      Y COMPONENT (FT/SEC)
LVZI      Z COMPONENT (FT/SEC)
$TOTAL PROPELLANT FLOW RATE
AGS TIME AGS TIME (SEC)
WDOT      PROPELLANT FLOW RATE
$RANGE FROM LM TO CSM
TIME      MISSION TIME (SEC)
RRPR      ENVIRONMENTS
RR        AGS
$END
```

5.2.2 Standard Plot Variables

WD NO - NUMBERS IN THIS COLUMN REFER TO THE POSITION OF THE VARIOUS PLOT PARAMETERS IN THE OUTPUT RECORD OF THE PLOT TAPE.

PARAM NAME - MNEMONIC ASSOCIATED WITH EACH VARIABLE. USED TO REQUEST VARIABLE FOR PLOT PROCESSING.

UNITS - UNITS OF PARAMETER ON OUTPUT TAPE.

REMARKS - BRIEF DESCRIPTION OF PARAMETERS TO HELP IDENTIFY THEM.

WD NO	PARAM NAME	UNITS	REMARKS
1	TIME	SEC	PRESENT TIME
2	AGS TIME	SEC	AGS TIME
3	A11PR		LM COSINE MATRIX (STABLE MEMBER TO BODY AXIS
4	A12PR		TRASFORMATION - ENVIRONMENTS)
5	A13PR		
6	A21PR		
7	A22PR		
8	A23PR		
9	A31PR		
10	A32PR		
11	A33PR		
12	EX	RADIANS	AGS ATTITUDE CONTROL ERROR COMMANDS ABOUT THE
13	EY		X, Y, AND Z BODY AXIS
14	EZ		
15	EXP	ARC MIN	ALIGNMENT ERRORS ABOUT X, Y, AND Z BODY AXIS.
16	EYP		(AGS VS ENVIRONMENTS)
17	EZP		
18	A11		LM COSINE MATRIX (STABLE MEMBER TO BODY AXIS
19	A12		TRANSFORMATION - AGS)
20	A13		
21	A21		
22	A22		
23	A23		
24	A31		
25	A32		
26	A33		
27	DAXPR	RADIANS	ROTATIONAL DISPLACEMENTS OF THE VEHICLE ABOUT ITS
28	DAYPR	RADIANS	X, Y, AND Z AXIS LAST 20 MSEC (ENVIRONMENTS)
29	DAZPR	RADIANS	
30	DAX	RADIANS	ROTATIONAL DISPLACEMENTS OF THE VEHICLE ABOUT ITS
31	DAY	RADIANS	X, Y, AND Z AXIS LAST 20 MSEC (AGS)
32	DAZ	RADIANS	
33	DVXPR	FT/SEC	VELOCITY GAINED ALONG THE X, Y, AND Z BODY AXIS
34	DVYPR	FT/SEC	LAST 20 MSEC (ENVIRONMENTS)
35	DVZPR	FT/SEC	
36	DVX	FT/SEC	VELOCITY GAINED ALONG THE X, Y, AND Z BODY AXIS
37	DVY	FT/SEC	LAST 20 MSEC (AGS)
38	DVZ	FT/SEC	
39	VD1XPR	FT/SEC	COMPONENTS OF THE SENSED ACCUMULATED VELOCITY IN

40	VD1YPR	FT/SEC	BODY COORDINATES, UPDATED EVERY 40 MSEC
41	VD1ZPR	FT/SEC	(ENVIRONMENTS)
42	VD1X	FT/SEC	COMPONENTS OF THE SENSED ACCUMULATED VELOCITY IN
43	VD1Y	FT/SEC	BODY COORDINATES, UPDATED EVERY 40 MSEC (AGS).
44	VD1Z	FT/SEC	
45	TXBA	LBS	THRUST ALONG THE X, Y, AND Z BODY AXIS
46	TYBA	LBS	(ENVIRONMENTS)
47	TZBA	LBS	
48	INRGML	DEGREES	INNER, MIDDLE, OUTER GIMBAL ANGLES NECESSARY FOR
49	MIDGML	DEGREES	TRANSFORMATION TO BODY AXIS COORDINATES FROM
50	OUTGML	DEGREES	STABLE MEMBER COORDINATES
51	ACCEL	FT/S/S	MAGNITUDE OF TOTAL ACCELERATION (ENVIRONMENTS)
52	DELVS	FT/SEC	MAGNITUDE OF SENSED ACCUMULATED VELOCITY (ENV)
53	LHEUL1	DEGREES	EULER ROTATIONS REQUIRED TO TRANSFORM X, Y, AND Z
54	LHEUL2	DEGREES	AXIS INTO THE BODY AXIS SYSTEM
55	LHEUL3	DEGREES	
56	AT	FT/S/S	MAGNITUDE OF TOTAL ACCELERATION ALONG BODY AXIS-AGS
57	WT	LBS	TOTAL WEIGHT OF VEHICLE (ENVIRONMENTS)
58	DELVGX	FT/SEC	COMPONENTS OF THE VELOCITY YET TO BE GAINED (AGS)
59	DELVGY	FT/SEC	
60	DELVGZ	FT/SEC	
61	WDOT	LBS/SEC	TOTAL PROPELLANT FLOW RATE
62	ACCXBA	FT/S/S	ACCELERATIONS ALONG THE X, Y, AND Z BODY AXIS
63	ACCYBA	FT/S/S	(ENVIRONMENTS)
64	ACCZBA	FT/S/S	
65	RXPR	FEET	POSITION COORDINATES COMPUTED IN THE INERTIAL
