HP 3000 Computer System

Sleuth Simulator Diagnostic Language

Reference Manual

HEWLETT
PACKARD

COMPUTER SYSTEMS DIVISION, 19447 PRUNERIDGE AVE., CUPERTINO, CALIF. 95014

Part No. 30341-90007
Index No. 3HDWR.070.30341-90007

Printed in U.S.A 5/81

NOTICE

The information contained in this document is subject to change without notice.

HEWLETT-PACKARD MAKES NO WARRANTY OF ANY KIND WITH REGARD TO THIS MATERIAL, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. Hewlett-Packard shall not be liable for errors contained herein or for incidental or consequential damages in connection with the furnishing, performance or use of this material.

LIST OF EFFECTIVE PAGES

The List of Effective Pages gives the date of the current edition and of any pages changed in updates to that edition. Within the manual, any page changed since the last edition is indicated by printing the date the changes were made on the bottom of the page. Changes are marked with a vertical bar in the margin. If an update is incorporated when an edition is reprinted, these bars are removed but the dates remain.

PRINTING HISTORY

New editions are complete revisions of the manual. Update packages, which are issued between editions, contain additional and replacement pages to be merged into the manual by the customer. The date of the title page of the manual changes only when a new edition is published. When an edition is reprinted, all the prior updates to the edition are incorporated.

CONTENTS

SECTION I - GENERAL INFORMATION

Paragraph Page

SECTION II - OPERATING INSTRUCTIONS

Paragraph Page

SECTION III - STATEMENT/COMMAND DESCRIPTIONS

Paragraph Page

GENERAL INFORMATION | SECTION I

1.0 INTRODUCTION

The Sleuth Simulator language simulates the HP 3000 Series II and III Sleuth programming language. The purpose of the Sleuth Simulator is to provide as many of the HP 3000 Series II/III statements as possible to a user of an HP 3000 HP-IB version computer system.

The simulator is writen in HP AID, a lower level language, and AID is written in SPL II. The simulator is actually a series of AID functions, which are a series of HP AID statements, and simulates each particular Sleuth statement defined in this manual. The simulator will maintain Sleuth's ability to run up to eight devices of various types concurrently.

Note that all the Sleuth commands and statements available for the HP 3000 Series II/III are not included in this manual.

1.1 HARDWARE REQUIREMENTS

The Sleuth Simulator can run on any HP 3000 HP-IB version computer system with the following minimum equipment:

- Memory - 256K bytes
- System Console
- Magnetic Tape Unit for cold loading DUSIII system.

1.2 SOFTWARE REQUIREMENTS

- Diagnostic Utility System III that includes AID and Sleuth Simulator.
- AID and Sleuth Simulator manuals

1.3 SLEUTH SIMULATOR LIMITATIONS

The Sleuth Simulator is a separate program written in the HP AID language. When you enter a Sleuth program, the Sleuth Simulator becomes a part of this program. With the Simulator becoming part of a user's program, the variables and word buffers normally available to an HP AID user program have been limited as follows:

Variables A through N are available

Word Buffers AA through NN are available

String Buffers &AA through &VV are available

All Reserved Variables are available

If you use any of the non-available simulation variables, word buffers, or string buffers, an error may be reported or the operation of your program could be adversly affected.

1.4 DISC LIMITATIONS

The Sleuth default mode for the file mask (13037 controller) in the HP 3000 Series II/III is cylinder mode. For HP 3000 HP-IB version computer systems, it will be "surface mode". This limitation is created by AID's inability to distinguish a difference between a parameter of zero and an omitted parameter (both appear as zero). For example; if the following statement is entered,

```
RDI 0,AA(0),0
```

the simulator will set the file mask to zero. If the last zero is not entered at all, AID will still pass the simulator a zero. Therefore, the simulator cannot distinguish between an omitted parameter and a zero (0).

2.0 LOADING SLEUTH SIMULATOR

The Sleuth Simulator program (written in the AID language) is physically located on the Diagnostic Utility System III (DUSIII) cold load tape under the file name SLEUTHSM. To execute the Sleuth Simulator program, perform the following procedure:

1. Cold load the DUSIII tape and press the console RETURN key.
2. Once the DUS program has output its title message and prompt (:) enter, "AID".
3. AID will respond with a prompt character (>) and line number:

   ```
   >10
   ```

4. Enter "LOAD SLEUTHSM". The Sleuth Simulator is now loaded and you may enter your program statements or commands.

2.1 ENTERING A SLEUTH PROGRAM

The simulator program will occupy lines 0000-4990, leaving 5000-9999 for user program entry. Note that the simulator will become part of the program entered.

2.2 DELETING A SLEUTH PROGRAM

The DELETE command must be used to erase lines of code generated by your entries. It will erase only the lines specified:

```
  D(elete) 5000/5100
```

To erase both the Sleuth Simulator and your program, use the EP command. If this occurs inadvertantly, you can load the simulator again by entering "LOAD SLEUTHSM".

All commands and statement descriptions can be found in Section III of this manual.

2.3 PROCEED MODE

The Sleuth Simulator does not turn off the proceed mode at any time. A user should use this HP AID statement with caution. Refer to the HP AID manual for more information on this statement.

2.4 LISTING PROGRAMS, BUFFERS, OR VARIABLES

A copy of the Sleuth Simulator and/or the Users Program may be obtained by use of the AID "LIST" command. To list the Sleuth Simulator program and the users program, enter the following on the next available line:

```
>6010 L
```

To list the Sleuth Simulator program, enter the following on the next available line:

```
>6020 L 1/5000
```

To list the users program, enter the following on the next available line:

```
>6010 L 5000/6000
```

Variables, AID reserved words, and buffers can be listed by the use of the following entries:

```
>6010 L V            (Lists all variables)

>6010 L V,C          (Lists variable C)

>6010 L R            (Lists all reserved words)

>6010 L R,PASSCOUNT  (Lists the contents of the reserved
                     word PASSCOUNT)

>6010 L B,AA         (Lists the entire buffer AA)

>6010 L !B,AA,1/10   (Lists words 1-10 of buffer AA in
                     HEX format)
```

Refer to the AID manual for more detailed information on the use of the List command.

2.5 PROGRAM EXECUTION

A user can execute their program by using the AID "RUN" command as shown in the following example:

```
>6050 BUMP C
>6060 NEXT 5020
>6070 RUN
```

A users program can be terminated programmatically with the use of the AID "END" command. A user can also stop execution of a program, at the console, by entering Control Y. This will place them in the AID entry mode.

STATEMENT/COMMAND STATEMENTS — SECTION III

3.0 INTRODUCTION

The following pages in this section will describe the capabilities of each simulated sleuth statement. Statements that have the same mnemonic as an HP AID statement, command, reserved variable, or buffer name (AA-NN), that are being simulated, will have an S preceeding the original mnemonic (i.e., compare buffer (CB) will become (SCB).

NOTE

> All buffers and variables must be in upper case. The simulator will not recognize lower case letters.

Functions that differ from the original Sleuth statement will either describe the difference or refer to an HP AID equivalent statement that will perform that specific task.

The syntax for each of the following statements defines what the parameters of the statement are. If a parameter is optional it will be enclosed by brackets (i.e., SEEK lun [,cylinder,head,sector]). If SEEK 3 is entered in a user program, then a seek for logical unit 3 to cylinder 0, head 0, sector 0 would be issued. The parameters that are not enclosed by brackets are required inputs. If any parameter is not entered, the default for that particular parameter is 0. This implies that a SEEK statement, by itself, will issue a seek for logical unit 0, to cylinder 0, head 0, sector 0.

The statement descriptions on the following pages are presented in alphabetical order.

3.1 STATEMENT/COMMAND SECTION INDEX

* HP AID STATEMENTS

Statement	Description	Page	
ID	Initialize data	33	
IDI	Initialize data immediate	35	
IF-THEN	Conditional program branch true	37	*
IFN-THEN	Conditional program branch false	38	*
INPUT	Input data	39	*
IS	Incremental Seek	40	
IT	Increment track	41	
LET	Assignment of variables	42	*
LOOPTO	Conditional Loop Branch	43	*
MC	Master Clear	44	
NEXT	End of For-Step-Until-Next loop	45	*
PAUSE	No-error Pause	46	
PE	Pause on Error	47	
POLL	Resume polling devices	48	
PRINT	Print to console without pause	49	*
PROC	Proceed without end of Chan Prog	50	*
RAND	Randomize	51	*
RC	Recalibrate	52	
RD	Read data (Disc)	53	
RD	Read Data (Tape)	55	
RDA	Request Disc Address	56	
RDB	Randomize Define Buffer	57	
RDI	Read Data Immediate	58	
REW	Rewind tape	60	
REWOFF	Rewind Offline	61	

* HP AID STATEMENTS

Statement	Description	Page	
RFS	Read Full sector	62	
RFSI	Read full sector immediate	63	
RP	Ripple Print	64	
RQST	Request status	65	
RRB	Read Record Backward	66	
RS	Random Seek	67	
RSA	Request Sector address	68	
RSYN	Request Syndrome	69	
RWO	Read with offset	70	
RWOI	Read with offset immediate	72	
RWV	Read without Verify	73	
RWVI	Read without Verify immediate	75	
SCB	Simulated compare buffer	77	
SEEK	Seek	78	
SELU	Select Unit	79	
SFM	Set File Mask	80	
SKRD	Seek and Read data	81	
SKWD	Seek and Write data	83	
SOUT	Switch output	85	
SST	Suppress Status	86	
STAT	Status Dump	87	
TIMEOUT	Channel Program Timeout flag	88	*

* HP AID STATEMENT

3.2 COMMANDS (AID/SLEUTH)

Refer to the AID Manual for commands, as all AID commands are valid for Sleuth programs.

3.3 STATEMENTS

General statements control system oriented data manipulation. Each statement description contains the formal name, the function name or mnemonic, the syntax of the statement, a parameter description, a description of the statements operation, and an example of the statement usage.

Statements provided in this manual include both Sleuthsm and some common AID statements. AID statements are included for convenience of the user. Refer to the AID manual for a complete set of available AID statements.

AR

==

FORMAL NAME: Address Record

FUNCTION NAME: AR

SYNTAX: >AR lun[,cylinder,head,sector]

PARAMETERS: lun - Logical unit number.

cylinder - cylinder address

head - head address

sector - sector address

OPERATION: Sets logical address specified in the cylinder, head, and sector parameters into 7910K and 13037 disc controllers only, and does not reposition the heads.

EXAMPLE:

```
>5000 DEV 0,6,1,10,0
>5010 AR 0,4,2,3
>5020 RDA 0
>5030 DISP 0,D
>5040 RUN
```

The above example uses the Address Record function to set the logical disc address into the disc controller. The Request Disc Address function and the Display function obtain and display the address.

ASSIGN

FORMAL NAME: Assign Data to Buffer

AID OPERATION NAME: ASSIGN data buffer(element)[,(repeat factor)],datal[,data2, , ,datan]

DESCRIPTION: Stores data into a data buffer. The word datal is stored into data buffer (element) and, if included data2 is stored in data buffer (element+1) and so on through datan which is stored in data buffer (element+n). If repeat factor is included the data pattern is repeated (repeat-factor) times.
Datal through datan must be numeric.

EXAMPLE(S):

```
>5000 DB AA,100,%55          .INITIALIZE AA TO %55
>5010 ASSIGN AA(50),5,10,15,20,25,30,35
      (AA(50)=5, AA(51)=10, . . . AA(56)=35)

>5010 ASSIGN AA(10),(10),!FF
      (AA(10) THROUGH AA(19))=!FF)

>5010 ASSIGN AA(80),(5),3,7
      (AA(80)=3, AA(81)=7, AA(82)=3, AA(83)=7 etc.)

>5010 LET A:=80,F:=5
>5020 ASSIGN AA(A),(F),3,7
      (Same as ASSIGN statement 5010 above)
```

BSF

FORMAL NAME: Backspace File

FUNCTION NAME: BSF

SYNTAX: >BSF lun

PARAMETER: lun - Logical unit number

OPERATION: This function issues a backspace file to a magnetic tape unit.

