Publication Number B29-374R04

OS/32 AIDS

USER'S GUIDE

PERKIN-ELMER

Interdata Division
2 Crescent Place
Oceanport, N.J. 07757

May 1978

PAGE REVISION STATUS SHEET

PUBLICATION NUMBER 29-374

TITLE OS/32 AIDS User's Guide

REVISION R04 DATE May 1978

PAGE	REV.	DATE	PAGE	REV.	DATE	PAGE	REV.	DATE
i/ii	R04	5/78	Index1					
iii	R04	5/78	thru					
iv	R04	5/78	Index5	R04	5/78			
1	R04	5/78						
2	R04	5/78						
3	R02	7/76						
4								
thru								
6	R04	5/78						
7	R02	7/76						
8	R02	7/76						
9	R04	5/78						
10	R04	5/78						
11								
thru								
15	R02	7/76						
16	R04	5/78						
17	R02	7/76						
18	R02	7/76						
19	R04	5/78						
20	R02	7/76						
21								
thru								
23	R04	5/78						
24	R02	7/76						
25	R04	5/78						
26	R04	5/78						
27								
thru								
31	R02	7/76						
32	R04	5/78						
33	R04	5/78						
34	R03	8/77						
35	R04	5/78						
A1-1/								
A1-2	R04	5/78						
A2-1	R04	5/78						
A2-2	R04	5/78						
A3-1	R04	5/78						
A3-2	R04	5/78						

PREFACE

This manual is a user's guide to OS/32 AIDS, a user-oriented assembly level debugging program intended for Interdata 32-bit processors. Section 1 introduces the AIDS environment and the program's features. Sections 2, 3, and 4 detail AIDS organization, data format options, and commands. Operating instructions are discussed in Section 5. The three appendices summarize OS/32 AIDS commands, define OS/32 terminology, and present sample dump formats.

The reader is referred to the following manuals for more information:

> Common Assembler Language (CAL) User's Manual,
> Publication Number 29-375.
> OS/32 MT Operator's Reference Manual,
> Publication Number 29-574.
> MTM Terminal User's Reference Manual,
> Publication Number 29-591.

TABLE OF CONTENTS

TABLE OF CONTENTS (Continued)

APPENDICES

TABLES

1. INTRODUCTION

OS/32 AIDS, Program Number 03-374, is a user-oriented assembly level debugging program intended for use on Interdata 32-Bit Processors. AIDS is intended to help the user debug programs (U-tasks), and presumes that the user program is running in an OS environment. A list of AIDS terminology can be found in Appendix 2 for quick reference.

1.1 Environment

AIDS requires 19.25kb of memory above the operating system and is operable under OS/32 MT. AIDS is reentrant and may have any number of simultaneous users when established as a reentrant library segment under OS/32 MT. Distinct additional storage, 2800 bytes, is required for each user and is allocated within the task's segment by AIDS when entered at AIDS starting location.

1.2 Features

AIDS checks the operating system and sets the proper segmentation registers. The task status word (TSW) is also set so that AIDS processes SVC14 calls, arithmetic faults, memory access faults, and illegal instruction faults. AIDS also tests the system for single and double precision floating point capabilities and sets flags to allow or disallow the floating point options available in AIDS.

AIDS can be used as either an interactive or batch debugging program. Interactively it reads a text line, called a directive, that is free format from the logical command unit (LU5) and executes that directive. Error diagnostics allow the user to correct directive errors easily. The abilities to display and modify the contents of memory locations, alter program flow, breakpoint, protect data locations, and execute single steps are useful for interactive debugging.

When operating in batch mode, directive errors can result in either a halt, which awaits operator intervention, or in an abort action, which cancels the user task. The trace, snapshot dumps, and protect capabilities make AIDS a useful batch-oriented debugging tool.

2. OS/32 AIDS ORGANIZATION

2.1 Basic Operations

1. Display and modify memory locations and the floating point and general registers.

2. Print sections of memory to a list device. Since AIDS uses Logical Units, output can be directed to any physical device.

3. Provide program utilities such as snapshot printouts, cell/register protections, trace execution and breakpointing.

4. Provide single step execution and the displaying of the current bias, location counter, task status or condition code.

5. Convert requested data or the current open cell from one data format to another.

2.2 Directives

A directive is the text form of an AIDS command. Directives have a definite syntax beginning with a unique character op-code followed by a suitable option character. If operand arguments are requested, they must be preceded by a space, separated by commas and ended with a carriage return. The format of a directive is:

```
DO  a1,a2,a3
```

where

D is the unique op-code character,
O is the requested option,
a1,a2,a3 are optional operands.

The following terms are used to explain the individual directives.

Address - The location of a cell or register.

Option - Specifies the data format of the directive (decimal, character, fullword hex, etc.).

Data - Data to be used when the directive is executed.

Count - A decimal field specifying a number of iterations.

Delay - A decimal field specifying the number of times a breakpoint or snapshot is encountered before being executed.

2.3 Address Conventions

An address field may have any of the following forms:

<u>Absolute</u> - the address field is interpreted as the hexadecimal fixed address of a memory cell within the user program space.

<u>Relative</u> - the address field is added to the AIDS 'BIAS' value to obtain the desired absolute address of a memory cell.

<u>Local Relative</u> - the address field is added to the AIDS current cell to obtain the address of the new desired cell address.

<u>Registers</u> - the address field is interpreted as the hexadecimal number of a general or floating point register.

<u>General Purpose Register</u> (GPR) - any hexadecimal digit from 0 to F preceded by a period (.).

<u>Single Precision Floating-Point Register</u> (SPFPR) - any even hexadecimal digit from 0 to E preceded by a colon (:).

<u>Double Precision Floating-Point Register</u> (DPFPR) - any even hexadecimal digit from 0 to E preceded by a semi-colon (;).

When specifying memory cells, address fields may be hexadecimal numbers up to six characters. An address field containing only a hexadecimal address is an absolute address. An address field followed by an asterisk (*) adds the current AIDS 'BIAS' to the given hexadecimal address to give the desired absolute address. An address field preceded by an asterisk (*) and a plus sign (+) adds the address of the current open cell to the given hexadecimal address to give the new desired open cell address.

Example:

Address Field	Meaning
1A0	Open memory cell at absolute address X'1A0'.
1A0*	Open memory cell at absolute address X'1A0' plus the value of the current AIDS 'BIAS'.
*+10	increment the address of the current open cell by X'10' and open that address.

If an address field is a register, the register number is preceded by the punctuation character period (.), colon (:), or semi-colon (;). These characters indicate the general purpose, and single and double precision floating-point registers, respectively.

Examples:

Address Field	Meaning
.A	General Purpose Register A
.2	General Purpose Register 2
:8	Single Precision Floating-Point Register
:E	Single Precision Floating-Point Register
;C	Double Precision Floating-Point Register
;4	Double Precision Floating-Point Register

The following examples are flagged as errors because the register addresses do not exist.