66	RYPR	FEET	SYSTEM AND TRANSFORMED TO AN AGS ALIGNED
67	RZPR	FEET	COORDINATE SYSTEM -CSM (ENVIRONMENTS)
68	VXPR	FT/SEC	VELOCITY COORDINATES COMPUTED IN THE INERTIAL
69	VYPR	FT/SEC	SYSTEM AND TRANSFORMED TO AN AGS ALIGNED
70	VZPR	FT/SEC	COORDINATE SYSTEM -CSM (ENVIRONMENTS)
71	RX	FEET	POSITION COORDINATES IN THE AGS COORDINATE SYSTEM-
72	RY	FEET	CSM (AGS)
73	RZ	FEET	
74	VX	FT/SEC	VELOCITY COORDINATES COMPUTED IN THE AGS COORDINATE
75	VY	FT/SEC	SYSTEM - CSM (AGS)
76	VZ	FT/SEC	
77	RXI	FEET	POSITION COORDINATES IN AN INERTIAL COORDINATE
78	RYI	FEET	SYSTEM - CSM (ENVIRONMENTS)
79	RZI	FEET	
80	VXI	FT/SEC	VELOCITY COORDINATES IN AN INERTIAL COORDINATE
81	VYI	FT/SEC	SYSTEM - CSM (ENVIRONMENTS)
82	VZI	FT/SEC	
83	RPR	FEET	MAGNITUDE OF RADIUS VECTOR - CSM (ENVIRONMENTS)
84	GAMI	DEGREES	ANGLE BETWEEN LOCAL HORIZ. AND INST. VEL. VEC.-ENV.
85	PHI	DEGREES	GEODETIC LATITUDE - CSM
86	RDOTPR	FT/SEC	RADIAL VELOCITY - CSM (ENVIRONMENTS)
87	ALTPR	FEET	ALTITUDE ALONG RADIUS VECTOR ABOVE BODY-CSM (ENV)
88	VPR	FT/SEC	MAGNITUDE OF VELOCITY VECTOR - CSM (ENVIRONMENTS)
89	LAMDA	DEGREES	INST. LONGITUDE EASTWARD FROM GREENWICH - CSM
90	PHIP	DEGREES	GEOCENTRIC LATITUDE - CSM
91	VHPR	FT/SEC	HORIZONTAL VELOCITY - CSM (ENVIRONMENTS)
92	LRXPR	FEET	POSITION COORDINATES COMPUTED IN THE INERTIAL
93	LRYPR	FEET	SYSTEM AND TRANSFORMED TO AN AGS ALIGNED
94	LRZPR	FEET	COORDINATE SYSTEM - LM (ENVIRONMENTS
95	LVXPR	FT/SEC	VELOCITY COORDINATES COMPUTED IN THE INERTIAL
96	LVYPR	FT/SEC	SYSTEM AND TRANSFORMED TO AN AGS ALIGNED
97	LVZPR	FT/SEC	COORDINATE SYSTEM - LM (ENVIRONMENTS)
98	LRX	FEET	POSITION COORDINATES IN THE AGS COORDINATE SYSTEM-

99	LRY	FEET	LM (AGS)
100	LRZ	FEET	
101	LVX	FT/SEC	VELOCITY COORDINATES COMPUTED IN THE AGS COORDINATE
102	LVY	FT/SEC	SYSTEM - LM (AGS)
103	LVZ	FT/SEC	
104	LRXI	FEET	LM POSITION COORDINATES IN AN INERTIAL COORDINATE
105	LRYI	FEET	SYSTEM (ENVIRONMENTS)
106	LRZI	FEET	
107	LVXI	FT/SEC	VELOCITY COORDINATES IN AN INERTIAL COORDINATE
108	LVYI	FT/SEC	SYSTEM - LM (ENVIRONMENTS)
109	LVZI	FT/SEC	
110	LRPR	FEET	LM MAGNITUDE OF RADIUS VECTOR (ENVIRONMENTS)
111	LGAMI	DEGREES	LM ANGLE BETWEEN LOCAL HORIZONTAL AND INSTATANEOUS INERTIAL VELOCITY VECTOR (ENV)
112	LPHI	DEGREES	LM GEODETIC LATITUDE
113	LRDOTPR	FT/SEC	LM RADIAL VELOCITY (ENVIRONMENTS)
114	LALTPR	FEET	LM ALTIDUDE ALONG RADIUS VECTOR ABOVE BODY (ENV)
115	R	FEET	LM MAGNITUDE OF RADIUS VECTOR (ENVIRONMENTS)
116	LVPR	FT/SEC	LM MAGNITUDE OF VELOCITY VECTOR (ENVIRONMENTS)
117	LLAMDA	DEGREES	LM INST. LONGITUDE EASTWARD FROM GREENWICH
118	LPHIP	DEGREES	LM GEOCENTRIC LATITUDE
119	LVHPR	FT/SEC	LM HORIZONTAL VELOCITY (ENVIRONMENTS)
120	V	FT/SEC	MAGNITUDE OF THE VELOCITY VECTOR
121	A11BD		DIRECTION COSINE FOR DESIRED THRUST DIRECTION
122	A12BD		OF Z BODY AXIS (ENVIRONMENTS)
123	A13BD		
124	A31BD		
125	A32BD		
126	A33BD		
127	GRAVX	FT/S/S	GRAVITY ACCELERATION COMPONENTS ALONG X, Y, Z BODY
128	GRAVY	FT/S/S	AXIS (ENVIRONMENTS)
129	GRAVZ	FT/S/S	
130	ALT	FEET	ALTITUDE ALONG RADIUS VECTOR ABOVE BODY (AGS)
