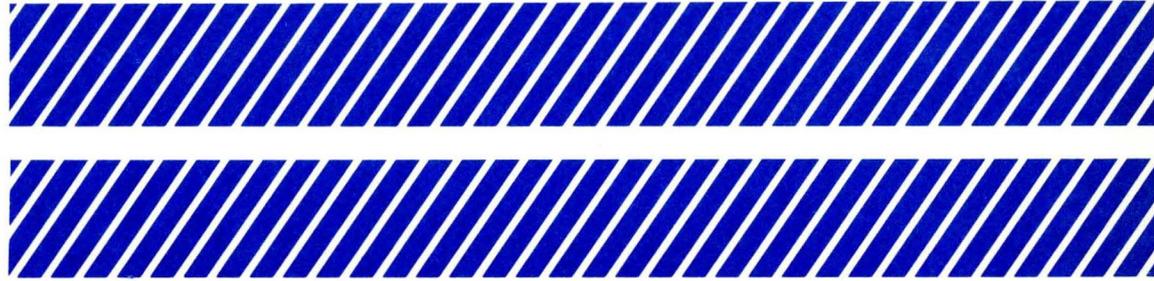


J 500 Diagnostic And Exerciser Programs



J500
DIAGNOSTIC AND
EXERCISER PROGRAMS

Publication V1-071-1

May 1980

PREFACE

This publication describes a set of standard programs for the J500 Videocomputer -- "diagnostics" for testing hardware, and "exercisers" for testing System II software. These programs are intended to be used by field service personnel.

REVISION HISTORY

Number:	Date:	Notes:
V1-071-0	4/79	First edition. Release 8.0 of System II.
V1-071-1	5/80	Revision. Release 8.1 of System II.

The character "!" in the left margin of a chapter title or a section heading indicates substantially new material throughout. The same character in the left margin of a text line indicates a significant spot change. Corrections of minor typographical or stylistic errors on update pages are not indicated.

The pages in the latest revision are listed below. Each page is a replacement or an insertion by page number, unless a deletion is specifically indicated. Note that each page includes its own revision level at the end of the publication number. For a two-sided page, the text on one side may be unchanged.

Title/0-2
TC-1/blank
1-1/blank
2-1/blank
EXFM-1/EXFM-2, EXFM-3/EXFM-4
EXHSH-1/EXHSH-2
EXPRNT-1/EXPRNT-2
J5LDAB-1/J5LDAB-2
J5LDAB-3/blank
J5UDD-11/J5UDD-12

© Copyright AM Jacquard Systems, 1979, 1980

TABLE OF CONTENTS

Chapter 1: INTRODUCTION

Chapter 2: THE PROGRAMS

COMTST	Asynch Controller Diagnostic
EXCIO	Character Output Exerciser
EXCRT	CRT Exerciser
EXFM	File Manager Exerciser
EXHSH	Hash File Exerciser
EXPRNT	Printer Exerciser
EXSB	Scheduler and Buffer Exerciser
EXSILA	SILA Exerciser
J5DMEM	Memory Diagnostic (Stand-Alone)
! J5LDAB	J500 Absolute Loader
J5UDD	Universal Disk Diagnostic
MEMTST	Memory Diagnostic
SKPTST	Skip Function Diagnostic
SPY	Register Monitor

Chapter 1: INTRODUCTION

This publication describes a set of standard test programs for AM Jacquard's J500 Videocomputer. A separate publication -- V1-028 -- covers similar programs for the J100.

The users of diagnostic and exerciser programs are normally the persons responsible for the installation and maintenance of a system. An assumption is therefore made here that the reader is familiar with the details of J500 hardware and software.

DIAGNOSTICS VS. EXERCISERS

In this manual, the term "diagnostic" is used for a program which performs tests at hardware level. The operations of peripheral devices -- disks, printers, and so on -- are thus verified by diagnostic programs, as are certain functions of the CPU and its memory.

Most diagnostic programs run under the control of System II. Those which do not require operator interaction will "detach" themselves as they begin execution -- that is, they will leave the keyboard and screen free for other uses.

A diagnostic described as "stand-alone" is initiated under System II control (using J5LDAB, which is described in this manual), but it then takes over the entire computer. When such a program terminates, System II must be reloaded from a suitable disk.

The term "exerciser" is used for a program which tests some part of the System II software, such as the scheduler or the file manager. Again, some of these programs require an operator; some run in a detached mode.