.10	Flagged as NO-SUCH
:3	Flagged as NO-SUCH
;9	Flagged as NO-SUCH

Table 1 lists the address mode, mode indicator (Key), and an example of each usage. A hexadecimal address value is represented by h.

TABLE 1. ADDRESS FIELDS

ADDRESS MODE	KEY	EXAMPLE	
ABSOLUTE	h	32F0	32F0
RELATIVE	h*	1B0*	1B0 + 'BIAS' VALUE
LOCAL	*+h	*+10	ADDRESS 'CURCELL' + 10
GPR	.h	.5	GENERAL PURPOSE REGISTER 5
SPFPR	:h	:E	SINGLE PRECISION FLOATING-POINT REGISTER
DPFPR	;h	;A	DOUBLE PRECISION FLOATING-POINT REGISTER

2.4 Logical Unit Assignments

AIDS uses three Logical Units for I/O. These units are initialized to the following within the program:

Command Input Unit	LU5	All directives are read from this unit.
List Unit	LU3	Dump printed output and other printe outputs are listed to this unit.

Binary Output Unit	LU2	Loader format and binary dump output are written to this unit.

The Logical Unit assignments for AIDS may be modified from the initial assignments to units 2, 3, and 5 by utilization of the START command or after START using the LU command.

The command input device must be assigned prior to program execution. The output logical units must be assigned prior to their use.

2.5 Modes of Use

AIDS may be used in either interactive or batch mode.

In interactive mode, a '>' prompt is displayed whenever a command is to be entered on the operator's console.

In batch mode, directive errors can result in either a halt, requiring operator intervention, or in abort action, which cancels the user's task. Messages are output to the message logging device assigned to the List unit.

2.6 Starting OS AIDS

The OS START command is used to Starc OSAIDS and to provide arguments specifying alternate LU assignments in place of the expected default.

```
DEFAULT ASSIGNMENTS ARE:    2 = Binary Device
                            3 = List Device
                            5 = Command Input Device
```

The following directive shows the key words which are recognized in the START command. All are optional and may be specified in any order, separated by commas.

FORMAT:

```
START , [ BINARY=[LU/] [V:FD] ] , [ LIST=[LU/] [ V:FD ] ],
        [ COMMAND=[LU/] [V:FD] ]
```

WHERE:

```
   [       ]  ARE OPTIONAL
```

BINARY = (Optional) specifies that an optional logical unit and/or file or device is desired for any requested binary output.

LIST = (Optional) specifies that an optional logical unit and/or file or device is desired for command response.

COMMAND = (Optional) specifies that an optional logical unit and/or file or device is desired for command response.

EXAMPLES: Assign LU 6 for Binary Output to Null device.

```
ST , BIN=6/NULL:
```

Assgin default list LU to a file on Volume M300 called result.

```
ST , LI = M300:result
```

Assign default binary output to the null device, the list device to LU 9, Volume M300 file RESULT, and the command device to the default volume, file called INPUT.PRB to LU 12

```
ST ,CO=12/INPUT.PRB,LI=9/M300:RESULT,BI=NULL:
```

3. DATA FORMAT OPTIONS

The following are the data format modes that a directive can be requested to interpret. This data format mode is identified by a single option character.

3.1 Halfword Hexadecimal Format (X)

On input AIDS accepts up to a maximum of four characters assumed to be right justified. If more than four characters are entered, AIDS accepts the four rightmost characters. Leading zeros can be omitted. Any character not a valid hexadecimal digit results in an error message. In all output, four hexadecimal digits are output with leading zeros included.

3.2 Fullword Hexadecimal Format (Y)

This mode is similar to the halfword format (X), except that up to eight hexadecimal digits are accepted on input and eight hexadecimal digits are printed on output.

3.3 Character Mode (C)

On input AIDS accepts a string of ASCII characters until a carriage return or a percent sign (%) is encountered. A maximum of sixty characters is permitted. On output each halfword is written on the output device as two ASCII characters with non-printing characters converted to periods (.) before output.

3.4 Halfword Decimal Format (H)

On input AIDS interprets the operand as a signed decimal integer. The plus (+) sign is assumed, but may be included. The minus sign (-) is required for negative numbers. An error message results if the magnitude is greater than 32767. Leading zeros may be omitted. On output the plus sign and leading zeros are omitted.

3.5 Fullword Decimal Format (F)

This format is the same as the halfword format (H) except that the magnitude may be as large as 2,147,483,647.

3.6 Single Precision Floating-Point Format (E)

On input AIDS accepts generalized floating-point real number representation. Any single precision floating-point number with a magnitude between .1 and 999999 is output without using exponential form. A minus sign is printed if the number is negative. Trailing zeros and decimal points are deleted.

All other numbers are printed in the format:

(-).nnnnn(E±e)

where

n is a decimal digit.
E indicates single precision exponentiation.
e is a one or two digit decimal number representing the exponent.
( ) parenthesis indicate optional parts of the number.

Examples:

Number	Printed Output
.00000002	.2E-7
-.002	-.2E-3
200	200
-20,000,000,000	-.2E 11
00.0	0
-2.00	-2
12345678	.12346E8

NOTE

Data will be normalized before output. Thus, non-floating point data output with E or D format, may produce an arithmetic fault.

3.7 Double Precision Floating-Point Format (D)

On input AIDS accepts generalized double precision floating-point real number representation. Any double precision floating-point number with a magnitude between .1 and 99999999999999 is output without using exponential form. A minus sign is printed if the number is negative. Trailing zeros and decimal points are deleted.

All other numbers are printed in the format:

(-).nnnnnnnnnnnnnn(D±e)

where

n is a decimal digit.
D indicates double precision exponentiation.
e is a one or two decimal digit representing the exponent.
( ) parenthesis indicate optional parts of the number.

Examples:

Number	Printed Output
123456789123456	.12345678912346D15
-.0000000002	-.2D-9
12345678912345	12345678912345
00.0	0

3.8 Double Precision Floating Point Format (Y)

On input AIDS accepts hexadecimal characters (0-9;A-F) and right justifies them in the register specified.

Examples:

NUMBER	PRINTED OUTPUT
A	0000 0000 0000 000A
ABCD	0000 0000 0000 ABCD
9ABCDEF	0000 0000 09AB CDEF
6789ABCDEF	0000 0067 89AB CDEF
3456789ABCDEF	0003 4567 89AB CDEF
F123456789ABCDEF	F123 4567 89AB CDEF

3.9 Disassembly Format (A)

The disassembly format option is applicable to printed output only and disassembles machine language into assembler language. Extended branch mnemonics are used where they are implied. There may be some variations concerning the extended mnemonics entered by the programmer and those disassembled by AIDS because some extended branches have identical op-codes for different mnemonics, i.e., BES, BZS. The format of disassembly output is as follows.