131	PERGPR	N.MI.	PERIGEE ALTITUDE ABOVE SPHERICAL REFERENCE BODY-ENV
132	APOGPR	N.MI.	APOGEE ALTITUDE ABOVE SPHERICAL REFERENCE BODY -ENV
133	DELVG	FT/SEC	MAGNITUDE OF VELOCITY YET TO BE GAINED (AGS)
134	PERG	N.MI.	PERIGEE ALTITUDE ABOVE SPHERICAL REFERENCE BODY-AGS
135	APOG	N.MI.	APOGEE ALTITUDE ABOVE SPHERICAL REFERENCE BODY -AGS
136	28J1	FT/SEC	VELOCITY DESIRED IN EXTERNAL DELTA-V MODE IN
137	28J2	FT/SEC	HORIZONTAL, OUT OF PLANE, AND RADIAL DIRECTION
138	28J3	FT/SEC	RESPECTIVELY
139	DQSX	FT/SEC	VELOCITY GAINED IN EXTERNAL DELTA-V MODE IN AGS
140	DQSY	FT/SEC	INTERNAL COORDINATES
141	DQSZ	FT/SEC	
142	INCL	DEGREES	ORBIT INCLINATION WITH RESPECT TO EQUAT. PLANE-AGS
143	DESNOD	DEGREES	ANGLE BETWEEN DESCENDING NODE AND LONG. OF X AXIS
144	ECC		ORBIT ECCENTRICITY
145	RPERG	FEET	PERIGEE RADIUS (ENVIRONMENTS)
146	VPERG	FT/SEC	PERIGEE VELOCITY
147	PERIOD		ORBIT PERIOD
148	RAPOG	FEET	APOGEE RADIUS (ENVIRONMENTS)
149	VAPOG	FT/SEC	APOGEE VELOCITY
150	TPERG	SEC	TIME TO PERIGEE (AGS)
151	ERX	FEET	NAVIGATION ERRORS IN POSITION COORDINATES OF AGS
152	ERY	FEET	COORDINATE SYSTEM
153	ERZ	FEET	
154	EVX	FT/SEC	NAVIGATION ERRORS IN VELOCITY COORDINATES OF
155	EVY	FT/SEC	AGS COORDINATE SYSTEM
156	EVZ	FT/SEC	

157	EVH	FT/SEC	LOCAL HORIZONTAL VELOCITY ERROR
158	EALTRT	FT/SEC	ALTITUDE RATE ERROR
159	EALT	FEET	ALTITUDE ERROR
160	RDOT	FT/SEC	RADIAL VELOCITY (AGS)
161	YPR	FEET	LM DISTANCE OUT OF CSM ORBITAL PLANE (ENVIRONMENTS)
162	VYOPR	FT/SEC	WEDGE VELOCITY (ENVIRONMENTS)
163	VH	FT/SEC	HORIZONTAL VELOCITY (AGS)
164	Y	FEET	LM DISTANCE OUT OF CSM ORBITAL PLANE (AGS)
165	WVEL	FT/SEC	WEDGE VELOCITY (AGS)
166	RRPR	FEET	RANGE FROM LM TO CSM (ENVIRONMENTS)
167	RRRTPR	FEET	RANGE RATE FROM LM TO CSM (ENVIRONMENTS)
168	RMSER	FEET	MAGNITUDE OF THE POSITION ERROR
169	RR	FEET	RANGE FROM LM TO CSM (AGS)
170	RRRT	FT/SEC	RANGE RATE FROM LM TO CSM (AGS)
171	RMSEV	FT/SEC	MAGNITUDE OF VELOCITY ERROR
172	RRE	FEET	RANGE ERROR
173	RRRTE	FT/SEC	RANGE RATE FROM LM TO CSM ERROR
174	XPCH	DEGREES	ANGLE BETWEEN X AND Z AXIS AND LOCAL HORIZONTAL
175	ZPCH	DEGREES	PLANE (ENVIRONMENTS)
176	WEDANG	DEGREES	ANGLE BETWEEN UNIT VECTOR PERPENDICULAR TO LM AND CSM ORBITAL PLANES (ENVIRONMENTS)
177	XAZ	DEGREES	ANGLES BETWEEN PROJECTION OF X AND Z BODY AXIS INTO
178	ZAZ	DEGREES	LOCAL HORIZONTAL PLANE AND PROJECTION OF INERIAL VELOCITY VECTOR INTO LOCAL HORIZONTAL PLANE
179	OMEGAX	DEG/SEC	ANGULAR VELOCITY VECTOR OF NAV BASE
180	OMEGAY	DEG/SEC	IN NAV BASE COORDINATES (ENVIRONMENTS)
181	OMEGAZ	DEG/SEC	
182	SALPAR		SINE OF EULER ANGLE ALPHA REGISTER USED FOR CONTROL OF TOTAL ATTITUDE DISPLAY (AGS)
183	CALPAR		COSINE OF EULER ANGLE ALPHA REGISTER USED FOR CONTROL OF TOTAL ATTITUDE DISPLAY (AGS)
184	SBETAR		SINE OF EULER ANGLE BETA REGISTER USED FOR CONTROL OF TOTAL ATTITUDE DISPLAY (AGS)
185	CBETAR		COSINE OF EULER ANGLE BETA REGISTER USED FOR CONTROL OF TOTAL ATTITUDE DISPLAY (AGS)
186	SGAMAR		SINE OF EULER ANGLE GAMMA REGISTER USED FOR CONTROL OF TOTAL ATTITUDE DISPLAY (AGS)