As the Table of Contents shows, this manual presents all diagnostics and exercisers in a single sequence, alphabetized by program name. To simplify future updates and insertions, each program has its own set of page numbers.

Chapter 2: THE PROGRAMS

All J500 diagnostics and exercisers are distributed on a diskette !
! labelled "TSVl_{nn}-5," in which "nn" is a release number -- "81" is
! Release 8.1, for example. This diskette includes an operating
system suitable for a J500, as indicated by the "-5" suffix.

When System II is loaded from the diskette, and the cursor
appears, the keyboard is open for a program execution command.
Details on command parameters are given in each write-up.

Any test program may also be executed under the control of a user-
! generated version of System II. In this case, the TSVl_{nn}
diskette is usually designated as the nominal device, so that
execution commands are simplified.

Note that J5LDAB can only load a stand-alone program from a
floppy disk, not from a cartridge.

! Some exercisers are written in BASIC, as noted in their
! descriptions. When such a program is to be executed, the BASIC
! run-time support program SRTL_{IB}.SB must be available on the
! current primary device, and the System II configuration must
! support BASIC.

COMTST -- Asynch Controller Diagnostic

COMTST is an execution sharable diagnostic program. It runs in the detached mode, and performs a loopback test of an asynchronous communications controller.

To use COMTST, a loopback plug is required at the end of the device cable. The following signal (and pin) connections are required:

TD (2) -- RD (3)
RTS (4) -- CTS (5) -- DCD (8)
DTR (20) -- DSR (6)
ETC (24) -- TC (15) -- RC (17)

The device to be tested must be defined during system generation under this question:

CHARACTER INPUT/OUTPUT DEVICES,
ASYNCHRONOUS COM LINES

Either at that time, or through the use of the ACUP utility program, certain device specifications must be established before COMTST can be used meaningfully.

The device must be set for internal clocking.

For a J100, COMTST requires --

- 8-bit characters, with no parity.
- a maximum baud rate of 2400.

For a J500, COMTST requires --

- 8-bit characters, with or without parity.
- a maximum baud rate of 19200. However, if the device has characteristics of IE, EM, or DH, the COMTST limit is 4800.

Initiation

COMTST is started -- automatically in background mode, detached from the initiating terminal -- with this System II command:

COMTST device

The "device" field here is the specific device to be tested. The program runs continuously until terminated with this System II command:

KILL COMTST

Description

COMTST sends out a sequence of characters on the communications line and examines the characters that are received via the loopback plug.

Error Messages

No normal messages are produced. If a character is not received (either because there is no loopback plug or an error condition exists) the following messages will be displayed on the terminal screen:

device LOST OUTPUT INTERRUPT	An interrupt did not occur within 0.3 seconds after the character was output.
device - LOST INPUT	An interrupt did not occur within 0.3 seconds of when an input character was expected.
device - BAD INPUT DATA, EXPECTED xx GOT xx	The input data was not as expected. The actual input and expected input are displayed.
device - CHARACTER OVERRUN	The character overrun flag in the device status was on when an input interrupt occurred.
device - FRAMING ERROR	The framing error flag in the device status was on when an input interrupt occurred.
device - LINE DISCONNECTED	There is no DSR signal.

EXCIO -- Character Output Exerciser

EXCIO is an execution sharable program. It tests both the system software and hardware interface for a character-oriented device -- for example:

Line printer

Character printer

Asynchronous line with remote printer or CRT

J100 Multiplexer line with remote printer or CRT

Initiation

To begin execution, enter this command:

EXCIO device count [device count]...

The parameters on the command line must appear in pairs. A pair consists of an output device name, followed by a count -- the number of times that a complete test is to be repeated on that device. Any number of pairs may be specified.

Description

The test consists of outputting a "canned" sequence of output lines to the designated device. These lines are designed to test all of the situations where editing of the output data is performed by the operating system.

The printed output from the test must be visually verified for accuracy by the operator.

A separate task is started for each device to be tested. After all the tasks have started, the message ALL TASKS STARTED is displayed on the terminal. The terminal is then released and the program continues running in the detached mode with all the devices specified being tested simultaneously. As each task terminates, the message TASK DONE is displayed. Error messages are also displayed if an error occurs.