(Address) (Contents) (Mnemonic) (Operands)

where

Address is a six digit hexadecimal address either absolute or relocatable.

Contents is one, two or three halfwords of machine code displayed in hexadecimal digits.

Mnemonic is the appropriate mnemonic for the machine instruction.

Operands are the addresses or literal operands for the displayed instruction with appropriate index registers. The display is always hexadecimal information.

Example:

```
00175CR  48DF 0002     LH        D,2(F)

Relative Location   Mnemonic    Operands
Address  Contents
```

Table 2 lists the data formats available within AIDS.

TABLE 2. AIDS DATA FORMATS

OPTION CHARACTER	FORMAT	EXAMPLE
X	HALFWORD HEXADECIMAL	3F0A
Y	FULLWORD HEXADECIMAL	41053AAB
H	HALFWORD DECIMAL (INTEGER)	1354
F	FULLWORD DECIMAL (INTEGER)	1748325
C	HALFWORD CHARACTER (ASCII)	AM
E	SINGLE PRECISION FLOATING-POINT	.45328E-6
D	DOUBLE PRECISION FLOATING-POINT	.45327983D-6
A	ASSEMBLY LANGUAGE	LB 5,1FA(E)

4. OS/32 AIDS COMMANDS

The functional description of all AIDS directives are identified and their syntax given with their legal options and operands in the following sections. Usage notes and cautions are given in each section where required. All directives are input followed by a carriage return when in the interactive mode.

The Directives are grouped into three sections:

- Cell/Register Display and Modification
- Utility Directives
- Environment Directives

4.1 Cell/Register Display and Modification

4.1.1 OPEN

Function: Allow the user to display the contents of the cell or register (address) in any of the data formats described previously. When a cell is opened, the mode and address become the 'CURCELL' and 'CURMODE'. The count field, when specified, is the number of logical data cells or registers to be opened and displayed beyond the indicated address. The count includes the present cell being opened. A default of one is assumed. If the option character is omitted, the cell is opened in the previous mode used.

Format: O[(option)] address [,count]

Legal Options: A, C, D, E, F, H, X, Y

Examples: Open cell 1004 and display in disassembly format for three consecutive cells.

```
OA 1004,3

001004:  0822        LR      2,2
001006:  43308062    BZ      106C
00100A:  C8300045    LHI     3,45
```

Open cell 1006 and display in character format.

```
OC 1006

001006:  C0
```

Open Double Precision Floating Point Register A and display as hexadecimal.

```
OY ;A

DF (A)   43AC1000 76390000
```

Open DPFPR 8 and display as floating point.

```
OD ;8
```

```
DF(8)    -.56326638318394D-46
```

Open Single Precision Floating Point Register C and display as hexadecimal.

```
OY :C
```

```
FP (C)  41100000
```

Open SPFPR C and display as floating point.

```
OE :C
```

```
FP (C)  10
```

Open General Purpose Registers 5 through 7 and display as fullword decimal.

```
OF .5,3
```

```
GP(5)    13
GP(6)    901437
GP(7)    -213
```

Open cell 21F0 in halfword hexadecimal.

```
OX 21F0
```

```
0021F0: 31F0
```

Open cell 21F0 + 10 in halfword hexadecimal with the current open cell at address 0021F0.

```
O *+10
```

```
002200: FE91
```

Open cell 21F0 in fullword hexadecimal.

```
OY 21F0
```

```
0021F0: 31F0AC82
```

Open relative location 1000 in character mode with the 'BIAS' value set to 2000.

```
OC 1000*
```

```
003000: EB
```

NOTE:

When an asterisk (*) modifier is specified following the command option character, the result is the display of the location counter in relative address form.

```
OC* 1000*

001000R  EB
```

4.1.2 REPLACE

Function: Allow the user to replace the contents of the current open cell or register with new data specified in the operand field of the directive. If the mode option is not specified, CURMODE is used to interpret the operand field.

A contiguous block of memory may be initialized to a specific value through the use of the count option. The count value defaults to one.

The execution of the Replace command automatically increments CURCELL to the next logical cell (i.e., the amount of memory operated on by the Replace command). This enables the user to conveniently replace memory without opening the next cell each time.

Format: R[(option)] data[,count]

Legal Options: D, E, F, H, X, Y, C

Set all DPFP registers to 0.

```
OD ;0        sets CURCELL to DPFPR (0)

DF (0)       -40.156254907208

R 0,8        Replace eight consecutive registers
```

Set SPFP register A to 385.671

```
OE :A        sets CURCELL to SPFPR (A)

FP (A)       10

R  385.671
```

Set all bits in memory from location X'1000' to location X'2000'.

```
OF 1000       sets CURCELL
 001000:      13501
R -1,2048     replaces 2048 cells with the fullword
              decimal value -1.
```

Set the value of the current open cell to halfword decimal 10353.

```
RH 10353
```

Store the characters ABCD starting with the current open cell.

```
RC ABCD
```

NOTE

When replacing in character mode, caution must be exercised not to overwrite memory beyond the desired location. AIDS accepts a character string up to sixty characters or until a carriage return is entered. It is therefore possible to enter a string which may exceed the anticipated space.

4.1.3 NEXT

Function: Causes the next logical cell to be opened, but not displayed. This cell becomes the new CURCELL.

Format: N

4.1.4 OPEN NEXT

Function: Displays the next logical cell or register. If an option is not specified, the mode last used by AIDS is used in the display of the cell.

Format: Carriage return

or: (Space)[(option)]

Legal Options: A, C, D, E, F, H, X, Y

Examples:

```
OH 1000
001000:       4120
(Space)
001002:       8830
```

```
OF .5
GP (5)       10
Carriage Return
GP (6)       20141
Carriage Return
GP (7)       0
```

4.1.5 OPEN PREVIOUS

Function: Displays the cell or register immediately preceding the current open cell or register. If no option is specified, AIDS uses the mode of the last operation.

Format: -[(option)]

Legal Options: C, D, E, F, H, X, Y

Example:

```
OH  1000
001000:      31F0
-
000FFE:      389A


OF .7
GP (7)       0
-
GP (6)       20141
-
GP (5)       10
```

4.1.6 JUMP

Function: Displays and opens the cell pointed to by the contents of the current open cell. If no options are specified, CURMODE is used as default. The new open cell becomes the new CURCELL.

Format: J[(option)]

Legal Options: A, C, D, E, F, H, X, Y

Example: Displays the contents of the cell pointed to by the contents of the present open cell (CURCELL).

```
OX  500
000500:      3140
JX
003140:      1AE
```

NOTE

The contents of the cell pointed to by CURCELL is interpreted as a full-word address.

4.1.7 ADD

Function: Allows the user to increment the contents of the current open cell or register with a given value.