187	CGAMAR		COSINE OF EULER ANGLE GAMMA REGISTER USED FOR CONTROL OF TOTAL ATTITUDE DISPLAY (AGS)
188	EXR	FEET	ROTATIONAL ATTITUDE ERRORS ABOUT THE X,Y, AND
189	EYR	FEET	Z BODY AXIS REGISTERS USED FOR CONTROL OF
190	EZR	FEET	THE VEHICLE AND FOR DISPLAY (AGS)
191	ALTR	FEET	ALTITUDE REGISTER FOR CONTROL OF NAV DISPLAYS (AGS)
192	ALTRTR	FT/SEC	ALTITUDE RATE REGISTER FOR CONTROL OF NAV DISP -AGS
193	LATVR	FT/SEC	LATERAL VELOCITY REGISTER FOR CONTROL OF NAV DISPLAYS - AGS

6.0 SIMULATOR OUTPUT

6.1 MAIN Processor User Routines

Several routines have been written and incorporated in the AGS MAIN Processor for user controlled functions. The scheduling of these routines is controlled by the user via the *PROGRAMS Preprocessor option. Listed below are names of the routines currently available and a brief description of their functions.

- CHKPNTW - checkpoint write
  This routine will generate checkpoint dumps for restart of the simulator.

- HIGHFQ - high frequency standard print
  This routine prints out the standard high frequency print variables.

- LOWFQ - low frequency standard print
  This routine prints out the standard low frequency print variables.

- MEMDUMP - memory dump
  This routine provides an octal dump of AEA memory. (No delta time)

- PLOTAP - standard plot tape output
  This routine will dump out on tape the variables in the standard plot output for plotting in the PLOT Processor.

- TERMIN - termination routine
  This routine will terminate the MAIN Processor execution. (No delta time).

6.2 Standard Print Output

Parameters listed under Initialization Print will be printed at the beginning of the simulation run. Those parameters listed under High Frequency Print and Low Frequency Print are controlled by the routines HIGHFQ and LOWFQ, respectively.

6.2.1 Initialization Print

CSM ORBITAL PARAMETERS

APØGPR	PERGPR	RAPØG	RPERG	PERIØD
VAPOG	VPERG	ECC	INCL	DESNØD

LM RI TO SM EULER ANGLES

EULER1	EULER2	EULER3

6.2.2 High Frequency Print

LM

TIME		AGS TIME		
A11PR	A12PR	A13PR	EX	EXP
A21PR	A22PR	A23PR	EY	EYP
A31PR	A32PR	A33PR	EZ	EZP
A11	A12	A13	DAXPR	DAX
A21	A22	A23	DAYPR	DAY
A31	A32	A33	DAZPR	DAZ
DVXPR	DVX	VD1XPR	VD1X	TXBA
DVYPR	DVY	VD1YPR	VD1Y	TYBA
DVZPR	DVZ	VD1ZPR	VD1Z	TZBA
INRGML	LHEUL1	DELVGX	ACCXBA	E1
MIDGML	LHEUL2	DELVGY	ACCYBA	E2
ØUTGML	LHEUL3	DELVGZ	ACCZBA	STAGED
ACCEL	AT	EXT1	EXT2	DELVS
DEDAAD	DEDADT	WT	WDØT	
OMEGAX	OMEGAY	OMEGAZ		

AGS FLAGS, SWITCHES, AND REGISTERS

S00	S07	S10	S11	S14
S16	S17	S507	S623	DEL2
DEL5	DEL20	DEL21	MU8	MU10

AGS FLAGS, SWITCHES, AND REGISTERS (cont.)

BRNCON	DISCW1	DISCW2	SALPAR	CALPAR
SBETAR	CBETAR	SGAMAR	CGAMAR	EXR
EYR	EZR	ALTR	ALTRTR	LATVR
TELEMR	DEDA R	DEDA R		

6.2.3 Low Frequency Print

CSM

TIME

RXPR	RX	RXI	RPR	VPR
RYPR	RY	RYI	GAMI	LAMDA
RZPR	RZ	RZI	PHI	PHIP
VXPR	VX	VXI	DØTPR	VHPR
VYPR	VY	VYI	ALTPR	
VZPR	VZ	VZI		

LM

TIME AGS TIME

RXPR	RX	RXI	RPR	VPR
RYPR	RY	RYI	GAMI	LAMDA
RZPR	RZ	RZI	PHI	PHIP
VXPR	VX	VXI	RDØTPR	VHPR
VYPR	VY	VYI	ALTPR	
VZPR	VZ	VZI	R	V
A11BD	A31BD	GRAVX	ALT	DELVG
A12BD	A32BD	GRAVY	PERGPR	PERG

LM (cont.)