EXAMPLE:

```
>5000 DEV 1,5,1,10,0
>5005 FOR B:=0 UNTIL 10
>5010 GAP 1
>5020 WFM 1
>5030 NEXT 5005
>5035 FOR C:=0 UNTIL 9
>5040 BSF 1
>5050 NEXT 5035
>5060 REW 1
>5070 RUN
```

This example demonstrates how a BSF function might be utilized in a user program. Eleven file marks are written on the tape then the tape unit is backspaced 10 file marks.

BSR

===

FORMAL NAME: Backspace Record

FUNCTION NAME: BSR

SYNTAX: >BSR lun

PARAMETER: lun - Logical unit number

OPERATION: This function will cause the magnetic tape unit to backspace one record from its present position.

EXAMPLE:

```
>5000 DEV 0,5,1,10,0
>5010 RDB AA(0),128
>5015 FOR C:=1 UNTIL 10
>5020 WD 0,AA(0)
>5030 NEXT 5015
>5035 FOR D:=1 UNTIL 9
>5040 BSR 0
>5050 NEXT 5035
>5060 REW 0
>5070 RUN
```

This example writes 10 records (128 words) then backspaces through nine of them.

BUMP

===

FORMAL NAME: Bump Pass Counter

AID OPERATION NAME: BUMP[;][H]

DESCRIPTION: Increments the Reserved Variable PASSCOUNT (unless the H parameter, which inhibits PASSCOUNT from incrementing, is used) and then prints that pass count on the Console. The pass counter (Reserved Variable PASSCOUNT) is initialized to zero whenever the RUN command is issued. Printing may be suppressed by a SNPR command and, if the optional semi-colon follows BUMP, no return-line feed will be issued after the pass counter value is printed.

EXAMPLE(S):

```
>5000 BUMP H
>5010 RUN
```

System outputs "END OF PASS 0". Note that passcount is still 0 after the print because of the H parm.

```
>5000 BUMP;
>5010 PRINT "FOUND A BUG!!"
>5020 RUN
```

System outputs "END OF PASS 1 FOUND A BUG!!".

CHB

==

FORMAL NAME: Change Buffer

FUNCTION NAME: CHB

SYNTAX: >CHB buf(0),type

PARAMETERS: buf - Buffer to be changed. This parameter must be any buffer AA(0)-NN(0) where AA-NN define buffer name and (0) sets an HP AID pointer to the first element in the buffer.

type - Type of change.

TYPE	FUNCTION
A	Fill with address
R	Randomize
I	Increment
D	Decrement
S	Circular shift left (shifts bits within each element 1 place to left)
W	Circular word shift (shifts words within buffer 1st to last and all else moves down one position)

OPERATION: The CHB command will change the contents of the specified buffer.

NOTE: Buffer manipulation with this function is slow.

EXAMPLE:

```
5000 DEV 0,6,1,20,0
5010 DB AA,4096
5020 ASSIGN AA(0),(1024),%52525,%125252,%66666,%33333
5030 DB BB,4096,0
5040 FOR C:= 1 UNTIL 100
5050 FOR I:=0 TO 410
5060 WD 0,AA(0),7,I,0,0
5070 RD 0,BB(0),7,I,0,0
5080 SCB 0,AA(0),BB(0),5
5090 NEXT 5040
5100 CHB AA(0),R
5110 NEXT 5030
5120 RUN
```

This example uses the CHB function to randomize the data buffer AA. It writes the preassigned buffer AA on the first 32 sectors of surface 0 (head 0), reads and compares the data. The buffer is then randomized and the process is repeated 100 times.

CL

==

FORMAL NAME: Clear

FUNCTION NAME: CL

SYNTAX: >CL lun

PARAMETER: lun - Logical unit number.

OPERATION: The clear function pertains to disc drives only. It will clear any clock offset, clear status, clear the interface busy bit and wait for a new command.

EXAMPLE:

```
>5000 DEV 0,6,1,10,0
>5010 CL 0
>5020 RUN
```

The above example issues a clear to a disc connected to channel 6, device 1.

CLREAD

FORMAL NAME: Cold Load Read

SYNTAX: >CLREAD lun, buf(0)

PARAMETERS: lun - An HP 7976 Magnetic Tape Drive

buf - Buffer into which data will be read. This parameter must be any buffer (AA(0)-NA(0) where AA-NN is the buffer name and 0 specifies the beginning position within the buffer where data is placed.

OPERATION: Places the HP 7976 in its HP7070E emulator mode and one record is read. The tape must be positioned at Beginning-Of-Tape (BOT) and can be either a 1600 or 6250 BPI.

NOTE: The purpose of the 7970E emulator mode is to allow the HP 7976 to be a cold-load device on HP systems designed to load from an HP 7970E.

EXAMPLE:

```
>5000 DEV 0,5,1,10,0,6250
>5010 RDB AA(0),4096
>5020 DB BB,4096,0
>5030 WD 0,AA(0)
>5040 REW 0
>5050 CLREAD 0,BB(0)
>5060 SCB 0,AA(0),BB(0)
>5070 REW 0
>5080 RUN
```

CORB

FORMAL NAME: Correct Buffer

FUNCTION NAME: CORB

SYNTAX: >CORB lun,buf(x)

PARAMETERS: lun - Logical unit number; must be an HP Disc using the HP 13037 controller.

buf(x) - Buffer to be corrected. This parameter must be any buffer AA(x)NN(x) where AA-NN define buffer name and (x) sets an HP AID pointer to the element in the buffer specified by the user.

OPERATION: This statement will correct the data buffer specified by the buf parameter according to the last syndrome requested for the logical unit designated.

EXAMPLE:

```
>5000 DEV 0,6,2,10,0
>5010 DB AA,6144,%66666
>5020 DB BB,6144,0
>5030 FOR A:= 0 TO 822
>5040 SEEK 0,A,1,0
>5050 WDI 0,AA(0)        (Note disc is in surface
                          mode)
>5060 RD 0,BB(0),1,A,1,0
>5070 IF SS(0)=%7400 THEN 5110
                        (disc status word 1 and
                        2 is stored in SS(0) and
                        SS(1))
>5080 SCB 0,AA(0),BB(0),5
>5090 NEXT 5030
>5100 END
>5110 RSYN 0
>5120 CORB 0,BB(0)
>5130 GOTO 5090
>5140 RUN
```

This program writes one track of data on surface 1, reads and checks for possible correctable errors and then compares buffers. If a possible correctable data error occurs, the data buffer (BB) will be corrected if the request syndrome (RSYN) indicates that the data is correctable.

DB

==

FORMAL NAME: Define Buffer

AID OPERATION NAME: DB Name, Length [,assignment data]

DESCRIPTION: Declares a buffer with a two (alpha) character name (AA, BB, ...NN) and a buffer length up to the allowable space available* (see MAXMEMORY under Reserved Variables). The parameter length is interpreted as a numeric (0 will delete the buffer). The only assignment (data) allowed at declaration is a string assignment for string buffers (see example) or numeric or variable for data buffers, where the entire buffer is stored with that string, numeric, or variable value. Dynamic allocation of buffers is allowed, but may cause large overhead in execution time since existing buffers are "packed" to allow room for a new buffer. Dynamic allocation will leave existing element values unchanged.

EXAMPLE(S): >5000 DB AA,100 (Declares the buffer AA as 100 words long)

>5000 DB &AA,10 (Declares the string buffer &AA as 10 bytes long (note AA and &AA are separate buffers))

>5000 DB &CC,100,"START" (Each sequential 5 byte set of &CC contains the 5 ASCII characters "START")

>5000 DB CC,100,0 (Stores 0 in all 100 elements of CC)

>5000 DB CC,110 (Reallocates CC to 110 words, first 100 elements remain intact)

>5000 DB CC,0 (Deletes buffer CC)

NOTE

Unless you want to clear a buffer by re-defining it again, do not include this command inside LOOP construct statements. Once it has been executed the first time the program is run, it would consume time for no purpose relevant to the LOOP.

* The limit depends on the ammount of memory taken up by Sleuthsm and the users program. At present Sleuthsm provides a user with 12.8K of memory.

DEV

==

FORMAL NAME: Device

FUNCTION NAME: DEV

SYNTAX: >DEV lun,chan,dev,errs,unit,dens

PARAMETERS: lun - Logical unit number (0 to 7).

chan - Channel number device connected to (0 to 15).

dev - HP-IB device number (1 to 7).

errs - Maximum error count the device is allowed (1 to 999).

unit - Device unit number (0 to 7).

dens - magnetic tape density, pertinient to HP 7976 only. Must be 1600 or 6250 BPI. Default is 6250 BPI.

OPERATION: The Device statement allows the user to define the characteristics of a particular device and to assign an error count and logical unit number to that device. This function will test for boundaries on all parameters, see if the entered channel and device are present, identify the device, obtain the device type for the 79XX discs (13037 controlled) and store these parameters in buffer ZZ for future use. The DEV function buffer (ZZ) is structured as described in Appendix A. If any of these parameters exceed the boundaries or if a non-existent channel or device has been entered an error message is output to the console and the program ends.

EXAMPLE:

```
>5000 DEV 1,7,2,10,0
          -OR-
>5000 DEV 0,6,1,25,3
```

NOTE

Do not include this command inside LOOP construct statement's. Once it has been executed, the first time the program is run, it would consume time for no purpose.

DISP

FORMAL NAME: Display

FUNCTION NAME: DISP

SYNTAX: >DISP lun,type

PARAMETERS: lun - Logical unit number.

type	function
D	Disc Address
R	Requested Status
S	Sector Address
Y	Last Syndrome

OPERATION: This function will display the item specified in the type parameter for the lun indicated.

EXAMPLE:

```
>5000 DEV 1,6,2,10,0    (7920 disc)
>5010 RS 1
>5020 RQST 1
>5030 DISP 1,R
>5040 RUN
```

This program will issue a random seek to a 7920A disc, request the status after the seek completes and print the status on the console.

DS

FORMAL NAME: Decremental Seek

FUNCTION NAME: DS

SYNTAX: >DS lun[,cylinder,head,sector]

PARAMETERS: lun - Logical unit number.

cylinder - cylinder address

head - head address

sector - sector address

This function will do an initial seek to the location specified by the cylinder, head, sector parameters, default is 0,0,0. Each time the instruction is executed the cylinder will be decremented by 1 until it reaches cylinder 0. When this occurs the disc will seek to the maximum cylinder. This function updates the internal disc address.

NOTE

This function only operates on 7910K and 13037 disc controllers.

This function does not decrement a common cylinder table. It sets up a cylinder table based on the statement number making this function call. Everytime that statement # makes a call to this function it will decrement its unique table. NOTE: When using this function remember that all read, write, and verify operations update the 7906/20/25A discs internal address. If a write operation of 128 words started at cylinder 100,0,0 and you issued a read command following it, the read would begin at cylinder 100,0,1 If a decremental seek was issued before the write operation, another decremental seek would be issued before the read to properly position the heads.

The maximum number of DS function calls (separate entries) allowed for each program is ten (10).

EXAMPLE:

```
>5000 DEV 0,6,1,10,0
>5010 DS 0,822,0,0
>5020 GOTO 5010
>5030 RUN
```

ES

===

FORMAL NAME: Enable Status

FUNCTION NAME: ES

SYNTAX: >ES

OPERATION: Enable Status will enable automatic checking of device status when utilizing Sleuth simulated statements (HP AID Functions).

NOTE

Enable status is an automatic default function of Sleuthsm and therefore, need only be used if a Supress Status (SST) command had previously been issued.