A convenient feature is provided by the second operand option. When the user specifies $data_2$, AIDS adds the two data values in the specified mode and displays the results. AIDS does not update memory or change CURMODE as a result of this calculation. Halfword modes are not valid with $data_2$ entry.

Format: A(option) $data_1$[,$data_2$]

Legal Options: D, E, F, H, X, Y

Examples: Increases the contents of SPFP register 6 by 5.31.

```
OE :6
FP (6)       10
AE  5.31
OE :6
FP (6)       15.31
```

Add two hexadecimal numbers

```
AY  4803051E, 100F
    4803152D
```

4.1.8 SUBTRACT

Function: This operation is similar to the ADD directive except that subtraction rather than addition occurs.

Format: S(option) $data_1$[,$data_2$]

Legal Options: D, E, F, H, X, Y

Decreases the contents of GP register 5 by 10

```
OF .5
GP (5)       10
Carriage Return
GP (6)       20141
Carriage Return
GP (7)       0
```

4.1.5 OPEN PREVIOUS

Function: Displays the cell or register immediately preceding the current open cell or register. If no option is specified, AIDS uses the mode of the last operation.

Format: -[(option)]

Legal Options: C, D, E, F, H, X, Y

Example:

```
OH  1000
001000:      31F0
-
000FFE:      389A


OF .7
GP (7)       0
-
GP (6)       20141
-
GP (5)       10
```

4.1.6 JUMP

Function: Displays and opens the cell pointed to by the contents of the current open cell. If no options are specified, CURMODE is used as default. The new open cell becomes the new CURCELL.

Format: J[(option)]

Legal Options: A, C, D, E, F, H, X, Y

Example: Displays the contents of the cell pointed to by the contents of the present open cell (CURCELL).

```
OX  500
000500:      3140
JX
003140:      1AE
```

NOTE

The contents of the cell pointed to by CURCELL is interpreted as a fullword address.

4.1.7 ADD

Function: Allows the user to increment the contents of the current open cell or register with a given value.

A convenient feature is provided by the second operand option. When the user specifies data$_2$, AIDS adds the two data values in the specified mode and displays the results. AIDS does not update memory or change CURMODE as a result of this calculation. Halfword modes are not valid with data$_2$ entry.

Format: A(option) data$_1$[,data$_2$]

Legal Options: D, E, F, H, X, Y

Examples: Increases the contents of SPFP register 6 by 5.31.

```
OE :6
FP (6)       10
AE  5.31
OE :6
FP (6)       15.31
```

Add two hexadecimal numbers

```
AY  4803051E, 100F
    4803152D
```

4.1.8 SUBTRACT

Function: This operation is similar to the ADD directive except that subtraction rather than addition occurs.

Format: S(option) data$_1$[,data$_2$]

Legal Options: D, E, F, H, X, Y

Decreases the contents of GP register 5 by 10.

```
OF .5
GP (5)       23
SF 10
OF           .5
GP (5)       13
```

Subtract two hexadecimal numbers.
Subtract 29FB from 135EDA

```
SY 29FB, 135EDA
   001334DF
```

NOTE

The address of the current cell is not incremented to the next logical cell or register in either the ADD or SUBTRACT Directive.

4.1.9 CONVERT

Function: To allow the user to convert data from one format (option 1) to another (option 2), the only allowable conversion operations are the following:

Character to hexadecimal	C→Y
Hexadecimal to character	Y→C
Decimal to hexadecimal	F→Y
Hexadecimal to decimal	Y→F
Single Precision FP to hexadecimal	E→Y
Hexadecimal to Single Precision FP	Y→E
Double Precision FP to hexadecimal	D→Y
Hexadecimal to double precision FP	Y→D

The execution of this command does not alter any memory or mode options within AIDS. Halfword modes are not applicable.

Format: C(option 1)(option 2) data

Legal Options: C, D, E, F, Y

Examples: Hexadecimal to character

```
CYC 44
    D
```

Single precision floating point to hexadecimal

```
CEY 10.0
    41A00000
```

Hexadecimal to single precision floating point

```
CYE 41A00000
    10
```

Decimal to hexadecimal

```
CFY 2858
    00000B2A
```

Hexadecimal to decimal

```
CYF B2A
    2858
```

Double precision floating point to hexadecimal

```
CDY -.67489226182806D47
    E7BD2525 24C52E6C
```

Hexadecimal to double precision floating point

```
CYD E7BD2525 24C52E6C
    -.67489226182806D47
```

4.2 Program Utility Directives

4.2.1 DUMP

Function: To output the contents of a section of memory defined by two address operands. The first address is the start address and the second address is the ending address of the requested section of memory to be printed. The format of the dump printout is specified by the option character in the directive. The address operands are required on all Dump commands except the Utility Dump command (DU).

The Utility Dump command displays the address locations of all program sentinals that have been set. If a bias is set in AIDS, and the sentinal address location is greater than the bias, then the address locations are printed relative to the bias and indicated by printing the character 'R' after the address location.

The output of all the dump options is written to Logical Unit 3 except the Loader format (L) and the memory image binary (B) which is written to Logical Unit 2.

The Loader format dump is an absolute program capable of executing under an operating system. The Binary format dump can be loaded by the '50 Sequence' or an SVC 1.

Examples of the dump formats written to Logical Unit 3 can be found in Appendix 3.

Format: D(option) address$_1$, address$_2$

NOTE

All dumps return to the command mode. The existence of active sentinels within user programs causes erroneous dumps. Sentinels should be removed by the user through the ZAP directive before dumping.

TABLE 3. DUMP COMMAND FORMATS

Option Character	FORMAT TYPE
A	Disassembly
B	Memory Image Binary
C	Character
D	Double Precision Floating Point
E	Single Precision Floating Point
F	Fullword Decimal
H	Halfword Decimal
L	Loader
U	Listing of Sentinel Addresses
X	Halfword Hexadecimal
Y	Fullword Hexadecimal

4.2.2 INSERT

Function: To allow the user to request one of the following AIDS operations: Breakpoint (X), Snapshot Dump (S), Protect (P), and Trace (T). The Insert command enters the user's request into the proper sentinel table.

The Insert operations have different formats for each request. If the operation character is followed by an asterisk (*), then the resulting sentinel is printed with the address location relative to the bias set in AIDS and identified with the character 'R' following the address location.

NOTE

To protect double precision cells, other than registers, it is necessary to protect both words individually.

4.2.2.1 BREAKPOINT (X)

Function: To allow the user to halt the execution of a program at a specified address location during the logical execution of a program. At this time, the user may examine and/or modify memory or register locations.

A maximum of thirty-two breakpoints can be maintained. The user can specify the number of times that the breakpoint is encountered before the execution of the program is halted. This delay request must not exceed 32,767. If the delay field is omitted, then a delay of zero is assumed and execution halts each time the breakpoint sentinel is encountered. When a delayed breakpoint is activated, the delay count is reset and the breakpoint will not reactivate until the delay count has been reached again.