A13BD	A33BD	GRAVZ	APØGPR	APØG
28J1	DQSX	INCL	RPERG	RAPØG
28J2	DQSY	DESNØD	VPERG	VAPØG
28J3	DQSZ	ECC	PERIØD	TPERG
ERX	EVX	EVH	RDØT	VH
ERY	EVY	EALTRT	YPR	Y
ERZ	EVZ	EALT	VYOPR	VYO
RRPR	RR	RRE	WEDANG	
RRRTPR	RRRT	RRRTE	XPCH	XAZ
RMSER	RMSEV		ZPCH	ZAZ

6.2.4 Standard Print Output Definitions

NOTES: A. This letter following a definition indicates the related parameters are calculated in the ENVIRONMENTS simulation.

B. This letter following a definition indicates the related parameters are calculated in the AGS flight program.

Mnemonic	Definition
28J1, 28J2 28J3	Velocity desired in the external ΔV mode in the horizontal, out of plane, and radial direction, respectively. (feet/sec)
A11, A12 A13	LM cosine matrix (stable member to body axis transformation).[B]
A11BD, A12BD, A13BD	Direction cosine for the desired thrust direction of the X body axis.[B]
A31BD, A32BD, A33BD	Direction cosines for the desired thrust direction of the Z body axis.[B]

Mnemonic	Definition
A11PR, A12PR, A13PR, A21PR, A22PR, A23PR, A31PR, A32PR, A33PR	LM cosine matrix (stable member to body axis transformation).[A]
DELT2	Staged Flag
DEL 5	Altitude hold flag.
DEL 20	LOGIC flag for engine control.
DEL 21	Lunar surface flag.
E1	Flag indicating descent engine on or off.[B]
E2	Flag indicating ascent engine on or off.[B]
EXT1	Flag indicating that PGNCS commanding descent engine on or off.
EXT2	Flag indicating that PGNCS commanding ascent engine on or off.
MU8	Ullage counter.
MU10	Minor cycle counter.
S00	AGS function selector
S07	Reference frame selector for external ΔV mode.
S10	Guidance mode selector.
S11	RCS - DPS/APS selector.
S16	CDH Apsidal selector.
S17	CSI elliptical/circular orbit logic.
S507	Orient Z body axis to thrust axis.
S623	Crew selection of steering vector.
ACCEL	Magnitude of the total acceleration (ft/sec^2).[A]

Mnemonic	Definition
ACCXBA, ACCYBA, ACCZBA	Accelerations along the X, Y and Z body axis (feet/sec^2).[A]
AGS TIME	AGS time (sec).[B]
ALT	Altitude of vehicle above the surface of the reference body measured along the radius vector (feet)[B]
ALTPR	Altitude of vehicle above the surface of the reference body measured along the radius vector (feet)[A]
ALTR	Altitude register used for the control of the Navigation Displays.
ALTRTR	Altitude rate register used for control of the Navigation Displays.
APOG	Apogee altitude above the spherical reference body (nautical miles).[B]
APOGPR	Apogee altitude above the spherical reference body (nautical miles).[A]
AT	Magnitude of the total acceleration (ft/sec^2).[A]
BRNCON	Branch control output in octal.
CALPAR	Cosine of the Euler angle alpha register which is used for control of the Total Attitude Display.
CBETAR	Cosine of the Euler angle beta register which is used for control of the Total Attitude Display.
CGAMAR	Cosine of Euler angle gama register which is used for control of the Total Attitude Display.
DAX, DAY, DAZ	Rotational displacements of the vehicle about its X, Y, and Z axis last 20 msec (radians).[B]
DAXPR, DAYPR, DAZPR	Rotational displacement of the vehicle about its X, Y, and Z axis last 20 msec (radians).[A]

Mnemonic	Definition
DEDAAD	Data Entry and Display Assembly (DEDA) address last entry or readout.[B]
DEDADT	DEDA data last entry or readout.[B]
DEDA R	Data Entry and Display Assembly (DEDA) register, a four bit input and output register which communicates serially with the DEDA and the AEA.
DELVG	Magnitude of the velocity yet to be gained (feet/sec).[B]
DELVGX, DELVGY, DELVGZ	Components of the velocity yet to be gained (feet/sec).[B]
DELVS	Magnitude of the sensed accumulated velocity (feet/sec).[A]
DESNOD	Angle between the descending node and the longitude of the X axis in the AGS reference coordinate system (degrees).[A]
DISCW1	Discrete word 1, an 8 bit AEA input register.
DISCW2	Discrete word 2, a 7 bit AEA input register.
DQSX, DQSY, DQSZ	Velocity gained in the external ΔV mode in AGS inertial coordinates. (feet/sec).[B]
DVX, DVY, DVZ	Velocity gained along the X, Y, and Z body axis last 20 msec (feet/sec).[B]
DVXPR, DVYPR, DVZPR	Velocity gained along the X, Y, and Z body axis last 20 msec (feet/sec).[A]
EALT	Altitude error (feet).
EALTRT	Altitude rate error (feet/sec).
ECC	Orbit eccentricity.
ERX, ERY, ERZ	Navigation errors in position coordinates of AGS coordinate system (feet).

Mnemonic	Definition
EULER1, EULER2, EULER3	First, second and third Euler rotations required for transformations to Stable Member coordinates from Reference Inertial coordinates (degrees).
EVH	Navigation errors in velocity coordinates of AGS coordinate system (feet/sec).