EXAMPLE:

```
>5000 DEV 0,6,1,15,2
>5005 RDB AA(0),128
>5010 DB BB,128,0
>5015 FOR A:= 0 TO 99
>5018 FOR B:= 0 TO 822
>5020 WD 0,AA(0),1,A,0,63
>5060 IFN B=822 THEN 5080
>5070 SST              (suppress status)
>5080 RD 0,BB(0),1,A,0,63
>5090 SCB 0,AA(0),BB(0),4
>5100 DB BB,128,0 (zero out buffer BB)
>5110 ES
>5120 NEXT 5015
>5130 NEXT 5018
>RUN
```

This example will test the last sector (63) on surface zero for a 7925A disc. The disc file mask (1) is set (line 5020) to allow the unit to increment beyond the end of cylinder. A test is made for cylinder 822. When cylinder 822 is reached, the status is suppressed for the read operation. This is required to prevent a seek check status error that will occur because of the buffering scheme of the 12745A disc interface. Refer to the RDA function for further information. Status checking is then enabled and this process is repeated for 99 more times.

ESTA

FORMAL NAME: Expected Status

FUNCTION NAME: ESTA

SYNTAX: >ESTA [status1[,mask[,status2[,mask[,status3[,mask]]]

PARAMETERS: status1 - First status word for discs, line printers or first two bytes of status for Mag Tape.

mask - A word of don't care bits. A 1 in the mask corresponds to a don't care bit in the status

status2 - Second status word for discs or third byte of status for Mag Tape. Mag Tape status byte is left justified (bits 0-7).

status3,mask - Similar to above except for HP 7976 only.

NOTE

If either the status or "mask" parameters are omitted, the omitted parameter will be zero

OPERATION: This statement changes the expected status of the next statement which utilizes status checking.

EXAMPLE:

```
>5000 DEV 2,6,2,10,0
>5010 ESTA !1300,,!8604,!6100
>5020 SEEK 2,150,3,0
>5030 GOTO 5010
>5040 RUN
```

In this example a 7925A disc will continue to seek to cylinder 150, head 3, sector 0. The expected status is set for a status 2 error of seek check. If the seek actually completes, then the following message will appear on the console:

```
79XX DISC STATUS WORD 1                     WORD 2
                 ------                     ------

STATUS IS  0 0 0 11111 0000 0000 / 0 00 0011 0 0 0 0 0 0 0 0 0
SHOULD BE  0 0 0 10011 0000 0000 / 1 XX 0011 X 0 0 0 0 0 1 0 0

CYLINDER = 0, SECTOR = 0, HEAD = 0
```

FMT

==

FORMAL NAME: Format

FUNCTION NAME: FMT

SYNTAX: >FMT lun, DTRACKS

PARAMETER: lun - Logical unit number.
DTRACKS - No. of tracks that have been flagged defect (7902 only)

OPERATION: This function will format a moving head disc (HP 7902, 7905,7906,7910K,7920 & 7925). It will also verify each track.

EXAMPLE:

```
>5000 DEV 0,6,1,10,0
>5010 FMT 0
>5020 RUN
```

When the program begins execution the following is output to the console

```
*Begin Format

 End Format

 End of AID user program
```

* For a 7902 disc, the message "Begin Verifying Formatted Disc" will also appear on the console.

FOR-STEP-UNTIL

FORMAL NAME: For-Step-Until

AID OPERATION NAME: F[OR] exp [STEP exp] UNTIL(or TO) terminator exp

DESCRIPTION: Provides a means of repeating a group of instructions between the FOR statement and a subsequent NEXT statement using a variable as a counter (the variable cannot be a buffer element). The STEP parameter is an optional increment of the FOR variable with a default of 1. The FOR-NEXT sequence is repeated until terminator expression value is exceeded by FOR variable value. FOR statements may be nested. Note that no execution occurs in the FOR statement after the initial execution. Note also that UNTIL or TO may precede the terminator expression but UNTIL will always be listed.

EXAMPLE(S):

```
>5000 FOR I:=5 TO 50
         |
         |
>5060 NEXT 5000
```

This for statement will execute the statements between 5000 and 5060 (46 times) with I=5 through 50 stepping one at a time.

```
>5000 FOR I:=5 STEP 8 UNTIL 50
         |
         |
>5060 NEXT 5000
```

This FOR statement will execute the statement between 5000 and 5060 (6 times) with I=5,13, 21,29,37,45.

```
>5000 for i:=5 step B:=8 until C:=50
         |
         |
>5060 NEXT 5000
```

This statement sequence provides the same sequence of the above statments.

```
>5000 FOR AA(2):=-5 TO 50
         |
         |
>5060 NEXT 5000
```

Buffer element AA(2) will step -5,-4,-3,-2,-1, 0,1,......50.

*If the STEP value is negative the sequence will repeat until the FOR value is less then the UNTIL value.

FSF

===

FORMAL NAME: Forward Space File

FUNCTION NAME: FSF

SYNTAX: >FSF lun

PARAMETER: lun - Logical unit number

OPERATION: This function will move the tape forward to the next file on the tape.

EXAMPLE:

```
>5000 DEV 0,5,1,15,0
>5010 RDB AA(0),4000
>5015 FOR C:=0 UNTIL 10
>5020 WD 0,AA(0)
>5030 WFM 0
>5040 NEXT 5015
>5050 REW 0
>5055 FOR D:=0 UNTIL 9
>5060 FSF 0
>5070 NEXT 5055
>5080 RUN
```

This example writes 11 records of random data with a file mark after each, rewinds the tape, then forward spaces through 10 of them.

FSR

FORMAL NAME: Forward Space Record

FUNCTION NAME: FSR

SYNTAX: >FSR lun

PARAMETER: lun - Logical unit number

OPERATION: This function will move the tape forward one record.

EXAMPLE:

```
>5000 DEV 0,5,1,15,2
>5010 RDB AA(0),8000
>5015 FOR C:=1 UNTIL 10
>5020 WD 0,AA(0)
>5030 NEXT 5015
>5040 REW 0
>5045 FOR D:=1 UNTIL 8
>5050 FSR 0
>5060 NEXT 5045
>5070 REW 0
>5080 RUN
```

This example writes 11 records of random data on tape, rewinds the tape, then forward spaces through 9 of them.

GAP

==

FORMAL NAME: Gap

FUNCTION NAME: GAP

SYNTAX: >GAP lun

PARAMETER: lun - Logical unit number

OPERATION: This function will write a gap on the specified magnetic tape unit.

EXAMPLE:

```
>5000 DEV 1,5,1,20,2
>5010 GAP 1
>5015 REW 1
>5020 RUN
```

This example erases a 3 inch portion of magnetic tape and rewinds.

GET

FORMAL NAME: Get (Used to obtain logical unit information only)

FUNCTION NAME: GET

SYNTAX: >Get lun,C or D or E or U

PARAMETERS: lun - Logical unit number.

C = Channel number
D = Device number
E = Error count
U = Unit number

OPERATION: This statement will read from the console these parameters only. The HP AID statement (INPUT) provides the standard capability of receiving operator input from the console.

NOTE: AID statement INPUT can be interspersed with Sleuth Simulator statements as with most AID statements.

EXAMPLE:

```
>5000 PRINT "ENTER THE CHANNEL NUMBER"
>5010 GET 0,C
>5020 PRINT "NUMBER OF ERRORS?"
>5030 GET 0,E
>5040 PRINT "NUMBER OF PASSES?"
>5050 INPUT A
>5060 FOR I:=1 TO A
>5070 PRINT "PASS NUMBER";I
>5080 NEXT 5060
>5090 RUN

ENTER THE CHANNEL NUMBER
?5
--
NUMBER OF ERRORS?
?10
---
NUMBER OF PASSES
?2
--
PASS NUMBER 1
PASS NUMBER 2

END OF AID USER PROGRAM
```

This example shows how the GET statement may be used to dynamically obtain information from the operator. Note that the operator input is underlined.

GOTO

FORMAL NAME: GO TO (Unconditional Branch)

AID OPERATION NAME: GOTO Statement Number

DESCRIPTION: Allows the program to branch unconditionally to another statement number.

EXAMPLE:

```
>5060 GOTO 5010
```

The above statement transfers control to statement 5010.

ID

===

FORMAL NAME: Initialize Data

FUNCTION NAME: ID

SYNTAX: >ID lun,buf(0)[,mask[,flag[,cylinder,head,sector]]]

PARAMETERS: lun - Logical unit number.

buf - Buffer from which data is read, then written to disc. This parameter must be any buffer, AA(0)-NN(0),where AA-NN define buffer name and (0) sets an HP AID pointer to the first element in the buffer.

Note: For 7902 discs, buffer size should not exceed one sector (128 words).

cylinder - Disc parameters indicating starting location
head - of initialization operation. The heads are
sector - assumed to be positioned over the correct cylinder, and will not be repositioned by these parameters.

flag - Flags the track as:

S - Spare
P - Protected
D - Defective
N - Non-flagged

mask - Loads file mask on the 13037 controller only. The mask bits are:

Bits	Function
12	Incremental/decremental seek. If set and bit 15 is a 1, the cylinder address will be decremented when End-of-Cylinder; otherwise, incremented.
13	Allow sparing
14	Cylinder/surface mode. If set, a cylinder consists of all available surfaces; End-of-Cylinder is set when the last sector of the last surface has been transferred. In surface mode, End-of-Cylinder is set when the last sector of any surface has been transferred.

BITS	FUNCTION
15	Allow incremental/decremental seek.

Default mask is surface mode. Default flag is non-flagged. Default cylinder, head and sector is 0,0,0.

OPERATION: Initialize Data function will perform an initialize operation on all 79XX disc drives. The initialize operation will begin at the cylinder, head and sector designated and will continue until the word count of the buffer is exhausted. The designation of the cylinder, head and sector parameters will be accomplished in this function by an Address Record command to the 7910K and 13037 disc controller and a seek for the 7902 controller.

EXAMPLE:

```
>5000 DEV 0,6,1,10,0
>5010 DB AA,6144,0
>5020 SEEK 0,10,0,0
>5030 ID 0,AA(0),3,D,815,0,0
>5040 SEEK 0,815,0,0
>5050 ID 0,AA(0),3,S,10,0,0
>5060 SEEK 0,10,0,0
>5070 RDI 0,AA(0),7
>5080 GOTO 5060
>5090 RUN
```

In this example a 7920A disc will seek to cylinder 10, head 0, sector 0, flag the entire track defective and place the address for its spare in the address field of each sector. It will then seek to a spare track (815) and flag it as a spare and write the address of the defective track in its address field. A loop is then set up to test the sparing feature.

NOTE: A 7902 disc drive does not have spare tracks but a defective track can be made invisible to the controller by flagging it defective and formatting the diskette. This process reduces the total number of available tracks on that surface.

IDI

FORMAL NAME: Initialize Data Immediate

FUNCTION NAME: IDI

SYNTAX: >IDI lun,buf(0)[,mask[,flag]]

PARAMETERS: lun - Logical unit number

buf - Buffer from which data is read,then written to disc. length determines word count of the write. This parameter must be any buffer AA(0)-NN(0) where AA-NN define buffer name and (0) sets an HP AID pointer to first element in the buffer.

Notes:1. For 7902 discs, buffer size should not exceed one sector (128 words).

2. Multiple sector transfers on a 7910 disc should not cross a track boundary. Flag one track at a time.

flag - Flags the track as:
S - Spare
P - Protected
D - Defective
N - Non-flagged

mask - Loads file mask on the 13037 controller only. The mask bits are:

Bits	Function
12	Incremental/decremental seek. If set and bit 15 is a 1, the cylinder address will be decremented when End-of-Cylinder; otherwise, incremented.
13	Allow sparing
14	Cylinder/surface mode. If set, a cylinder consists of all available surfaces; End-of-Cylinder is set when the last sector of the last surface has been transferred. In surface mode, End-of-Cylinder is set when the last sector of any surface has been transferred.

BITS	FUNCTION
15	Allow incremental/decremental seek.

Default flag is non-flagged. Default cylinder, head, sector is 0,0,0. Default mask is surface mode.

OPERATION: This function will perform an initialize operation on a moving head disc. The internal disc address will be used as the starting point of the write.

EXAMPLE:

```
>5000 DEV 0,6,1,10,0
>5010 DB AA,6144,0
>5020 FOR C:=0 UNTIL 410
>5030 IS 0
>5040 IDI 0,AA(0)
>5050 IS 0,0,1,0
>5060 IDI 0,AA(0)
>5070 NEXT 5020
>5080 RUN
```

The above example formats the upper cartridge on a HP7906A disc.