Inserting a breakpoint involves inserting the hexadecimal special instruction X'E1E0' (SVC 14,0) at the requested address in the user's code. The instruction replaced by this special AIDS instruction is saved in the breakpoint table and is restored when the breakpoint is cleared. Executing a breakpoint does not clear the breakpoint.

When a breakpoint is encountered, the following message results:

```
BP (address)
```

The address field is absolute unless the relative display character (*) was requested at the time the breakpoint was inserted. The character 'R' is printed after the address location if the address is relative.

Format: IX[*] address[,delay]

Examples: Set a breakpoint at X'3519E' with no delay

```
IX 3519E
```

Set a delayed breakpoint at X'51E' with a delay of 10.

```
IX 51E,10
```

Set a delayed breakpoint at the current location +20 and display the address absolute

```
IX *+10,150
```

Set a breakpoint at relative address 50 and display relative.

```
IX* 50*
```

Set a breakpoint at relative address 7E and display relative with a delay of 20.

```
IX* 7E*,20
```

NOTES

An error message results if a breakpoint is inserted on an existing breakpoint. Breakpoints must be inserted on the first halfword of an instruction.

If the instruction on which the breakpoint is set is not a branch instruction, then the instruction has been executed by the AIDS interpreter.

4.2.2.2 PROTECT (P)

Function: To allow the capability of protecting single and/or double precision floating point registers, general purpose registers or memory locations. Memory locations can be fullword or halfword protected. If the character 'H' is specified, then protection is for a halfword location or else fullword protection is assumed. A maximum of thirty-two protect sentinels is allowed.

The protection of a register or memory location does not modify the user's program. The value of the register or memory location is saved by AIDS at the time the protect option is specified and causes the AIDS interpreter to execute each user instruction. Each protected item value is checked with the stored value in the protect table. If the value of the protected item has changed, then the new value is saved and the following message is output.

```
(address1) CH (address2) FR (data1) to (data2)
```

Address1 is the address location of the instruction that caused the contents of address2 (the protected item) to change from data1 to data2. All data is displayed in hexadecimal, but the address can be relative or absolute.

Format:

```
IP[H][*] address
```

Examples:

Protect single precision floating point register C

```
      IP :C
      00365E: CH FP(C)FR 413896E1 to 42000000
```

Protect location 100 relocatable and display relative.

```
      IP* 100*
      001F32R CH 000100R FR 0000385E to 1000385E
```

Protect double precision floating point register 8

```
      IP ;8
0006A0: CH DF(8)
FR E7BD2525 24C52E6C to 42000000 00000000
```

NOTE

To protect double precision cells other than registers, it is necessary to protect both words individually.

4.2.2.3 TRACE (T)

Function: To enable the user to follow the execution of a program through the continuous listing of the location counter, the hexadecimal representation of the instruction, the mnemonic and operands of the instruction to the list device. A Branch Trace (B) lists the value of the location counter only if a branch instruction is encountered and taken.

The present value of the location counter and the disassembly of the instruction is listed under a normal trace, while only the location of the branch instruction and its destination address are listed under the Branch Trace option.

An error message results if overlapping trace areas are specified. There are no sentinels inserted in the user's program from the trace feature. A maximum of thirty-two trace regions may be defined.

Format: IT[B][*] address1, address2

Examples: Set a trace region between X'1A0' and X'1E0'

```
IT 1A0,1E0
GO 1A0

0001A0:  7820 FF74          LD      2,118
0001A4:  3742               LDR     4,2
0001A6:  7E60 FF8E          LMD     6,138
0001AA:  7E60 FFBA          STMD    6,168
0001Ae:  7040 FFDE          STD     4,190
0001B2:  3942               CDR     4,2
0001B4:  4330 4000 01C8     BE      1C8
0001C8:  7960 FF9C          CD      6,168
0001CC:  4330 4000 01E0     BE      1E0
0001E0:  2418               LIS     1,8
```

Set a trace region between X'1B2' and X'200' and trace relative locations. Assume a bias set to X'100'.

```
IT* 1B2,200
GO  1A0
```

```
0000B2R  3942                CDR      4,2
0000B4R  4330 4000 01C8      BE       1C8
0000C8R  7950 FF9C           CD       6,168
0000CCR  4330 4000 01E0      BE       1E0
0000E0R  2418                LIS      1,8
0000E2R  7980 FF5A           CD       8,140
0000E6R  4330 4000 01FA      BE       1FA
0000FAR  79A1 4100 0168      CD       A,168(1,1)
000100R  4330 4000 0214      BE       214
```

4.2.2.4 SNAPSHOT DUMP (S)

Function: To allow the display of register or memory locations at specified memory locations during the logical execution of a program.

Delays are available on each snapshot sentinel and more than one type of dump is available simultaneously on a single snapshot sentinel. Snapshot dumps are identical with normal dump outputs. The user should refer to the DUMP directive explanation for format samples.

The setting of a snapshot sentinel involves inserting an SVC 14,0 at the current open cell. For this reason, AIDS does not permit the user to insert a breakpoint sentinel over a snapshot sentinel or vice versa. A maximum of thirty-two snapshot sentinels can be inserted by the user. They may each be on a different sentinel, on the same snapshot sentinel, or divided among snapshot sentinels in any fashion.

Format: IS(option)[*] address1,address2 ,delay

Legal Options: A, B, C, D, E, F, H, L, X, Y

NOTE

The snapshot sentinel is inserted at the address of the current 'open cell'. To insert the snapshot at any other address, that address must be the current open cell address.

Examples: Dump memory from X'100' to X'200' in single precision floating point with no delay, when location 400 is reached.

```
OH 400        set open cell
ISE 100,200
```

Dump memory from CURCELL to CURCELL+100 after delaying 50 times. Dump in relocatable assembly language.

```
      ISA*  *,*+500,50
```

Dump memory in binary format from X'40000' to X'42000' each time the current open cell is encountered. Output is written to Logical Unit 2 for Loader and Binary format dumps.

```
      ISB  40000,42000
```

Dump all the general purchase registers in fullword hexadecimal every 25 times the instruction at location 1000 is encountered.

```
      OH  1000
      ISY .0,.F,25
```

Dump all the general purpose registers in fullword hexadecimal every 10 times the instruction is encountered. Also dump single precision floating point registers 2 through 8 every 20 times the same instruction is encountered.

```
      OH  1000
      ISY .0,.F,10
      ISE :2,:8,20
```

4.2.3 ZAP

Function: To provide the capability to remove utility sentinels. These sentinels can be removed singularly, by class or all set sentinels at once.

The option character of the directive specifies the class of sentinel (SNAPSHOT, TRACE, PROTECT or BREAKPOINT) to be removed. The address field specifies which particular sentinel is to be removed. If the address field is omitted, then all sentinels for the specific class are deleted.