EVX, EVY, EVZ	Navigation errors in velocity coordinates of AGS coordinate system (feet/sec).
EX, EY, EZ	AGS attitude control error commands about the X, Y, and Z body axis (radians).[B]
EXP, EYP, EZP	Alignment errors about X, Y and Z body axis. (arc min) (AGS vs Environments)
EXR, EYR, EZR	Rotational attitude errors about the X, Y, and Z body axis registers used for control of the vehicle and for display.
GAMI	Angle between the local horizontal and the instantaneous inertial velocity vector (deg).[A]
GRAVX, GRAVY, GRAVZ	Gravity acceleration components along the X, Y and Z body axis (feet/sec^2).[A]
INCL	Orbit inclination with respect to equatorial plane (degrees).[A]
INRGML, MIDGML, OUTGML	Inner, middle, outer Gimbal angles necessary for transformation to Body axis coordinates from Stable Member coordinates (degrees).
LAMDA	Instantaneous longitude of the vehicle measured positive eastward from the Greenwich Meridian (degrees).
LATVR	Lateral velocity register used for control of the Navigation Displays.
LHEUL1, LHEUL2, LHEUL3	Euler rotations required to transform from a local horizontal coordinate system, where the X axis is defined by $(\vec{R} \times \vec{V}) \times \vec{R}$ the Z axis is defined $-\vec{R}$, and the Y axis completes a right hand system, into the body axis system (degrees).

Mnemonic	Definition
OMEGAX, OMEGAY OMEGAZ	Angular velocity vector of nav base in nav base coordinates degrees/sec.[A]
PERG	Perigee altitude above the spherical reference body (nautical miles).[B]
PERGPR	Perigee altitude above the spherical reference body (nautical miles).[A]
PERIOD	Orbit period (seconds).
PHI	Geodetic Latitude (degrees).
PHIP	Geocentric Latitude (degrees).
RAPOG	Apogee radius (feet).[A]
R	Magnitude of the radius vector (feet).[B]
RDØT	Radial Velocity (feet/sec).[B]
RDØTPR	Radial Velocity (feet/sec).[A]
RMSER	Magnitude of the position error (feet).
RMSEV	Magnitude of the velocity error (feet/sec).
RPERG	Perigee radius (feet).[A]
RPR	Magnitude of the radius vector (feet).[A]
RR	Range from LM to CSM (feet).[B]
RRE	Range error (feet).
RRPR	Range from LM to CSM (feet).[A]
RRRT	Range rate from LM to CSM (feet/sec).[B]
RRRTE	Range rate from LM to CSM error (feet/sec).
RRRTPR	Range rate from LM to CSM (feet/sec).[A]
RX, RY, RZ	Position coordinates in the AGS coordinate system (feet).[B]
RXI, RYI, RZI	Position coordinates in an inertial coordinate system (feet).[A]

Mnemonic	Definition
RXPR, RYPR, RZPR	Position coordinates computed in the inertial system and transformed to an AGS aligned coordinate system (feet).[A]
SALPAR	Sine of the Euler angle alpha register which is used for control of the Total Attitude Display.
SGAMAR	Sine of Euler angle gamma register which is used for control of the Total Attitude Display.
STAGED	Flag indicating that the descent engine has been staged.[B]
TELEMR	Telemetry Register, a 24 bit shift register which is used to output serial telemetry data.
TIME	Present time (sec).[A]
TPERG	Time to perigee (sec).[B]
TXBA, TYBA, TZBA	Thrust along the X, Y and Z body axis (pounds).[A]
V	Magnitude of the velocity vector (ft/sec).[B]
VAPOG	Apogee velocity (ft/sec).
VD1X, VD1Y, VD1Z	Components of the sensed accumulated velocity in body coordinates, updated every 40 msec (ft/sec).[B]
VD1XPR, VD1YPR, VD1ZPR	Components of the sensed accumulated velocity in body coordinates, updated every 40 msec in AGS (ft/sec).[A]
VH	Horizontal velocity (ft/sec).[B]
VHPR	Horizontal velocity (ft/sec).[A]
VPERG	Perigee velocity (ft/sec)
VX, VY, VZ	Velocity coordinates computed in the AGS coordinate system (ft/sec).[B]

Mnemonic	Definition
VXPR, VYPR, VZPR	Velocity coordinates computed in the inertial system and transformed to an AGS aligned coordinate system (ft/sec).[A]
VXI, VYI, VZI	Velocity coordinates in an inertial coordinate system (ft/sec)[A]
Y	Wedge distance, LM distance out of CSM orbital plane (feet).[B]
YPR	Wedge distance, LM distance out of CSM orbital plane (feet).[A]
WDØT	Total propellant flow rate (pounds/sec).[A]
WEDANG	Wedge angle, angle between unit vectors perpendicular to CSM and LM orbital planes (degrees).[A]
VYO	Wedge velocity (ft/sec).[B]
VYOPR	Wedge velocity (ft/sec).[A]
XAZ	Angle between the projection of the X body axis into the local horizontal plane and the projection of the inertial velocity vector into the local horizontal plane (degrees).[A]
XPCH	Angle between X body axis and local horizontal plane. (degrees)[A]
ZAZ	Angle between the projection of the Z body axis into the local horizontal plane and the projection of the inertial velocity vector into the local horizontal plane. (degrees).[A]
ZPCH	Angle between Z body axis and local horizontal plane (degrees).[A]

6.3 Trace Format

The Trace consists of nine output parameters. They are as follows:

1) Instruction address (octal)
2) Instruction name
3) Operand address (octal)
4) Contents of the A-register (octal)
5) Contents of the Q-register (octal)
6) Contents of the operand address (octal)
7) Contents of the index register
8) The overflow indicator (1 means overflow)
9) Contents of the minor cycle counter (decimal)

These parameters represent the state after instruction execution.