IF THEN

FORMAL NAME: If-Then Control

AID OPERATION NAME: IF exp [[SPECIAL OPERATOR exp][SPECIAL OPERATOR exp]] THEN statement number

DESCRIPTION: Allows the executing program to evaluate "exp" and if it is true (non-zero)* to transfer control to the statement number specified. "Exp" may be a simple variable, data buffer element, assignment or expression. Expressions may be seperated by a special relational operator not allowed in any other expression. The allowable special operators are:

GT (greater than)
LT (less than)
GE (greater than or equal to)
LE (less than or equal to)
NE (not equal to)
EQ (equal to)

Each expression is evaluated and then tested (left to right) with the special operator. The result(s) of the special operator evaluation(s) is logically ANDed and if the overall result is true, control is transferred to the THEN statement. Up to three expressions are allowed.

EXAMPLE(S):

```
>5000 IF AA(2) THEN 5050 (If AA(2) is true
                          ,non-zero, go to 5050)

>5000 IF A OR B THEN 5030 (The expression "A or B"
                           is evaluated)

>5000 IF 14 LE A:=A+1 LE 20 THEN 5020
                       (Test if A+1 is between 14
                          and 20 INCLUSIVE)

>5000 IF A:=A+1 GE B:=B+1 GE C:=C+1 THEN 5200
                     (Test IF (A+1)>=(B+1)>=(C+1))

>5000 IF 1 LT B LT 100 THEN 5020
                 .TEST IF B IS BETWEEN 1 AND 100**.
```

* See IFN Statement for the reverse branch condition.
**Note that statement 5000 would not execute the same as IF 1<B<100 THEN 5020 which executes as "IF(1<B)<100 THEN 5020" where the result of 1<B will be -1 or 0.

IFN THEN

===

FORMAL NAME: IF-NOT-THEN

AID OPERATION NAME: IFN exp THEN statement

DESCRIPTION: Identical to the IF-THEN statement (see IF-THEN) except the expression "exp" is tested for falsity in determining if control is passed to the label "statement". The expression value is not altered by the NOT function.

EXAMPLES:

```
>5000 IF 1 LE A LE 14 THEN 5020
                    (If A is between 1 and 14
                    go to 5020)

>5000 IFN 1 LE A LE 14 THEN 5020
                    (If A is "NOT" between 1
                    and 14 go to 5020)

>5000 IF A THEN 5020  (If A<>0 go to 5020)

>5000 IFN A THEN 5020 (If A=0 go to 5020)
```

INPUT

==

FORMAL NAME: Input Data

AID OPERATION NAME: INPUT x,[y],...[n]
I x,[y],..[n]

DESCRIPTION: Provides capability of receiving operator input from the Console and assigning that input to a variable(s). x may be a simple variable, buffer element, string buffer or Reserved Variable. When executing, input prompts with a ? or ?? to signify an input is expected (see Special Characters). Each input value must be separated by a comma. See the Reserved Variable INPUTLEN for determining the character length of the input.

EXAMPLES:

```
>5000 INPUT A (value input from console is
               interpreted and then stored in A)

>5000 INPUT AA(2)
 (the console input will be stored in AA(2))

>5000 INPUT &BB(2,6)
 (5 characters are accepted from console and stored
 in string buffer BB starting at element 2 - string
 buffers must be used to contain ASCII characters)

>5000 INPUT A,B,C
 (3 numeric values, separated by commas are
 accepted from the console and stored in variables
 A, B, and C respectively)

>5000 INPUT A
 (1 numeric value is accepted from the console)
```

NOTE: If you fail to enter the correct amount of input parameters at the console, the INPUT function will output a double ?? until all parameters, called for by the INPUT statement, have been entered.

IS

==

FORMAL NAME: Incremental Seek

FUNCTION NAME: IS

SYNTAX: >IS lun[,cylinder,head,sector]

PARAMETERS: lun - Logical unit number

cylinder - cylinder address

head - head address

sector - sector address

OPERATION: This function will do an initial seek to the address specified in the cylinder, head and sector parameters. Each time this command is executed it will increment the cylinder address. This function updates the internal disc address. Default cylinder, head and sector is 0,0,0.

NOTE: This function is only valid when used for operations on 7910K and 13037 disc controller

This function does not increment a common cylinder table. It sets up a cylinder table based on the statment number making this function call. Only ten (10) IS function calls are allowed per program.

NOTE: All read, write, and verify operations update the 7906/20/25A disc internal address. Multiple use of this function can be used to position the 7906/20/25A discs before write and read operations. See example for the IT function for more information.

EXAMPLE:

```
>5000 DEV 0,6,1,15,0
>5010 FOR D:=1 UNTIL 2000
>5020 IS 0
>5030 NEXT 5010
>5040 RUN
```

The above example causes a moving head disc to execute one cylinder incremental seeks.

IT

==

FORMAL NAME: Increment Track

FUNCTION NAME: IT

SYNTAX: >IT lun[,cylinder,head,sector]

PARAMETERS: lun - Logical unit number

cylinder - cylinder address

head - head address

sector - sector address

OPERATION: This function will do an initial seek to the location specified in the cylinder, head and sector parameters. It will increment the head address by one each time this function is called. After the last head has been selected the next increment will proceed to the next cylinder. If the cylinder equals the last cylinder in the disc the function will seek to 0 and start over. This function updates the internal disc address.

This function is only valid when used in operations on 7910K and 13037 disc controllers.

This function does not increment a common track table. It sets up a track table based on the statement number making this function call. A maximum of ten (10) IT function calls per program are allowed.

EXAMPLE:
```
>5000 DEV 0,6,1,10,0
>5005 DB AA,6144
>5010 ASSIGN AA(0),(1536),%125252,%66666,%33333,%52525
>5020 DB BB,6144
>5030 FOR D:=1 UNTIL 4115
>5040 IT 0
>5050 WDI 0,AA(0)
>5060 DB BB,0
>5065 IT 0
>5070 RDI 0,BB(0),1
>5080 SCB 0,AA(0),BB(0),5
>5090 NEXT 5030
>5100 RUN
```

The above example indicates how this function may be used to test all surfaces on an HP 7920 disc. The increment track (IT) utilizes a separate counter for the write and read operation, thus assuring proper position of the heads before each write and read operation.

LET

FORMAL NAME: Assignment

AID OPERATION NAME: [LET] variable:= Any variable, numeric, expression or string.

DESCRIPTION: Allows assignment to a variable, data buffer or string buffer the value of any variable, numeric, expression or string. This Sleuthsm command, unlike Sleuth, requires the use of the colon ":".

EXAMPLE(S):

```
>5000 LET A:=10     .A IS ASSIGNED THE VALUE 10
>5000 LET C:=D+E    .C IS ASSIGNED THE SUM OF D+E.
>5000 LET AA(2):=!F   .ELEMENT 2 OF THE BUFFER AA IS
                      .THE HEXADECIMAL VALUE F.

>5000 LET A:=C:=4   .MULTIPLE VARIABLE ASSIGNMENTS
>5000 LET A:=4,B:=7 .MULTIPLE EXPRESSION ASSIGNMENT
                       ALLOWED.
>5000 LET AA(4):=B .ELEMENT 4 OF BUFFER AA IS GIVEN
                   .THE VALUE OF THE B VARIABLE.

>5000 LET &AA(5,9):="HELLO"
              .&AA(5,6)=HE, &AA(7,8)=LL,
              &AA(),10)=OO
>5000 A:=10         .IDENTICAL TO FIRST EXAMPLE
>5000 LET A:=B<C  .A=-1 if B<C else A=0
```

*The LET keyword may be omitted but a subsequent list will display it.

LOOPTO

FORMAL NAME: Conditional Loop Branch

AID OPERATION NAME: LOOPTO label

DESCRIPTION: Causes a branch to the statement specified in label if a LOOP Command was previously issued, otherwise no action occurs

EXAMPLE(S):

```
> 5100 SECTION 1,5200
        .
        .
        .
> 5200 SECTION 2,5500
        .
        .
        .
> 5500 LOOPTO 5100 . Go to 5100 if LOOP flag is
                   . set.
```

The following example shows how to set the LOOP flag:

```
   >5000 PAUSE  (allows loop flag to be set)
   >5010 DEV 0,6,1,10,0
   >5020 .....
     .
     .
   >5050 .....
   >5060 RUN
:LOOPON    (turns on loop flag)
:GO        (program executes)
```

MC

FORMAL NAME: Master Clear

FUNCTION NAME: MC

SYNTAX: >MC lun

PARAMETER: lun - Logical unit number.

OPERATION: This function will clear the specified unit by issuing a device clear. For the 2608A printer, a master clear will be sent and for the 2631A printer a clear 1 (device clear with parity enable) is issued.

EXAMPLE:

```
>5000 DEV 4,6,1,10,5
>5010 SST
>5020 SEEK 4,823   (Create a seek check)
>5030 ES
>5040 MC 4         (Clear device and status)
>5050 SEEK 4       (Seek to zero (0))
>5060 GOTO 5010
>5070 RUN
```

This example will continually loop forcing a seek check error on a 7920A disc. The status is suppressed during the error condition, enabled afterword and cleared out by the master clear (MC). The heads are then repositioned to cylinder 0, head 0, sector 0.

NOTE: Recalibrate will not work in place of the seek in line 5050 because the disc address (cyl. 823) is beyond the maximum limit.

NEXT

FORMAL NAME: End of For-Next loop

AID OPERATION NAME: NEXT x

DESCRIPTION: Specifies the end of a For-Next set of statements, where x must be the statement number of respective FOR statement (not a variable).

EXAMPLE(S):

```
> 5010  LET J:=5
> 5020  FOR K:=1 UNTIL 20
> 5030  LET BB(K):=J, J:=J+5
> 5040  NEXT 5020
```

This set of statements would store BB(1)=5, BB(2)=10 BB(20)=100.

PAUSE

FORMAL NAME: Non-Error Pause

AID OPERATION NAME: PAUSE

DESCRIPTION: Creates an unconditional pause in the execution of the user program. This statement is suppressed only by the SNPS command. After a prompt character (>) is printed on the console the operator may enter any valid command.

EXAMPLE(S):

```
> 5010  PAUSE
------
> 5020  RUN
------
>  (Enter any valid command)
-
```

PE

==

FORMAL NAME: Pause On Error

FUNCTION NAME: PE

SYNTAX: >PE lun

PARAMETER: lun - Logical unit number

OPERATION: This function will notify the user that an error has occurred and stop the execution of the users program.

Once the function has been executed it can only be defeated by an AID suppress non-error pause command (SNPS). Program can also be continued by typing GO.

EXAMPLE:

```
>5000 DEV 0,6.1,15,0
>5010 PE 0
>5020 FOR C:=1 UNTIL 4000
>5030 RS 0
>5040 NEXT 5020
>5050 RUN
```

POLL

==

FORMAL NAME: Poll

FUNCTION NAME: POLL

SYNTAX: >POLL lun

PARAMETERS: lun - Logical unit number.

OPERATION: This function causes the HP13037 disc controller to resume polling.

EXAMPLE:

```
>5000 DEV 0,6,1,10,0
>5010 POLL 0
>5020 RUN
```

PRINT

==

FORMAL NAME: Print to Console without Pause

AID OPERATION NAME: PR[INT] [string] [; (or ,)] [string] etc.

DESCRIPTION: Enables data, print spacing* or strings to be output to list device. This statement must be used to print non- error messages only (see EPRINT or PRINTEX for error message reporting). This PRINT will only be suppressed by the SNPR command. PRINT strings may be concatenated with (;) to suppress return- line feed or (,) which generates a return-linefeed.

EXAMPLE(S):

```
> 5010  PRINT "A";2;"BC","DE";3;"FGH"
> 5020  RUN
A  BC
DE   FGH

        -or-

> 5010  DB &AA,10,"ABCDEFG"
> 5020  PRINT &AA(3,6);2;&AA(0,2)
> 5030  RUN
DEFG  ABC
> 5030
```

* See PRINT SPACING under Special Characters.