If the option character 'Z' is entered, no address field is allowed, and all sentinels of all four classes are deleted.

Format: Z(option) [address]

Legal Options: P, S, T, X, Z

Examples: Unprotect the double precision floating point register E

```
ZP ;E
```

Unprotect memory location D00

```
ZP D00
```

Remove the trace region from X'3000' to X'3500'

```
ZT 3000
```

Remove all breakpoint sentinels

```
ZX
```

Remove all sentinels of all classes.

```
ZZ
```

4.2.4 GO

Function: To start or return processing to a user program with the machine state and condition code restored to its initial or interrupted value.

The GO command starts execution at the location specified by the address field. If the address field is omitted, execution continues with the next logical instruction following the last sentinel encountered. Initially, the default value of the address field is X'100'.

If an address field ONLY is specified,AIDS will place a carriage return at the current UTOP.

If User Start Options are specified, all characters from the one immediately following the comma thru the carriage return (80 characters maximum) will be moved to current UTOP.

If the GO directive is a continuation following a breakpoint, the instruction at the breakpoint location is executed interpretively and normal execution proceeds. The breakpoint is not removed.

Format: GO [address] [,user start options]

NOTE:

If user start options are specified (see 2.6), a maximum of 80 characters is allowed.

Examples: Execute program at address X'3000'

```
GO 3000
```

Execute program at current open cell plus 10

```
GO *+10
```

Continue executing from last breakpoint

```
GO
```

Execute program at address X'100' after placing options at CTOP for user.

```
GO 100, BINARY=6,LIST=9,COMMAND=12
```

4.2.5 WHERE

Function: To print the current value of the 'Location Counter' in AIDS. If a bias has been set and the address of the Location Counter is greater than the bias, the printed value is relative to the set bias.

Format: W

Examples: Print the current address of the 'Location Counter' and the AIDS bias is zero.

```
W
000476:
```

Print the current address of the 'Location Counter' with the AIDS bias set.

```
BI 100
W
 000376R
```

4.2.6 EXECUTE

Function: To execute one or a given number of instructions of a user program. The starting address can be given or defaulted to the address of the last executed instruction. If the character 'T' is entered following the directive character, then a printout of each instruction and its address is printed when the instruction is executed.

All inserted sentinels are not affected by the Execute directive. A true breakpoint causes a return to the AIDS command level regardless of the number of instructions requested to be executed. The instruction address location is printed relative to the bias if the address is greater than the bias.

Format: X[T] [count][,address]

Examples: Execute and print four instructions starting at address location X'1A0' with no bias set.

```
XT 4,1A0
   AIDS OUTPUT

0001A0: 7820 FF74     LD      2,118
0001A4: 3842          LDR     4,2
0001A6: 7F60 FF8E     LMD     6,138
0001AA: 7E60 FFBA     STMD    6,168
```

Execute and print two instructions starting at addres location X'1A0' with the bias set to X'100'.

```
BI 100
XT 2,1A0
   AIDS OUTPUT

0000A0R 7830 FF74     LD    2,118
0000A4R 3842          LDR   4,2
```

Execute the next logical instruction without printing the instruction.

```
X
```

Execute and print the next instruction with the bias set to X'100'.

```
XT
   AIDS OUTPUT

0000AAR 7E60 FFBA     STMD   6,168
```

NOTE

The AIDS 'Location Counter' is pointing to the next logical executable instruction.

4.3 Environment Control

4.3.1 Logical Unit

Function: To reassign the logical unit assignments for AIDS if there is a conflict with the logical units used by the program to be debugged.

The Logical Unit command is used with three required (non-default) arguments. The OS should be used to establish the correct physical device assignments prior to using the 'LU' command.

Format:

```
LU U1,U2,U3
```

where: U1 = the decimal number for the Command Input Unit
U2 = the decimal number for the List Output U
U3 = the decimal number of Binary Output Unit

Example: Reassign the list output unit to logical unit 4, and the binary output unit to logical unit 6.

```
LU 5,4,6
```

4.3.2 BIAS

Function: To set or display the address value contained in the memory cell BIASVAL within AIDS. When the bias has been set to a value, any use of the relative address option causes the bias value to be added to the input, address is input, and subtracted in the case of relative addresses when displayed by AIDS.

If the address field is omitted then the value of the bias is displayed, else the bias is set to the hexadecimal value found in the address field.

Format: BI [address]

Example: Set the bias to X'6A0'.

```
BI 6A0
```

Display the current value of the bias.

```
BI
0006A0
```

4.3.3 PAUSE

Function: To allow an exit to the operating system. The operating system CONTINUE command can be used to return to AIDS. All pointers contained in AIDS are not modified.

Format: P

4.3.4 END

Function: To terminate AIDS and return control to the operating system by executing an SVC 3,0.

Format: EN

4.3.5 LOG

Function: To allow the input command stream and error message to be copied to the list unit. This directive is useful in a batch environment.

Format: L

4.3.6 NO LOG

Function: To cancel the Log directive. AIDS is initialized to the no-log mode.

Format: NL

4.3.7 BATCH

Function: To terminate the job and return to the operating system via an SVC 3,1 in the event a directive error occurs.

Format: B

4.3.8 NO BATCH

Function: To turn off the Batch mode. Any error causes AIDS to return to the command input unit for another directive.

Format: NB

4.3.9 TYPE STATUS

Function: To display and allow the updating of the user's task status word (TSW). The information displayed is bits 0 through 31 of the 64 bit task status word. AIDS is left in a mode such that a new task status word can be entered by means of the Replace directive. This is an action affecting internal AIDS tables, therefore no incrementing after the Replace command is allowed. Entry is in fullword hexadecimal format and does not have to be specified by the user.

Format: TS

Example: Display the current TSW and replace it with the new TSW 06007010

```
TS
  00007210
R 6007010
TS
  06007010
```

4.3.10 TYPE CONDITION CODE

Function: To display and allow the updating of the user's program condition code. The information displayed is bits 28 through 31 of the 64 bit task status word. AIDS is left in a mode such that a new condition code can be entered by means of the Replace directive. Entry is in halfword hexadecimal and need not be specified by the user.

Format: TC

Example: Display the current condition code and set it to X'2'.

```
TC
        AIDS response
  3
R 2
```

5. OPERATING INSTRUCTIONS

5.1 Operation Under OS/32-MT

AIDS uses User Dedicated Locations to accommodate SVC 14, arithmetic faults, memory access faults and illegal instruction faults. These UDL locations are X'40' - X'4F', X'60' - X'60' - X'6F', X'80' - X'9F', X'CO' - X'CF' and X'FO' - X'FF'. When these faults occur, registers are saved at user registers 0-15 in an impure area.