EXAMPLE 11

LOC	NAME	OP.ADD	AREG	QREG	OP.	INDX	O.F.	CYCLE
7010	ADD	0760	377600	001000	210010	0	1	400

APPENDIX

THE READ DATA ROUTINE

Identification

READ DATA, Mnemonic Input Routine
CDC 3600/3800 COMPASS Subroutine
D. H. Hay and Bruce Johnson, May 20, 1964
Computation and Analysis Division
Manned Spacecraft Center

Acknowledgement

This subroutine is a direct translation of the IBMAP subroutine, SG RDAT, developed by Mr. Robert P. Crabtree of the Manned Spacecraft Center. The FØRTRAN calling sequences and input data card formats are identical for the two routines. Conversion algorithms and diagnostic procedures are similar.

Purpose

This subroutine allows the FØRTRAN 63 or CØMPASS user to enter data into his program at execution time without the use of, or the restrictions normally imposed by, formats. As many or as few variables as desired may be entered with each data set. Variables not entered will not be changed in value but will remain as they were before the subroutine was called. The data can be either punched in a free field or in a fixed field as all blanks are ignored. Legal modes include Integer, Real (either "F" type or "E" type), Complex, Double Precision, Alphabetic (BCD or HOLLERITH), and Octal. Optional exits are provided allowing the user to gain control if an error in the data cards or an end-of-file on the input tape is encountered.

Restrictions

This subroutine can be used only on the CDC 3600/3800 under the SCOPE/SCAMS Operating System. The size of the numbers and the accuracy retained on floating point numbers are limited only by the size of the word and the floating point/double precision hardware associated with the CDC 3600/3800.

Method

Standard double precision conversion techniques are employed throughout. Where a single precision value is desired, only the single precision result is kept, and in the case of the floating point numbers, the most significant portion is rounded before being stored.

Error checks are liberally distributed throughout the conversion code to insure that all ill-formed data cards are detected rather than improperly converted.

Usage

1. Calling Sequence

CALL RDATA $(ARG_1, ARG_2, \ldots, ARG_n)$

Where ARG_1, ARG_2, ..., ARG_n are variables or names of arrays into which data is to be entered. The name READ DATA may be substituted for RDATA in the above sequence.

2. Data Set

A Data Set is a card or a group of cards which will be read with one call of the RDATA subroutine. A Data Card is usually terminated after the last column (column 72) has been scanned but may be terminated prior to this by placing a slash (0-1) in the column where termination is desired. All punches to the right of the slash will be ignored, and the scan will continue with the first column of the next data card. A Data Set is terminated when the scan encounters a dollar sign (11-3-8); the rest of the data card is ignored and control is returned to the first executable statement after the "CALL RDATA" statement.

3. Data Fields

A Data Field is a continuous set of columns on a data card which causes one value to be entered into the calling program. A Data Field starts in column 1 of the data card or in the first column following the comma which has terminated the previous Data Field and is terminated by a comma, slash, dollar sign or the end of the data card (column 72). A Data Field cannot be split between two data cards and may assume one of the following forms:

1. MC(S) = N,
2. MC = N,
3. (S) = N,
4. N,
5. ,

where M specifies the Mode of the variable or array into which the data is to be entered as follows:

I	Integer
E or F	Real (Floating Point)
C	Complex
D	Double Precision

C — is an integer indicating into which variable in the argument list that data is to be entered (i.e., a "1" would indicate the variable which corresponds to ARG_1 a "2" to ARG_2, etc.)

S — is a positive non zero linear subscript indicating the element of the array associated with it. This subscript will be automatically doubled internally for complex and double precision numbers so that the linear subscript notation will agree with that used by FORTRAN 63.

N — is a numeric field, as described under Section 4.

A Data set must start with a Data Field of form (1) or (2) (see above). If form (2) is used for arrays, S will automatically be assumed to be 1. Once an array (MC) has been defined, subsequent fields of form (4) will cause the value of the fields to be entered into consecutive elements of the array defined. An empty field (two commas, punched in consecutive columns or separated by blank columns) will cause the corresponding array element to be left at its previous value (i.e., it will be skipped). The linear subscript may be redefined at any time by using form (3) without renaming the mode and argument count. As indicated above, all data fields should be separated by commas. There should not be a comma after the last data field on the card, however, because the end of a card acts as an implied comma and an additional comma would produce an empty field between it and the end of the card.

4. Numeric Fields

a. Decimal Fields

A decimal field has two components, the second of which is optional:

1. The Principal Part - a decimal number with or without a decimal point. The decimal point, if any, may appear at the beginning of, end of, or within the principal part. If omitted it is assumed to be located at the right-hand end of the principal part.

2. The Exponent Part - consists of the letter E or D followed by a signed or unsigned decimal integer. The Exponent part must follow the Principal Part. (Note: the "D" and "E" are completely interchangeable in this case.)