PROC

==

FORMAL NAME: Proceed

AID OPERATION NAME: PROC[N]

DESCRIPTION: This statement is used to enable(or disable when the N is added) the proceed mode. AID normally waits for each Channel program to interrupt before continuing to the statement following the RSIO. This normal mode of having I/O with wait maybe changed to the proceed mode(i.e. I/O without wait) by using this state

EXAMPLE(S): (Assume that AA and BB are pre-defined Channel programs)

```
> 5000   PROC           .PERFORM I/O WITHOUT WAIT
> 5010  LET CHANNEL:=2
> 5020  RSIO AA        .START CHANNEL PROGRAM AA
> 5030  LET CHANNEL:=3
> 5040  RSIO BB        .START CHANNEL PROGRAM BB
> 5050  PROC N         .WAIT HERE FOR I/O TO FINISH
```

RAND

===

FORMAL NAME: Randomize

FUNCTION: RAND

SYNTAX: >RAND var

PARAMETER: var - A variable designated by a letter A thru N.

OPERATION: This function generates a positive random number and places it in the designated variable.

EXAMPLE:

```
>5000 RAND A
>5010 RAND B
>5020 RAND C
>5030 RUN
```

This example places a random number in the variables A,B & C.

RC

FORMAL NAME: Recalibrate

FUNCTION NAME: RC

SYNTAX: >RC lun

PARAMETER: lun - Logical unit number.

OPERATION: This function performs a recalibrate operation on a 7906/20/25A moving head disc. At the completion of this operation the heads are located at cylinder 0.

EXAMPLE:

```
>5000 DEV 2,6,1,10,0
>5010 FOR A:=1 UNTIL 50
>5020 SEEK 2,822,0,0
>5030 RC 2
>5040 NEXT 5010
>5050 RUN
```

The above example will seek a HP7920A disc to cylinder 822 then recalibrate. This process is repeated fifty times.

RD (DISC)

FORMAL NAME: Read Data

FUNCTION NAME: RD

SYNTAX: >RD lun,buf(0)[,mask[,cylinder,head,sector]]

PARAMETERS: lun - Logical unit number.

buf - Buffer into which data, from disc, is written. This parameter must be any buffer AA(0)-NN(0) where AA-NN define buffer name and (0) sets an HP AID pointer to the first element in the buffer.

cylinder - starting cylinder address
head - starting head address
sector - starting sector address

mask - Loads file mask on the 13037 controller only. The mask bits are:

Bits	Function
12	Incremental/decremental seek. If set and bit 15 is a 1, the cylinder address will be decremented when End-of-Cylinder; otherwise, incremented.
13	Allow sparing
14	Cylinder/surface mode. If set, a cylinder consists of all available surfaces; End-of-Cylinder is set when the last sector of the last surface has been transferred. In surface mode, End-of-Cylinder is set when the last sector of any surface has been transferred.
15	Allow incremental/decremental seek.

Mask default is surface mode. For the cylinder, head and sector parameter default is 0,0,0.

OPERATION: This function will perform a read operaton on the LUN indicated. For 7906/20/25A discs, it includes a Set File Mask command followed by a Seek command to the cylinder, head and sector parameters. The 7902 discs will just seek to the designated cylinder, head, and sector parameters.

At the completion of a 7906/20/25A disc read operation the internal disc address will be four sectors beyond the end of the last read operation. The buffering scheme of the 12745A interface, which has two 128 word buffers, reads three sectors worth of information before receiving an end of data from the CPU, which terminates the transfer. By the time the 12745A notifies the 13037A disc controller to stop the read, another sector has begun to be read. The disc has now read three sectors that were not transferred. The internal disc address is updated to point to the next sector. If a one sector read at cylinder 100, head 0, sector 0 were performed, the internal disc address will indicate cylinder 100, head 0, sector 4.

EXAMPLE:

```
>5000 DEV 0,6,1,10,0
>5010 RDB AA(0),6144
>5025 FOR C:=0 UNTIL 822
>5030 DB BB,6144,0
>5040 WD 0,AA(0),,C
>5050 RD 0,BB(0),1,C
>5060 SCB 0,AA(0),BB(0),4
>5075 NEXT 5025
>5080 RUN
```

This example writes, reads, and compares buffers of a random data pattern on surface zero of a 7920A disc.

RD (TAPE)

FORMAL NAME: Read Data

FUNCTION NAME: RD

SYNTAX: >RD lun,buf(0)

PARAMETER: lun - Logical unit number

buf - Buffer into which data will be read. This parameter must be any buffer AA(0)-NN(0) where AA-NN define buffer name and (0) sets an HP AID pointer to the first element in the buffer.

OPERATION: This function will perform a read operation on the lun specified.

EXAMPLE:

```
>5000 DEV 0,5,1,10,1
>5010 RDB AA(0),4000
>5020 DB BB,4000,0
>5030 WD 0,AA(0)
>5040 REW 0
>5050 RD 0,BB(0)
>5060 SCB 0,AA(0),BB(0),3
>5070 REW 0
>5080 RUN
```

This example indicates how a read data operation may be performed on magnetic tape. This program writes one 4000 word record on magnetic tape then reads and checks the data.

RDA

FORMAL NAME: Request Disc Address

FUNCTION NAME: RDA

SYNTAX: >RDA lun

PARAMETER: lun - Logical unit number.

OPERATION: This function will return the current disc address stored in the controller. If a data error occurred it contains the address of the current sector; if not, it contains the address of the next logical sector. This function may be used to determine where an error occurred during a verify or any other function which terminates with a error. The address can be displayed on the console with DISP function.

At the completion of a 7906/20/25A disc read operation the internal disc address will be four sectors beyond the end of the last read operation. The buffering scheme of the 12745A interface, which has two 128 word buffers, reads three sectors worth of information before receiving an end of data from the CPU, which terminates the transfer. By the time the 12745A notifies the 13037A disc controller to stop the read, another sector has begun to be read. The disc has now read three sectors that were not transferred. The internal disc address is updated to point to the next sector. If a one sector read at cylinder 100, head 0, sector 0 were performed, the internal disc address will indicate cylinder 100, head 0, sector 4.

EXAMPLE:

```
>5000 DEV 0,6,1,10,0
>5010 RS 0
>5020 RDA 0
>5030 DISP 0,D
>5040 RUN
```

This example utilizes the RDA function to obtain the last address from a moving head disc.

RDB

FORMAL NAME: Randomize Data Buffer

FUNCTION NAME: RDB

SYNTAX: > RDB name,length

PARAMETERS: name - Two letter buffer name.

length - Number of words allocated to buffer.

OPERATION: This function defines randomized data buffers only. Note: The HP AID statements (DB and ASSIGN) should be used for string buffers. This function is the same as the Sleuth DB statement for randomizing buffers. It does not provide the other features of the Sleuth DB statement. They can be implemented with AID commands. For example:

Function	HP AID Format
Define a data buffer (AA) and fill it with minus 1.	DB AA,100,-1
Define a string buffer (BB), 10 elements long	DB &BB,10
Define a data buffer (CC) with alternating data patterns of %33333 and %66666 for 100 words.	DB CC,100 ASSIGN CC(0),(50), %33333,%66666

Note: This function operates slowly (approx. 21 secs for 3972 words) due to the overhead required to access and modify a buffer.

EXAMPLE:

```
>5000 DEV 0,6,1,10,0
>5010 RDB AA(0),6144
>5020 FOR D:=0 UNTIL 822
>5030 SEEK 0,D,2,0
>5040 WDI 0,AA(0)
>5050 NEXT 5020
>5060 RUN
```

RDI
===

FORMAL NAME: Read Data Immediate

FUNCTION NAME: RDI

SYNTAX: >RDI lun,buf(0)[,mask]

PARAMETERS: lun - Logical unit number.

buf - Buffer into which data is read. Length determines word count of read. This parameter must be any buffer AA(0)-NN(0) where AA-NN define buffer name and (0) sets an HP AID pointer to the first element in the buffer.

mask - Loads file mask on the 13037 controller only. The mask bits are:

Bits	Function
12	Incremental/decremental seek. If set and bit 15 is a 1, the cylinder address will be decremented when End-of-Cylinder; otherwise, incremented.
13	Allow sparing
14	Cylinder/surface mode. If set, a cylinder consists of all available surfaces; End-of-Cylinder is set when the last sector of the last surface has been transferred. In surface mode, End-of-Cylinder is set when the last sector of any surface has been transferred.
15	Allow incremental/decremental seek.

Default mask is surface mode.

OPERATION: This function will perform a read operation with the internal disc address designating the starting point of the read. This function updates the internal address.

EXAMPLE:

```
>5000 DEV 0,6,1,10,0
>5010 DB AA,128,0
>5020 RDI BB(0),128
>5040 WD 0,BB(0),7,120,2,0
>5045 SEEK 0,120,2,0
>5050 RDI 0,AA(0),7
>5060 SCB 0,BB(0),AA(0),3
>5070 GOTO 5020
>5080 RUN
```

This program writes, reads and compares the continually changing data for cylinder 120, head 2, sector 0.

REW

FORMAL NAME: Rewind

FUNCTION NAME: REW

SYNTAX: >REW lun

PARAMETER: lun - Logical unit number

OPERATION: This function will issue a rewind command to the magnetic tape unit specified.

EXAMPLE:

```
>5000 DEV 0,5,1,15,0
>5005 DB &AA,128
>5010 ASSIGN &AA(0),(32),%123,%377,%345,0
>5015 FOR C:=1 UNTIL 10
>5020 WD 0,&AA(0)
>5030 NEXT 5015
>5040 REW 0
>5050 RUN
```

This example writes eleven 128-byte records of data then rewinds the tape unit with the REW function.

REWOFF

FORMAL NAME: Rewind And Reset

FUNCTION NAME: REWOFF

SYNTAX: >REWOFF lun

PARAMETER: lun - Logical unit number

OPERATION: This function will rewind and reset the specified magnetic tape unit.

EXAMPLE:

```
>5000 DEV 0,5,1,15,0
>5010 REWOFF 0
>5020 RUN
```

This example places mag tape unit 0 offline.

RFS

FORMAL NAME: Read Full Sector

FUNCTION NAME: RFS

SYNTAX: >RFS lun,buf(0)[,cylinder,head,sector]

PARAMETERS: lun - Logical unit number.

buf - Buffer into which data is read. Buffer length determines word count of read. This parameter must be any buffer AA(0)-NN(0) where AA-NN define buffer name and (0) sets an HP AID pointer to the first element in the buffer.

cylinder - cylinder address
head - head address
sector - sector address

OPERATION: This function will execute a full sector read operation on a 7910K and the 7906/20/25A discs. The heads will be positioned over the correct cylinder. The default cylinder, head, sector is 0,0,0. The length of the buffer determines the the word count of the read.

EXAMPLE:

```
>5000 DEV 0,6,1,10,0
>5010 DB AA,138,%52525
>5015 LET AA(0):=!80FE          (SYNC WORD)
>5020 DB BB,138,0
>5030 SEEK 0,123,0,0
>5040 WFS 0,AA(0),123,0,0
>5050 RFS 0,BB(0),123,0,0
>5060 SCB 0,AA(0),BB(0),3
>5070 RUN
```

In this example cylinder 123, head 0, sector 0 has had its address field written over by buffer AA. This track should be reformatted before proceeding.

RFSI

FORMAL NAME: Read Full Sector Immediate

FUNCTION NAME: RFSI

SYNTAX: >RFSI lun,buf(0)

PARAMETER: lun - Logical unit number.

buf - Buffer into which data is read. Buffer length determines word count of read. This parameter must be any buffer AA(0)-NN(0) where AA-NN defined as buffer name and (0) sets an HP AID pointer to the first element in the buffer.

OPERATION: This function will perform a full sector read operation on a 7910K and the 7906/20/25A discs. The length of the buffer determines the word count of the read. The internal disc address will be used for the starting point. The internal disc address may be set by a Seek or an Address Record command.