AIDS automatically tests for single and double precision floating point capabilities and sets flags to allow or disallow the floating point features. The task status work (TSW) is set for SVC 14, arithmetic fault, memory access fault and illegal instruction fault. Option AFC must be specified when the program is established using TET/32, if the arithmetic fault is to operate properly and return control to AIDS.

Operation as Part of Task

AIDS may be established by TET as part of a user task by including it at establishment time (TET/32 INCLUDE command). AIDS will debug either U-Tasks or E-Tasks when included as part of the task. When starting AIDS as part of the user's task, issue the operating system START command giving the origin address of AIDS within the task's space. Additional storage must be requested by the use of the TET/32 GET command. The command GET B00 or EXPAND 11 allocates the proper additional space required.

Operation as Reentrant Library segment

AIDS may be established and used as a reentrant Library segment which allows it to be shared by all active tasks. As an RL segment, AIDS executes an SVC 2,2 to get storage of X'600' bytes for a work area. A task to be debugged with AIDS should, at TET time, make allowance for the required area with the TET/32 EXPAND Command.

As an RL segment, AIDS debugs only U-Tasks. To start AIDS when established as an RL segment, issue the following command sequence:

```
TASK   TASK ID
START  (REG)XXXX
```

where: (REG) = segmentation register of the system .LIB partition and XXXX is the origin address of AIDS in the library segment. The restart address of AIDS is (REG) XXXX+4 If AIDS is restarted at (REG) XXXX, another X'600' bytes of work storage is requested.

The procedure to establish a task using a reentrant library is outlined in the OS/32 MT Operator's Reference Manual, Publication Number 29-574.

The following example of TET commands establishes the task TEST. TSK as a program using AIDS as a reentrant library. TEST.OBJ is the binary copy of the program to be debugged and OSAIDS.RTL is the reentrant library containing AIDS.

Load the library partition with OSAIDS.RTL. Load and Start TET/32 and enter the following commands from the command input device:

Command	Meaning
ES TA	Establish the task
GET B00	Request additional storage
OPT AFC	Set Arithmetic FAULT Continue
INC TEST (.OBJ)	Load the program
RESOLVE OSAIDS (.RTL)	Resolve AIDS pointers
BU TA, TEST	Build the task
MAP PR:	Display the map
END	End of Job

5.2 Error Messages

The following error messages are printed by AIDS when error conditions are encountered.

Messages	Explanation
BIAS ERR	A relative address computation has yielded an invalid address.
II to INTPRT XXXXXX	The AIDS interpreter has encountered a privileged or illegal instruction in the execution thread. XXXXXX is the 6 digit hexadecimal location counte
BX REG ERR	The interpreter has encountered a BXH and BXLE instruction with R1 field greater than 13.
ILL CMD	The command key on the last directive was not valid.
OS/32 AIDS Rnn-nn	Message issued after AIDS entered initially and memory allocation is complete where n is the current revision level.
DATA ERR	Data or characters of other than the anticipated format found in the directive field.
SNTX	User has violated prescribed directive syntax.
OPT MSNG	Option missing in the directive.
MOD ERR	Information required in the directive is missing.
ILG OPT	An illegal option has been entered.
DUP ERR	Attempt to book a sentinel which already exists.
FULL	Attempt to book a sentinel in a list that already contains eight entries.
NO SUCH	Requested a register that does not exist.
ILG SVC	An SVC 14 has occurred which was not set by AIDS.
I/O ERR XXXX	If an error is encountered during I/O, AIDS notifies the operator with this message. XXXX represents the information returned by the supervisor to the status word of the parameter block associated with the I/O.
LIM ERR	Limits on trace directive overlap already existin trace region.

Messages	Explanation
ILLEGAL START OPTIONS	A misspelled or incorrect key word in START command.
EXCESS/DUPLICATE START OPTIONS	More than the required number of logical units is requested or a previously requested unit is duplicated.
ILLEGAL LU SELECTION	An LU assignment is requested with a negative value or it exceeds 254 decimal.
FILE/DESCRIPTOR ERROR IN START OPTIONS	An error in the typing of the file descriptor is detected or a non-existent file was specified.
INSUFFICIENT SPACE FOR GO OPTIONS	Not enough memory to store user options as input.
ADDRESS OUTSIDE OF ALLOCATED MEMORY	An OPEN or REPLACE command specifies a location outside of the user's segment.

The disposition of all errors is that AIDS returns to command mode and awaits the next directive. The exception is Batch mode which causes AIDS to execute an SVC EOJ to return to the OS.

APPENDIX 1

OS/32 AIDS COMMAND SUMMARY

1. Cell/Register Display and Modification

O(opt) (a)(,d)	Open item or contiguous block.
R(opt) X(,d)	Replace open cell/register with data.
N	Open next cell/register but do not display.
CR (opt)	Open next cell/register and display contents.
-(opt)	Open previous cell/register and display contents.
J(opt)	Jump to address of CURCELL and display contents.
A(opt) n	Add n to contents of CURRENT open cell/register.
A(opt) n_1,n_2	Add n_1 to n_2 and display result.
S(opt) n	Subtract n from contents of open cell/register.
S(opt) n_1,n_2	Subtract n_2 from n_1 and display result.
C(opt_1)(opt_2) X	Convert X to specified data form.

2. Utilities

D(opt) a_1,a_2	Dump contents to list on binary device.
IX(*) a(,d)	Insert breakpoint at a with delay d.
IP(H)(*) a	Protect item a.
IT(B)(*) a_1,a_2	Insert trace with limits a_1 to a_2.
IS(opt)(*) a_1,a_2(,d)	Request snapshot dump at current open cell from a_1 to a_2 with delay d.
Z(opt) (a)	Remove sentinel(s) at address a.
GO (a)	Transfer control to user's program at address a.
W	Print the current value of AIDS location counter.
X(G) (d)(,a)	Execute d instructions starting at address a and print each instruction if T is entered.

3. Environment

LU d,d,d	Set AIDS Logical Units.
BI (h)	Set or display AIDS BIAS Value.
P	Pause to operating system.
EN	Exit to operating system and end job.
L	Log input commands to list unit.
NL	Do not log input commands to list unit.
B	Terminate program on directive errors.
NB	Do not terminate on directive errors.
TS	Display user task status word and allow updating.
TC	Display user current condition code and allow updating.

4. Legend

a,a_1,a_2	Generalized AIDS addresses
d	Decimal Number
h	Hexadecimal Number
( )	Optional Field
X	Real, Decimal, Character, or Hexadecimal Data
opt	Legal Command Options
n	Real, Decimal, or Hexadecimal Number

APPENDIX 2

OS/32 AIDS TERMINOLOGY

The following terms are used widely in this document and their meanings are established here for the reader.

Absolute Refers to a fixed memory address within the user program space.

AIDS Automatic Interactive Debugging System.