A complete description of the output to each device is given below:

1. The device is opened and one line is printed: OPENING device name.

2. Two blank lines are printed, then one line of each length from 1 to line length characters is printed. The first line consists of one uppercase "A", the second consists of two uppercase "B's", etc.

3. Two blank lines are printed, then one line containing the heading TAB EXPANSION TEST:, then the following four lines, shown here in hexadecimal:

```
4109 4242 0943 4343 0944 4444 4409 4545 4545 4509 4646 4646 4646
0947 4747 4747 4747 0948 4848 4848 4848 4800
```

```
3120 5441 4220 0941 0941 0941 0941 0941 0941 0941 0941 00
```

```
3220 5441 4253 0909 4109 0941 0909 4109 0941 00
```

```
3420 5441 4253 0909 0909 4109 0909 0941 00
```

4. Two blank lines are printed, then a line containing the heading SPECIAL SYMBOLS AND UPPER CASE LETTERS:, then one line consisting of all bytes from X'21 to X'60.

5. Two blank lines are printed, then a line containing the heading LOWER CASE LETTERS:, then one line consisting of all bytes from X'61 to X'7A.

6. Two blank lines are printed, then a line containing the heading FORM FEED TEST (C TERMINATOR):, and terminated with X'0C.

7. A line containing the heading ALL TERMINATORS FROM 0 TO F (EXCEPT 9 AND C) USING WRL:, followed by 14 lines of two characters each, shown here in hexadecimal.

```
3000
3101
3202
3303
3404
3505
3606
3707
3808
410A
420B
440D
450E
460F
```

(Note: 09, Tab, was skipped)

(Note: 0C, Form Feed, was skipped)

8. Two blank lines are printed, then a line containing the heading ALL TERMINATORS FROM 0 TO F (EXCEPT 9 AND C) USING WRS:, followed by 14 lines of two characters each as listed above.

9. Two blank lines are printed, then a line containing the heading RIPPLE TEST:, followed by maximum width lines until each byte from X'20 to X'7F has been printed in each column exactly one time.

10. Two blank lines are printed, then a line containing the heading CLOSING device name; then the device is closed.

Normal Messages

ALL TASKS STARTED One task for each device, number of passes pair in the command line has been started.

TASK DONE Testing of one device was completed and its test task was terminated.

Error Messages

DEVICE NOT FOUND

OPEN ERROR xx device

CLOSE ERROR xx device

UNEXPECTED ERROR RETURN xx device

SPECIFY DEVICE Missing a pair of command parameters.
#-OF-PASSES
IN COMMAND LINE

EXCRT -- CRT Exerciser

EXCRT tests keyboard and screen operations. Operator interaction is required during execution; the terminal may not be used for other functions while executing EXCRT.

Initiation

To begin execution, enter this command:

EXCRT

EXCRT is normally terminated by running the test to completion. However, each phase of the test allows the CANCEL key to bypass that phase.

Description

EXCRT performs a series of tests; each test begins automatically when the previous test is completed, unless bypassed by the operator. Operator visual verification of data on the terminal screen is required to validate most of the tests.

Each of the following notes begins with the message identifying a test.

CRT MEMORY TEST

This test ensures that the memory of the CRT is operating correctly. It places a pattern of characters on the screen beginning in the lower right position (line 24, column 80), shifting left and up a word at a time. Allow the test to run at least until the entire screen is filled. If an error is detected, "CRT MEMORY TEST FAILURE" appears.

ROLL MODE FULL SCREEN TEST

This test rolls a full line of the complete character set up one line from the bottom, continuously, and shifts the bottom line one character to the right with each roll. Visual verification is required to assure that the characters match up on a left diagonal.

ROLL MODE SPLIT SCREEN TEST

This test is identical to the previous one, except that only the bottom 9 lines of the screen are affected.

FULL SCREEN MODE TEST

This test writes to the top line and rolls each line down one, shifting the data to the right one character on each write. Visual verification should detect the shift and that a line rolls from top to bottom identically.

WRITES ON BOTTOM LINE LINES OF EVERY LENGTH TEST

This test writes a line of every length, starting with one character at the far left, to the bottom line of the screen and then rolls up, increasing the length by one character on each write. Visual verification should assure that the characters match on a left diagonal.

WRITES ON BOTTOM LINE WHEN IN FULL SCREEN MODE

This test is identical the previous one, except that only the bottom line is affected.