EXAMPLE:

```
>5000 DEV 0,6,1,10,0
>5010 DB AA,138,%125252
>5015 LET AA(0):=!80FE          (SYNC WORD)
>5020 DB BB,138,0
>5050 WFS 0,AA(0),150,1,5
>5055 SEEK 0,150,1,5
>5060 RFSI 0,BB(0)
>5070 SCB 0,AA(0),BB(0),5
>5080 RUN
```

This example issues a SEEK (from WFS function) to address 150,1,5 on disc, writes and reads full sectors, then compares the data. The disc will require formatting after the use of the WFSI function.

RP

FORMAL NAME: Ripple Print

CAUTION

Do not use the Ripple Print function with the HP 2680A Page Printer.

FUNCTION NAME: RP

SYNTAX: >RP lun,linelength

PARAMETERS: lun - Logical unit number.

linelength - Number of columns defining the area of ripple print.

OPERATION: This function will write write a ripple pattern on the lun indicated and continue until stopped with CNTRL Y or until 32767 lines have been printed.

EXAMPLE:

```
>5000 DEV 0,7,2,10,0
>5010 RP 0,132
>5020 RUN
```

This example would result in a 132 column ripple print on DEV 2.

RQST

==

FORMAL NAME: Request Status

FUNCTION NAME: RQST

SYNTAX: >RQST lun

PARAMETER: lun - Logical unit number.

OPERATION: This function will return two words of status from the disc controllers (status words 1 & 2). This status may be displayed using the DISP function. The status will be stored in buffer SS(0) AND SS(1) for any disc function error. This may be useful for user programs.

EXAMPLE:

```
>5000 DEV 0,6,1,10,0
>5010 SEEK 0,10,0,0
>5020 RQST 0
>5030 DISP 0,R
>5040 RUN
```

RRB

==

FORMAL NAME: Read Record Backward

FUNCTION NAME: RRB

SYNTAX: >RRB lun, buf(0)

PARAMETER: lun - logical unit number

buf - Buffer into which data will be read. This buffer must be any buffer AA(0) - NN(0) where AA-NN defines buffer name and (0) sets an HP AID pointer to the first element in the buffer.

OPERATION: This function will read from the last element (byte) in the record toward the first. The bits within the bytes will remain the same if the record is read backward or forward.

EXAMPLE:

```
>5000 DEV 0,5,1,5,0
>5010 DB AA, 1000, %125252
>5020 FOR C := 0 TO 9
>5030 DB BB, 1000,0
>5040 WD 0, AA(0)
>5050 RRB 0, BB(0)
>5060 SCB 0, AA(0), BB(0),4
>5070 NEXT 5020
>5080 RUN
```

This program will write one record, read it backwards and compare buffer 10 times.

Note: This function is useful when a record cannot be read because of a tape error. The data beyond the error can be recovered.

RS

==

FORMAT NAME: Random Seek

FUNCTION NAME: RS

SYNTAX: >RS lun

PARAMETER: lun - logical unit number.

OPERATION: This function will cause a moving head disc to seek randomly. This function will update the internal disc address.

EXAMPLE:

```
>5000 DEV 1,6,1,15,0
>5010 RS 0
>5020 GOTO 5010
>5030 RUN
```

RSA

FORMAL NAME: Request Sector Address

FUNCTION NAME: RSA

SYNTAX: >RSA lun

PARAMETER: lun - Logical unit number.

OPERATION: This function will return the logical sector address of the sector currently under the heads. This address may be displayed using the DISP function.

Note: This function does not apply to 7902 and 7910K

EXAMPLE:

```
>5000 DEV 0,6,1,10,0
>5010 RS 0
>5020 RSA 0
>5030 DISP 0,S
>5040 RUN
```

The following messages are output as a result of the above program executing:

```
Requested Sector Address for Logical Unit 0 is: 33

End of AID user program

>5040
```

RSYN

FORMAL NAME: Request Syndrome

FUNCTION NAME: RSYN

SYNTAX: >RSYN lun

PARAMETER: lun - Logical unit number.

OPERATION: This function will obtain a seven word syndrome from the HP13037 (disc controller). A request syndrome operation may be issued after any read or verify operation which terminates with a possible correctable data error. The seven words of information will be read into an internal buffer which may be displayed with the DISP function. The format of the syndrome returned is as follows:

WORD	DEFINITION
1	Status
2	Cylinder
3	Head/Sector
4	Displacement
5	Pattern 1
6	Pattern 2
7	Pattern 3

```
EXAMPLE: >5000 DEV 0,6,1,15,0
         >5010 DB AA,6144,%133333
         >5020 DB BB,6144,0
         >5030 FOR C:=1 UNTIL 5000
         >5040 IT 0
         >5050 WDI 0,AA(0),2
         >5055 IT 0
         >5060 RDI 0,BB(0),3
         >5070 IF SS(0) = %7400 THEN 5090 (unit # in status word)
         >5080 NEXT 5030
         >5085 END            .Terminates program
         >5090 RSYN 0
         >5100 DISP 0,Y
         >5110 RUN
```

NOTE: When attempting to use an entire cylinder on a read operation, the controller will attempt to read beyond the end of cylinder because of the buffering in the controller. The file mask must therefore be set to increment (file mask=1, 3, or 7).

This example writes and reads data on a HP7920 disc. If a correctable data error is detected, the syndrome is requested and displayed.

RWO

FORMAL NAME: Read With Offset

FUNCTION NAME: RWO

SYNTAX: >RWO lun,buf(0),mask,offset[,cylinder,head,sector]

PARAMETER: lun - Logical unit number.

buf - Buffer into which data is read. Buffer length determines word count of read. This parameter must be any buffer AA(0)-NN(0) where AA-NN define buffer name and (0) sets an HP AID pointer to the first element in the buffer.

mask - Loads file mask on the 13037 controller only. The mask bits are:

Bits	Function
12	Incremental/decremental seek. If set and bit 15 is a 1, the cylinder address will be decremented when End-of-Cylinder; otherwise, incremented.
13	Allow sparing
14	Cylinder/surface mode. If set, a cylinder consists of all available surfaces; End-of-Cylinder is set when the last sector of the last surface has been transferred. In surface mode, End-of-Cylinder is set when the last sector of any surface has been transferred.
15	Allow incremental/decremental seek.

offset - Contains cylinder offset and the separator clock information.

OFFSET WORD FORMAT

```
0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15
- - - - - - A D S - [   CYL OFFSET   ]
```

-	=	Don't care
A	=	Advance separator clock by 10 nanoseconds.
D	=	Delay separator clock by 10 nanoseconds.
S	=	Sign bit (offset direction)
CYL OFFSET	=	Range of offset (+63 to -63 moves heads from track center). Increment depends on drive type

cylinder	-	cylinder address
head	-	head address
sector	-	sector address

NOTE: Default for cylinder,head,sector is 0,0,0.

OPERATION: This function operates like a normal read except an offset word is transmitted to the drive before executing. The cylinder, head and sector parameters are passed to the disc controller with an Address Record command.

NOTE: This function is valid for 7906/20/25A moving head discs only. This function cannot read a spare track with offset. The offset information is lost when the controller seeks from a flagged defective track to its spare. This is a feature of the controller.

RWOI

FORMAL NAME: Read With Offset Immediate

FUNCTION NAME: RWOI

SYNTAX: >RWOI lun,buf(0),mask,offset

PARAMETER: lun - Logical unit number.

buf - Buffer into which data is read. Buffer length determines word count of read. This parameter must be any buffer AA(0)-NN(0) where AA-NN define buffer name and (0) sets an HP AID pointer to the first element in the buffer.

offset - Contains cylinder offset and the separator clock information.

```
                OFFSET WORD FORMAT
                ------------------
    0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15
    - - - - - - - A D S -  [  CYL OFFSET  ]


                - = Don't care
                A = Advance separator clock by
                    10 nanoseconds.
                D = Delay separator clock by
                    10 nanoseconds.
                S = Sign bit (offset direction)
       CYL OFFSET = Range of offset (+63 to -63
                    moves heads from track center).
                    Increment depends on drive type
```

OPERATION: This function executes like a normal read except offset word is sent to the disc before executing. Heads are assumed to be positioned when using this function.

NOTE: This function is valid for a 7906/20/25A moving head disc only and it cannot read a spare track with offset. The offset information is lost when the controller seeks from a flagged defective track to its spare. This is a feature of the controller.

RWV

FORMAL NAME: Read Without Verify

FUNCTION NAME: RWV

SYNTAX: >RWV lun,buf(0)[,mask[,cylinder,head,sector]]

PARAMETER: lun - Logical unit number.

buf - Buffer into which data is read. Buffer length determines word count of read. This parameter must be any buffer AA(0)-NN(0) where AA-NN define buffer name and (0) sets an HP AID pointer to the first element in the buffer.

mask - Loads file mask on the 13037 controller only. The mask bits are:

Bits	Function
12	Incremental/decremental seek. If set and bit 15 is a 1, the cylinder address will be decremented when End-of-Cylinder; otherwise, incremented.
13	Allow sparing
14	Cylinder/surface mode. If set, a cylinder consists of all available surfaces; End-of-Cylinder is set when the last sector of the last surface has been transferred. In surface mode, End-of-Cylinder is set when the last sector of any surface has been transferred.
15	Allow incremental/decremental seek.

cylinder - cylinder address

head - head address

sector - sector address

Note: Default for cylinder,head,sector is 0,0,0.
Mask default is surface mode.

OPERATION: This function operates like a normal read function but does not verify the preceding sector. No address checking or sparing operations occur unless a track boundary is crossed during the operation.

NOTE: This function is valid only for 7906/20/25A moving head discs only.

EXAMPLE:

```
>5000 DEV 0,6,1,10,0
>5010 DB AA,6144,0
>5020 FOR C:=0 TO 410
>5030 RWV 0,AA(0),2,C,0,0
>5040 NEXT 5020
>5050 RUN
```

This example uses the RWV function to read one entire surface from an HP 7906A disc.

RWVI

FORMAL NAME: Read Without Verify Immediate

FUNCTION NAME: RWVI

SYNTAX: >RWVI lun,buf(0)[,mask]

PARAMETERS: lun - Logical unit number.

buf - Buffer into which data is read. Buffer length determines word count of read. This parameter must be any buffer AA(0)-NN(0) where AA-NN define buffer name and (0) sets an HP AID pointer to the first element in the buffer.

mask - Loads file mask on the 13037 controller only. The mask bits are:

Bits	Function
12	Incremental/decremental seek. If set and bit 15 is a 1, the cylinder address will be decremented when End-of-Cylinder; otherwise, incremented.
13	Allow sparing
14	Cylinder/surface mode. If set, a cylinder consists of all available surfaces; End-of-Cylinder is set when the last sector of the last surface has been transferred. In surface mode, End-of-Cylinder is set when the last sector of any surface has been transferred.
15	Allow incremental/decremental seek.

NOTE: Default mask is surface mode.

OPERATION: This function operates like a normal read operation but does not verify the preceding sector. No address checking or sparing operations occur unless a track boundary is crossed during the operation. The starting point of the read will be taken from the internal disc address.

EXAMPLE:

```
>5000 DEV 0,6,1,10,0
>5010 DB AA,128,0
>5020 RS 0
>5030 RWVI 0,AA(0)
>5040 GOTO 5020
>5050 RUN
```

This example randomly seeks and uses the RWVI function to read one sector of information.

SCB

==

FORMAL NAME: Simulated Compare Buffer

FUNCTION NAME: SCB

SYNTAX: >SCB lun,bufl(0),buf2(0),errcount[,maxcount]

PARAMETERS: lun - Logical unit number of device being tested.

bufl(0),buf2(0) - Buffers of which the contents are to be word by word compared.

errcount - Maximum number of errors to be displayed for each execution.

maxcount - Maximum number of words to be compared. Uses smallest buffer length if defaulted.

OPERATION: The Compare Buffer command will compare word by word each element of buffers 1 and 2.