Batch Refers to the mode of operation where the command input device is non-keyboard (e.g., card reader). Differs from the interactive mode (keyboard) in command stream logging and error disposition.

Bias The value added to a relocatable address to obtain the true absolute location.

Cell A memory location operated on by AIDS.

CURCELL The most recent cell operated upon by AIDS.

CURMODE The data mode during the last AIDS operation.

Delays Event counters associated with breakpoint and snapshot dumps which allow sentinels to be encountered a controlled number of times before causing actual program interruption.

Displacement A value defined as the difference between an absolute address and the AIDS bias value.

Doubleword A 64-bit memory word which must begin on a doubleword boundary.

Environment Refers to the system under which AIDS is operating, as well as the particular peripheral and Logical Unit assignments at run time.

Faults The arithmetic, machine malfunction, and illegal instruction faults which must be considered under OS/32-MT to insure proper operation of the AIDS interpreter.

Formats The available interpretations of binary data by AIDS.

Fullword A 32-bit memory unit which must begin on a 32-bit boundary.

Generalized AIDS Address Address forms which the user employs to indicate:

- Relative program address
- Absolute program address
- Local relative program addresses
- General purpose registers
- Single precision floating point registers
- Double precision floating point registers

Halfword	A 16-bit memory unit which msut begin on a 16-bit boundary.
Log	The recording of directives and error messages on the list unit.
Logical Cell	Refers to the amount of memory associated with a data type.
Modifiers	Mandatory specification characters immediate following an AIDS command.
Operand Field	A data field included in an AIDS directive.
Options	Non-mandatory command or address modifiers.
Program Space	User's partition.

APPENDIX 3

SAMPLE DUMP FORMATS

```
H-    H DUMP FROM 012344: TO 012384:
    012344: -14272     32 -11703      0 -11767      4   4104 -11767
    012354:      3   4104 -11767      2   4104 -11767      1 -11703
    012364:      6 -14272     41 -11703      5    773 -14272     58
    012374: -15408    256   9013  23343     56 -14272     82 -11703
    012384:      6 -14208      6  16800   -132    773      0  -6112
```

```
F-    F DUMP FROM 012344: TO 012384:
    012344:    100911168      2150985        53769       266248
    012354:   -771162109    269013513       135176   -771162111
    012364:   -766967802   -935329751   -766967803     50710592
    012374:      3851216     16786229   1529806904   -935329710
    012384:   -766967802   -931135482   1101070204     50659328
```

```
C-    C DUMP FROM 012EB8: TO 012F00:
    012EB8:  4241 4C52 4152 4E52 434C 5220 4F52 5852   BALRARNRCLR ORXR
    012EC8:  4C52 4352 5352 4D48 5220 4448 5220 5352   LRCRSRMHR DHR SR
    012ED8:  4C53 534C 4C53 4C50 5357 5220 4D52 4452   LSSLLSLPSWR MRDR
    012EE8:  4C49 5320 4C43 5320 4149 5320 5349 5320   LIS LCS AIS SIS
    012EF8:  4C45 5220 4345 5220 4145 5220 5345 5220   LER CER AER SER
```

```
E-    E DUMP FROM 015000: TO 015040:
    015000:   123,456      .4E-2      85.1001       .129E-5
    015010:   456982       9,87001    14.3          189.6
    015020:   -21.5        -1867.89   .118114E-13   756.21
    015030:   1289.5       -11        .345E-12      .216E-3
    015040:   .789002E-2   .789602    .3456         123.457
```

```
D-    D DUMP FROM 005000: TO 005040:
    005000:       .24074531026496D22       .14806597998898D22
    005010:       -.1924827394061D52       -.56326638318394D-46
    005020:       -.67489226182806D47      -40.156254907208
    005030:       -.96249409446709         -.13449104575638
    005040:       .38491100435209D31       .38491100433459D31
```

```
X-    X DUMP FROM 012EB8: TO 012F00:
    012EB8:  4241 4C52 4152 4E52 434C 5220 4F52 5852
    012EC8:  4C52 4352 5352 4D48 5220 4448 5220 5352
    012ED8:  4C53 534C 4C53 4C50 5357 5220 4D52 4452
    012EE8:  4C49 5320 4C43 5320 4149 5320 5349 5320
    012EF8:  4C45 5220 4345 5220 4145 5220 5345 5220
```

```
Y-  012EB8:   42414C52   41524E52   434C5220   4F525852
    012EC8:   4C524352   53524D48   52204448   52205352
    012ED8:   4C53534C   4C534C50   53575220   4D524452
    012EF8:   4C495320   4C435320   41495320   53495320
    012EF8:   4C455220   43455220   41455220   53455220


A-     A DUMP FROM 012000: TO 012028:
    012000:   4100  F5CA        BAL      0,115CE
    012004:   C6D0  0010        OHI      D,10
    012008:   4300  80C6        B        120D2
    01200C:   C540  002E        CLHI     4,2E
    012010:   2134              BNES     012018
    012012:   C6D0  0020        OHI      D,20
    012016:   230C              BS       01202E
    012018:   C540  003A        CLHI     4,3A
    01201C:   2134              BNES     012024
    01201E:   C6D0  0040        OHI      D,40
    012022:   2306              BS       01202E
    012024:   C540  002A        CLHI     4,2A
    012028:   4330  804E        BE       1207A



Utility- PROTECT:020000:  GR(F)   FR(4)   015000:  FR(2)   FR(0)   GR(1)
         TRACE  :017000: 013000: 012A00: 012800: 012600: 012301: 012200:
         BREAK  :015020: 015010: 01500E: 01500A: 015008: 015004: 015000:
         SNAP   :015000: 015000: 015000: 017000: 017000: 012345: 012345:
```

INDEX

CUT ALONG LINE

PUBLICATION COMMENT FORM

Please use this postage-paid form to make any comments, suggestions, criticisms, etc. concerning this publication.

From ______________________ Date ______________________

Title ______________________ Publication Title ______________________

Company ______________________ Publication Number ______________________

Address ______________________

____

____

FOLD FOLD

Check the appropriate item.

□ Error Page No. ________ Drawing No. ______________

□ Addition Page No. ________ Drawing No. ______________

□ Other Page No. ________ Drawing No. ______________

Explanation:

FOLD FOLD

Fold and Staple
No postage necessary if mailed in U.S.A.

STAPLE STAPLE

FOLD FOLD

NO POSTAGE
NECESSARY
IF MAILED
IN THE
UNITED STATES

BUSINESS REPLY MAIL

FIRST CLASS PERMIT NO. 22 OCEANPORT, N.J.

POSTAGE WILL BE PAID BY ADDRESSEE

PERKIN-ELMER

Computer Systems Division
2 Crescent Place
Oceanport, NJ 07757

TECH PUBLICATIONS DEPT. MS 322A

FOLD FOLD

STAPLE STAPLE