b. Octal Fields

An octal field consists of a string of not more than 16 (or 32 for double precision variables) octal digits (0 through 7) with or without a sign and followed by the letter Ø. A positive decimal integer N placed after the Ø will cause N zeros to be appended to the right of a positive octal number. The effect of a minus sign is to ones complement the entire word after the zeros are appended. For instance, -17Ø10 would appear as

7777607777777777

while 17Ø10 would appear as

0000170000000000

The number -6Ø15 would appear as

1777777777777777

whereas 6Ø15 becomes

6000000000000000

For double precision variables the minus sign will cause both the high and low order parts of the number to be complemented so that -101Ø15 would produce the two words

7777777777777767

6777777777777777

The high order part of the octal number is placed in the most significant portion of the double precision variable locations.

c. Alphabetic Fields

Alphabetic information may be read into Real or Complex variables (not Integer or Double Precision) through the use of an Alphabetic Field which may take on one of the following forms:

1. nHxxx...xxx,

2. Htxxx...xxxt,

Form (1) is recognized by the presence of a count (n) preceding the "H". In this case the n characters (here blanks are treated as legitimate alphabetic characters) following the "H" are taken as a hollerith string. Form (2) is recognized by the lack of a count (or a zero count) in front of the "H". In this case the character punched in the first column after the "H" is taken as a termination character (t) and all characters following the hollerith string.

Any legal hollerith character can be used as a termination character except, of course, the termination character cannot appear in the hollerith string.

Hollerith strings are broken down into eight characters per word and stored in ascending elements of the associated array. If the last word is not full, the characters are left adjusted and trailing blanks are added on the right. Alphabetic fields may be continued from card to card, column 1 being considered adjacent to column 72 of the preceding card. This is the only exception to the rule that fields cannot be split between cards. Alphabetic fields, like all other fields, should always be followed by a comma, slash, dollar sign, or end of card.

5. Special Features

To facilitate keypunching, the asterisk (11-4-8) may be used to denote subscription. That is MC*S = ... is equivalent to MC(S) = ..., is equivalent to (S)=....

The letter "K" may be used in place of the letter "Ø" in Octal fields to eliminate the confusion caused by the similarity between the letter "Ø" and the digit "0".

6. Alternate Exits

Normally, the RDATA subroutine, upon detecting an end-of-file on the input tape, will bring out the comment, "**ALL DATA PROCESSED**," and terminate execution through the subroutine EXIT. This action may be modified by the following sequence in the calling program:

```
ASSIGN i to n

CALL SETEOF (n)
```

where i is a statement number and n is a non-subscripted integer variable. After this sequence has been executed, an end-of-file on the input tape will cause control to be passed to statement number i without printing the comment. In addition, the input tape will be back-spaced so that subsequent entries to this (or any other input) routine will again cause an end-of-file condition. The sequence

```
CALL SETEOF (0)
```

will cause the end-of-file action to be reset to its normal state.

When a conversion error (illegal character, overflow, etc.) is detected, the normal procedure is to print out the bad card along with a comment describing the error condition, and then terminate execution through the subroutine Q8QERROR. This action may be modified by the following sequences in the calling program:

```
ASSIGN i to n

CALL SETERR (n)
```

After this sequence has been executed, an error will still cause the comments to be printed, but control will be passed to statement i rather than Q8QERROR. The sequence

```
CALL SETERR (0)
```

will cause the error action to be reset to its normal state.

7. Alternate Input Unit

Normally, logical unit sixty is used to input data cards. This condition may be modified by the following sequence, CALL SET UNIT (n)

This call will cause data to be read from logical unit n until the input unit is re-designated by another SET UNIT call.

8. Example

Program:

```
        COMPLEX CX1
        DOUBLE PRECISION TERM
        DIMENSION X(50), FMT(24), Y(50)
100     CALL RDATA (N, X, Y, CX1, TERM, FMT, ALPHA)
        WRITE (6, FMT)
...     ...     ...     ...     ...     ...     ...
```

Data Cards:

```
...     ...     ...     ...     ...     ...     ...
F6 = H*(16H1EXAMPLE PROGRAM)*
I1 = 3   /   NUMBER OF DATA POINTS FOR THIS CASE
E2 = 17.96347,,1.732E10, C4 = 1,2, E3 = -1.414, D5 =
3.14159265358979328
E3(5) = 1E-6, 1E-7,, 1E-9, (20) = 1E-21      $
...     ...     ...     ...     ...     ...     ...
```

This program in conjunction with this data set will cause the following to be true:

```
        FMT (1) = (16H1E
        FMT (2) = XAMPLE
                                (alphabetic information)
        FMT (3) = PROGR
        FMT (4) = AM)
        N       = 3             (integer)
        X(1)    = 0.17963470 E 02
```

```
X(3)    = 0.17320000 E 11  (real)
CX1     = 0.10000000 E 01, 0.20000000 E 01 (complex)
TERM    = 0.31415926535897932 D 01    (double precision)
Y(1)    =-0.14140000 E 01
Y(5)    = 0.10000000 E -05
Y(6)    = 0.10000000 E -06 (real)
Y(8)    = 0.10000000 E -08
Y(20)   = 0.10000000 E -20
```

All other variables and array positions will remain as they were before RDATA was called.