EXAMPLE:

```
>5000 DEV 0,6,1,10,0
>5005 DB AA,6144
>5010 ASSIGN AA(0),(1536),%125252,%52525,!FFFF,!AAAA
>5025 FOR C:=0 UNTIL 822
>5030 IT 0
>5040 WDI 0,AA(0),7
>5045 IT 0
>5050 VERI 0,48
>5060 DB BB,6144,0
>5065 IT 0
>5070 RDI 0,BB(0),7
>5080 SCB 0,AA(0),BB(0),4
>5090 NEXT 5025
>5100 RUN
```

The above example indicates how a compare buffer operation may be used to help evaluate the results of an operation. One track of information is written on a 7920A disc. It is then verified and compared until the entire disc is checked.

SEEK

FORMAL NAME: Seek

FUNCTION NAME: SEEK

SYNTAX: >SEEK lun[,cylinder,head,sector]

PARAMETERS: lun - Logical unit number.

cylinder - Disc parameters

head - for the moving

sector - head discs.

NOTE: Default cylinder,head,sector is 0,0,0.

OPERATIONS: This function will cause a disc to position its heads over the specified cylinder. This function also updates the internal disc address.

EXAMPLE:

```
>5000 DEV 0,6,1,10,0
>5010 SEEK 0
>5020 SEEK 0,200,3,23
>5030 SEEK 0,405,1,5
>5040 GOTO 5010
>5050 RUN
```

This example executes seek operations on a HP7906/20/25 disc.

SELU

FORMAL NAME: Select Unit

FUNCTION NAME: SELU

SYNTAX: >SELU lun,unit

PARAMETER: lun - Logical unit number

unit - Temporary unit select in the range of 0 to 3. Does not affect the logical unit number. The unit does not have to be on-line.

OPERATION: This function will select the unit specified in the unit parameter.

EXAMPLE:

```
>5000 DEV 3,5,1,10,3
>5010 SELU 3,1
>5020 RUN
```

This example defines the tape unit as 3, but it selects unit 1.

SFM

FORMAL NAME: Set File Mask

FUNCTION NAME: SFM

SYNTAX: >SFM lun,mask

PARAMETERS: lun - Logical unit number.

mask - Loads file mask on the 13037 controller only. The mask bits are:

Bits	Function
12	Incremental/decremental seek. If set and bit 15 is a 1, the cylinder address will be decremented when End-of-Cylinder; otherwise, incremented.
13	Allow sparing
14	Cylinder/surface mode. If set, a cylinder consists of all available surfaces; End-of-Cylinder is set when the last sector of the last surface has been transferred. In surface mode, End-of-Cylinder is set when the last sector of any surface has been transferred.
15	Allow incremental/decremental seek.

OPERATION: This function will set the file mask on the HP13037 disc controller from bits 8-15 of the mask parameter. When the disc controller is first powered up the mask is set to surface mode, where no sparing or automatic seeking is performed.

NOTE: Default for all functions using the file mask parameter is 0 (surface mode).

EXAMPLE:

```
>5000 DEV 2,6,1,15,3
>5010 SFM 2,7
>5020 RUN
```

This example loads the file mask on the 13037 disc controller with a value of 7.

SKRD

FORMAL NAME: Seek Read Data

FUNCTION NAME: SKRD

SYNTAX: >SKRD lun,buf(0)[,mask[,cylinder,head,sector]]

PARAMETERS: lun - Logical unit number.

buf - Buffer into which data is read. Buffer length determines word count of read. This parameter must be any buffer AA(0)-NN(0) where AA-NN define buffer name and (0) sets an HP AID pointer to the first element in the buffer.

mask - Loads file mask on the 13037 controller only. The mask bits are:

Bits	Function
12	Incremental/decremental seek. If set and bit 15 is a 1, the cylinder address will be decremented when End-of-Cylinder; otherwise, incremented.
13	Allow sparing
14	Cylinder/surface mode. If set, a cylinder consists of all available surfaces; End-of-Cylinder is set when the last sector of the last surface has been transferred. In surface mode, End-of-Cylinder is set when the last sector of any surface has been transferred.
15	Allow incremental/decremental seek.

cylinder - cylinder address

head - head address

sector - section address

NOTE: Default for cylinder,head,sector is 0,0,0.

OPERATION: This function will perform a seek to the specified location and read that data into the specified buffer. The internal disc address is updated by this function.

EXAMPLE:

```
>5000 DEV 0,6,1,10,2
>5010 DB AA,128,0
>5020 SKRD 0,AA(0),2,10,1,2
>5030 RUN
```

SKWD

FORMAL NAME: Seek write Data

FUNCTION NAME: SKWD

SYNTAX: >SKWD lun,buf(0)[,mask[,cylinder,head,sector]]

PARAMETERS: lun - Logical unit number.

buf - Buffer from which data is read, then written to disc. Buffer length determines word count of read. This parameter bust be any buffer AA(0) through NN(0), where AA-NN define buffer name and (0) sets an HP AID pointer to the first element in the buffer.

mask - Loads file mask on the 13037 controller only. The mask bits are:

Bits	Function
12	Incremental/decremental seek. If set and bit 15 is a 1, the cylinder address will be decremented when End-of-Cylinder; otherwise, incremented.
13	Allow sparing
14	Cylinder/surface mode. If set, a cylinder consists of all available surfaces; End-of-Cylinder is set when the last sector of the last surface has been transferred. In surface mode, End-of-Cylinder is set when the last sector of any surface has been transferred.
15	Allow incremental/decremental seek.

cylinder - cylinder address

head - head address

sector - sector address

NOTE: Default for cylinder,head,sector is 0,0,0.

OPERATION: This function will issue a seek to specified location and read the data, then write it into the specified buffer. The internal disc address is then updated.

NOTE: This function is only valid for 79XX discs that are controlled by the 13037 controller.

EXAMPLE:

```
>5000 DEV 0,6,1,10,0
>5010 RDB AA(0),4096
>5020 DB BB,4096,0
>5030 FOR C:=1 UNTIL 100
>5040 RAND I
>5050 LET A:=I MOD 411
>5060 LET B:=I MOD 4
>5070 LET C:=I MOD 48
>5080 SKWD 0,AA(0),7,A,B,C
>5090 SKRD 0,BB(0),7,A,B,C
>5100 IF SS(0) = %7400 THEN 5120
>5105 SCB 0,AA(0),BB(0),3
>5110 NEXT 5030
>5115 END                  .Terminates program
>5120 RSYN 0
>5130 DISP 0,Y
>5150 RUN
```

This example uses the SKWD and SKRD functions to test a HP7906 disc.

SOUT

FORMAL NAME: Switch Output

FUNCTION NAME: SOUT

SYNTAX: >SOUT

OPERATION: Switch output will output error messages to the lineprinter or the system console. Initially error messages will be output to the system console. Each SOUT statement will alternate the output device for error messages.

EXAMPLE:

```
>5000 DEV 0,6,1,10,1
>5010 DB AA,128,%052525
>5020 SOUT
>5030 FOR B:=0 TO 410
>5040 SEEK 0,B,0,0
>5050 WD 0,AA(0),7,B,0,0
>5070 NEXT 5030
>5080 RUN
```

The above example will switch the reporting of error messages from the console to the lineprinter. Note if a lineprinter is not connected to the system, the output will default to the console.

SST

FORMAL NAME: Suppress Status

FUNCTION NAME: SST

SYNTAX: >SST

OPERATION: This function will disable status checking for all statements following this function statement. Status checking can be re-enabled by using the Enable Status function (ES) of AID.

EXAMPLE:

```
>5000 DEV 5,6,1,10,3
>5010 DB AA,128,0
>5020 SST
>5030 RD 5,AA(0),1,822,8,47
>5040 PRINT "BUFFER AA(4)=",AA(4)
>5050 RUN
```

This example suppresses the status error that would normally occur when trying to read the last sector of a 7925A disc. Refer to the RDA function for more information on 12745A interface operation during a read. The fifth word of the sector is then displayed on the console.

STAT

==

FORMAL NAME: Status Dump

FUNCTION NAME: STAT

SYNTAX: >STAT lun[,C or ,D]

PARAMETERS: lun - logical unit number

C - Obtains channel registers (0-F) status

D - Obtains device status

OPERATION: This statement will obtain status from the channel or device specified whether an error has occurred or not and print it on the console. If the device is an HP 7976, the status from the last operation is printed.

EXAMPLE:

```
>5000 DEV 0,6,1,10,0
>5020 STAT D
>5030 RUN
```

The above example will print the device status of the disc. In this case status words 1 and 2 will be displayed on the console.

TIMEOUT

FORMAL NAME: Channel Program Timeout Flag

AID OPERATION NAME: TIMEOUT

DESCRIPTION: To disable the software timer, the user program may set TIMEOUT equal to -1. To increase the time allowed by N times, the user may set TIMEOUT to N. The timeout period is approximately 3 seconds.

INITIALIZED TO: Zero

EXAMPLE(S):

```
> 5010  .SET UP FOR SCOPE LOOP
> 5020  LET CHANNEL:=2
> 5030  TIMEOUT:=-1 .DISABLE I/O TIMEOUTS
> 5040  DB CC,3,!1400      .READ DISC ADDRESS
> 5050  BSIO AA
> 5060  WR 8,CC(0),2
> 5070  RR 8,CC(1),4
> 5080  GOTO 5060
> 5090  RSIO
> 5100 RUN
```

VER

==

FORMAL NAME: Verify

FUNCTION NAME: VER

SYNTAX: >VER lun,secount[,cylinder,head,sector]

PARAMETERS: lun - Logical unit number.

secount - Numbers of sectors to be verified.

cylinder - cylinder address
head - starting head address
sector - starting sector address

NOTE: Default for cylinder,head,sector is 0,0,0.

OPERATION: This function will verify the data on a number of sectors on a moving head disc.

EXAMPLE:

```
>5000 DEV 1,6,1,10,3
>5010 SFM 1,7
>5020 FOR I:=0 TO 410
>5030 SEEK 1,I,0,0
>5040 VER 1,192,I,0,0
>5050 NEXT 5020
>5060 RUN
```

This example verifies one cylinder at a time until the entire 7906 disc is checked.

VERI

FORMAL NAME: Verify Immediate

FUNCTION NAME: VERI

SYNTAX: >VERI lun,secount

PARAMETERS: lun - Logical unit number.

secount - Number of sectors to be verified.

OPERATION: This function will verify the data on a number of sectors on a moving head disc. The starting point will be the internal disc address.

EXAMPLE:

```
>5000 DEV 4,6,1,15,2
>5010 DB AA,128,%155555
>5020 RS 4
>5030 WDI 4,AA(0)
>5040 VERI 4,1
>5050 GOTO 5020
>5060 RUN
```

This example seeks to random locations, writes and verifies one sector.

WD (DISC)

FORMAL NAME: Write Data

FUNCTION NAME: WD

SYNTAX: >WD lun,buf(0)[,mask[,cylinder,head,sector]]

PARAMETERS: lun - Logical unit number.

buf - Buffer from which data is read, then written to disc. Buffer length determines word count of read. This parameter must be any buffer AA(0) through NN(0), where AA-NN define the buffer name and (0) sets an HP AID pointer to the first element in the buffer.

mask - Loads file mask on the 13037 controller only. The mask bits are:

Bits	Function
12	Incremental/decremental seek. If set and bit 15 is a 1, the cylinder address will be decremented when End-of-Cylinder; otherwise, incremented.
13	Allow sparing
14	Cylinder/surface mode. If set, a cylinder consists of all available surfaces; End-of-Cylinder is set when the last sector of the last surface has been transferred. In surface mode, End-of-Cylinder is set when the last sector of any surface has been transferred.
15	Allow incremental/decremental seek.

cylinder - cylinder address

head - head address

sector - sector address

NOTE: Default for cylinder,head,sector is 0,0,0.

OPERATION: This function will write the data specified by the buf parameter beginning at the location specified by the cylinder, head, sector parameters. An Address Record command will be issued to the disc controller to pass the cylinder, head and sector parameters. This is required to simulate the Sleuth format. This function updates disc internal address.

EXAMPLE:

```
>5000 DEV 0,6,1,10,0
>5010 RDB AA(0),6144
>5020 FOR I:=0 TO 410
>5040 WD 0,AA(0),7,I,0,0
>5050 NEXT 5020
>5060 RUN
```

This example fills one surface of a HP7906 disc with random data.

WD (LP)

FORMAL NAME: Write Data

CAUTION

Do not use the WD (LP) command with the HP 2680A page printer.

FUNCTION NAME: WD

SYNTAX: >WD lun,buf(0),mode,linelength

PARAMETERS: lun - Logical unit number

buf - Buffer containing write data. This parameter must be any buffer AA(0)-NN(0) where AA-NN define buffer name and (0) sets an HP AID pointer to the first element in the buffer.

linelength - Length of each line.

mode - Format character as indicated below:

	BIT							COMMAND
CODE	9	10	11	12	13	14	15	
0	0	0	0	0	0	0	0	SUPPRESS SPACE **
1	0	0	0	0	0	0	1	SINGLE SPACE
2	0	0	0	0	0	1	0	DOUBLE SPACE
63	0	1	1	1	1	1	1	63 SPACES
64	1	0	0	0	0	0	0	Chan 1(Top of form)*
65	1	0	0	0	0	0	1	Chan 2(bottom of form)*
66	1	0	0	0	0	1	0	Chan 3(Single space forms step-over)*
67	1	0	0	0	0	1	1	Chan 4(Double space forms step-over)
68	1	0	0	0	1	0	0	Triple space forms step-over)*
69	1	0	0	0	1	0	1	Next one-half page*
70	1	0	0	0	1	1	0	Next one-fourth page*
71	1	0	0	0	1	1	1	Next one-sixth page*

*Assigned according to HP programming standards.

** Not allowed for 2608A. Results are indeterminate.

OPERATION: This function will write data from the specified buffer, perform the indicated mode command over the length specified by the linelength parameter and transmit this data to the lun specified.

EXAMPLE:

```
>5000 DEV 0,7,2,10,0
>5010 DB AA,66,12345
>5015 FOR C:= 1 UNTIL 100
>5020 WD 0,AA(0),1,132
>5030 NEXT 5015
>5040 RUN
```

WD (TAPE)

==

FORMAL NAME: Write Data

FUNCTION NAME: WD

SYNTAX: >WD lun,buf(0)

PARAMETER: lun - Logical unit number

buf - Buffer that contains the write data. This parameter must be any buffer AA(0)-NN(0) where AA-NN define buffer name and (0) sets an HP AID pointer to the first element in the buffer.

OPERATION: This function will execute a write operation on the specified unit.

EXAMPLE:

```
>5000 DEV 0,5,1,10,0
>5010 RDB AA(0),8000 (statement takes approx. 45 sec)
>5015 FOR C:= 1 UNTIL 50
>5020 WD 0,AA(0)
>5030 NEXT 5015
>5040 REW 0
>5050 RUN
```

This example writes records of 8000 random words of data 50 times on mag tape unit 0.

WDI

FORMAL NAME: Write Data Immediate

FUNCTION NAME: WDI

SYNTAX: >WDI lun,buf(0)[,mask]

PARAMETERS: lun - Logical unit number.

buf - Buffer from which data is read, then written to disc. Buffer length determines word count of read. This parameter must be any buffer AA(0) Through NN(0), where AA-NN define buffer name and (0) sets an HP AID pointer to the first element in the buffer.

mask - Loads file mask on the 13037 controller only. The mask bits are:

Bits	Function
12	Incremental/decremental seek. If set and bit 15 is a 1, the cylinder address will be decremented when End-of-Cylinder; otherwise, incremented.
13	Allow sparing
14	Cylinder/surface mode. If set, a cylinder consists of all available surfaces; End-of-Cylinder is set when the last sector of the last surface has been transferred. In surface mode, End-of-Cylinder is set when the last sector of any surface has been transferred.
15	Allow incremental/decremental seek.

OPERATION: This function will write data on a moving head disc. The internal disc address will designate the starting point of the write operation. This function updates the internal disc address.

EXAMPLE:

```
>5000 DEV 0,6,1,10,0
>5010 DB AA,128,%155555
>5020 RS 0
>5030 WDI 0,AA(0),7
>5040 GOTO 5020
>5050 RUN
```

WFM

==

FORMAL NAME: Write File Mark

FUNCTION NAME: WFM

SYNTAX: >WFM lun

PARAMETER: lun - Logical unit number

OPERATION: This function will write a file mark on the specified unit.

EXAMPLE:

```
 >5000 DEV 0,5,1,10,0
 >5010 DB FF,6000,%22222
*>5020 WD 0,FF(0)
 >5030 WFM 0
 >5040 REW 0
 >5050 FSF 0
 >5060 REWOFF 0
 >5070 RUN
```

This example writes a file mark on mag tape, rewinds and then forward spaces to the file mark.

* This parameter must be any buffer AA(0) through NN(0) where AA through NN define buffer name and (0) sets an HP AID pointer to the first element in the buffer.

WFS

FORMAL NAME: Write Full Sector

FUNCTION NAME: WFS

SYNTAX: >WFS lun,buf(0)[,cylinder,head,sector]

PARAMETERS: lun - Logical unit number.

buf - Buffer from which data is read, then written to disc. Buffer length determines word count of read. This parameter must be any buffer AA(0) through NN(0), where AA-NN define buffer name and (0) sets an HP AID pointer to the first element in the buffer.

cylinder - starting cylinder address

head - starting head address

sector - starting sector address

NOTE: Default for cylinder,head,sector is 0,0,0.

OPERATION: This function will write a full sector on a moving head disc. Note the disc should be formatted after this operation. This function writes over the address field.

NOTE: This function is valid for 7910 and 79XX discs only. It updatesthe internal address of the disc.

EXAMPLE:

```
 >5000 DEV 1,6,1,10,2
 >5010 DB AA,138,%125252
*>5015 LET AA(0):=!80FE,AA(1):=400,AA(2):=!305
 >5020 DB BB,138,0
 >5040 WFS 1,AA(0),400,1,5
 >5050 RFS 1,BB(0),400,1,5
 >5060 SCB 1,AA(0),BB(0),5
 >5070 RUN
```

* AA(0) equals the sync word.

This example performs a single write full sector and a read full sector on a HP7906/20/25A disc. The buffers are then checked to verify the data.

WFSI

==

FORMAL NAME: Write Full Sector Immediate

FUNCTION NAME: WFSI

SYNTAX: >WFSI lun,buf(0)

PARAMETERS: lun - Logical unit number.

OPERATION: Buffer from which data is read, then written to disc. Buffer length determines word count of read. This parameter must be any buffer AA(0) through NN(0), where AA-NN define buffer name and (0) sets an HP AID pointer in the first element in the buffer.

OPERATION: This function will perform a full sector write operation on a 7910K and 7906/20/25A discs only. The internal disc address will be used and updated as the starting point.

EXAMPLE:
```
EXAMPLE: >5000 DEV 0,6,1,10,0
          >5010 RDB AA(0),138
          >5020 DB BB,138,0
          >5030 SEEK 0
          >5034 RFSI 0,AA(0)
          >5037 LET AA(5):= -AA(5)
          >5040 WFSI 0,AA(0)
          >5050 RDI 0,BB(0)
          >5060 RUN
```

This example uses the WFSI function to force a possible correctable data error negating the sixth word of buffer AA and using the original CRC value.

SLEUTH Simulator Diagnostic Language

RESERVED BUFFERS AND VARIABLES | APPENDIX A

The following lists those buffers and variables reserved for use by the Sleuth Simulator functions along with a brief explanation of assignment.

BUFFER	USAGE
00	RESERVED
PP	RESERVED
QQ	RESERVED
RR	Contains Magnetic tape and line printer commands.
SS	Contains status command and information as follows: Last status in 0 and 1 Status request command in 2 Expected status in 3 and 4 Don't care masks in 5 and 6 Printer status in 7 HP 7970E Tape status in 8 and 9; ESTA in 10 - 15
TT	Contains disc syndrome information
UU	Channel program buffer for general usage
VV	Channel program buffer primarily for obtaining disc address and other general usage.
WW	Used for passing commands and information - usage is as follows: 0 = command 1-3, 6-8, and 16 = command or information 4 = disc cylinder information 5 = Head and sector information

Sleuth Simulator Diagnostic Language

BUFFER USAGE

9 = counter

10 = DSJ information

11 = Sleuthsm variable usage indicator

12 = Not used

13-15 = disc parameters (cyl,hd,sect)

17 = SCB error count

18 = suppress status flag

19 = Pause on error flag

20 = DSJ information

21 = # of sect/cyl

22 = command

23 = cyl information

24 = head and sector information

25 = Head count for verify error

26 Sector count for verify error

27 = Next sector after verify error

28 and 29 = internal disc address (head & sector)

30-32 = beginning cylinder, head, and sector address for a read or verify

33 = final head address

34 = final sector address

35 = SOUT counter

40-49 = IS function line number

50-59 = IS function cylinder number

60-69 = DS function line number

70-79 = DS function cylinder number

BUFFER USAGE

80-89 = IT function line number

90-99 = IT function head number

100-109 = IT Function cylinder number

XX Channel program buffer that is variable in length and is built every time a channel program is executed.

YY CPVA buffer (See Appendix B in the 7906,7920,7925

Verifier Manual for more information).

ZZ This buffer contains DEV function parameters as follows:

USAGE

Initially variables O - Z act as pointers to set up the logical unit table. This table contains all necessary parameters for a particular device usage and is located in buffer ZZ. Once the variable information stored in buffer ZZ, all are then available for general usage by the Sleuth simulator functions.

NOTE: Attempted use of these variables by a user will adversely affect the users program.

V 0-7 = Logical unit number

8 = Ripple print counter

9 = Printer page length counter

W 10-17 = Channel number

18 = Identify code of executing device

19 = FMT cyl counter

X 20-27 = device number

28 = number of bytes per page (WD function)

29 = Data overrun counter for 12745A

BUFFER	USAGE
Y	30-37 = number of errors
	38 = Top of form indicator (WD function)
	39 = Number of characters to be printed on the next page for the WD function.
Z	40-47 = Unit number
O	50-57 = EXP status 2 word
P	60-67 = Mask words
Q	70-77 = 13037 disc type
R	80-87 = device indentification code
	88-90 = HP 7976 LDEV 0 STATUS
	91-93 = HP 7976 LDEV 1 STATUS
	94-96 = HP 7976 LDEV 2 STATUS
	97-99 = HP 7976 LDEV 3 STATUS
	100-102 = HP 7976 LDEV 4 STATUS
	103-105 = HP 7976 LDEV 5 STATUS
	106-108 = HP 7976 LDEV 6 STATUS
	109-111 = FP 7976 LDEV 7 STATUS

STRING BUFFER	USAGE
&WW	RESERVED
&XX	RESERVED
&YY	Print buffer
&ZZ	Contains information for user error reporting as follows:
	0 = variable name
	1-2 = buffer name

INDEX OF STATEMENTS/COMMANDS BY DEVICE TYPE

GENERAL	DISC	TAPE	LP
* ASSIGN	AR	BSF	RP
* BUMP	CL	BSR	WD
CHB	CORB	CLREAD	
DB	DISP	FSF	
DEV	DS	FSR	
ES	FMT	GAP	
ESTA	ID	RD	
* FOR-STEP-UNTIL	IDI	REW	
GET	IS	REWOFF	
* GOTO	IT	RRB	
* IF THEN	POLL	SELU	
* IFN THEN	RC	WD	
* INPUT	RD	WFM	
* LET	RDA		
* LOOPTO	RDI		
MC	RFS		
* NEXT	RFSI		
PAUSE	RQST		
PE	RS		
* PRINT	RSA		
* PROC	RSYN		
* RAND	RWO		
RDB	RWOI		
SOUT	RWV		
SST	RWVI		
STAT	SCB		
* TIMEOUT	SEEK		
	SFM		
	SKRD		
	SKWD		
	VER		
	VERI		
	WD		
	WDI		
	WFS		
	WFSI		

* HP AID STATEMENTS