TAB EXPANSION TEST

The following should appear on the screen:

V	V	V	V	V	V	V
123	1234	12345	123456	1234567	12345678	
A	A	A	A	A	A	A
A		A		A		A
		A				A
						A

If each "A" is under a "1" (or a "V"), then the test succeeded.

TEST OF RDL

This tests the RDL (read line) I/O call. Type in exactly ABCDEF. If you type in an error, a message will prompt you to repeat ABCDEF. If you hit CANCEL, EXCRT will be terminated.

TEST OF RCRTBN

This tests the RCRTBN (read bottom line without roll) I/O function. Hit the RETURN key. If you hit CANCEL, EXCRT will be terminated.

LOGICAL LIGHTS TEST

If this is run on a J500, which has no status lights, it will serve as a test of the audible alarm at the points where the rightmost light would have been accessed. CANCEL may be used to skip the test completely.

TEST OF KEYBOARD INPUT

The program must find out what type of keyboard is being used. Then it tests the accuracy of the keyboard input. Key in exactly the line requested and press RETURN.

TEST OF FUNCTION KEYS

When you press <RETURN> to begin this test, the screen is cleared, and the message PRESS FUNCTION KEY 1 will appear. Press F1. Each of the function keys will be requested in this manner. If the code is not received properly, or if you press the wrong key, the message "GOT FUNCTION KEY xx EXPECTED yy" appears.

TEST MCRT AND WCRT CALLS

When you press <RETURN> to initiate this test, the screen will be erased and a full screen display will appear with a message in the middle asking you to visually verify that the top half matches the bottom half. If an error occurs, the message MCRT OR WCRT ERROR will appear.

LOWERCASE TEST

Type in the requested line and press RETURN.

PUT BRACKETS WHERE YOU WANT PROTECTS

This test fills the screen with periods. Enter some sets of left and right brackets to mark protected areas. Press CANCEL. When the message "THE SCREEN SHOULD NOW BE PROTECTED" appears, verify that the cursor can't be moved into any protected area. Press a function key to repeat; press CANCEL to continue.

TEST OF CRT ATTRIBUTE CODES

This test exercises such programmable features as underlining, blinking, and intensity control.

TEST OF INITIAL ATTRIBUTES

This test checks the attributes which apply when no control codes are in the CRT memory. When the effect of each attribute disappears, press F1 to bring it back, or any other function key to continue. When all attributes have been tested, this final message appears:

END OF CRT TEST

EXFM -- File Manager Exerciser

EXFM is an execution sharable program. It runs in the detached mode, and tests the sequential disk file management facilities of the operating system.

Initiation

To begin execution, enter this command:

- ! EXFM[switches] file1 file2 file3 [P=d]
- ! /A Repeats Phase 1 continuously; overrides /S.
- ! /F Causes a System II FMSG call whenever an error related to
! a UFT is detected. If the error was detected by EXFM
! rather than by System II, the resultant message will have
! a code of 0.
- /N Suppresses the display of normal status messages. Error
messages will still be displayed.
- /S Causes Phase 1 to be skipped.
- /O Causes the four-digit hexadecimal offset from the
beginning of the program to the point of error detection
to be appended to each error message. Intended for use
by maintenance personnel who have source listing.
- file1 The names of three non-existent files must be specified.
- file2 These files will be created, written, read and deleted
- file3 before normal program termination.
- P=d An optional field specifying, in decimal, the maximum
percentage of the disk to be used for FILE1 during phase
2. The default is 3.

Once EXFM is initiated, the terminal is released for other uses. EXFM will execute continuously until terminated by entering this System II command:

KILL EXFM

NORMAL MESSAGES

Following program termination (via the KILL command), the following message will be displayed:

TOTAL # OF ERRORS DETECTED = hex count

During execution of EXFM, the following message will appear each time that Step 11 of Phase 2 is executed:

TASK DONE xx

! Here, "xx" is a hex value: 1, 2, or 3 (Task 1); 4, 5, or 6 (Task
! 2); 7, 8, or 9 (Task 3).

DESCRIPTION

Phase 1A (Task 1)

Task 1 performs the following operations in the sequence shown:

1. Creates an empty FILE1.
2. Opens FILE1 for reading and writing without sharing.
3. Attempts to open FILE1 for reading and writing without sharing; should receive error code 39 -- "open options prohibit request."
4. Closes FILE1.
5. CHATR's FILE1 to read only.
6. Opens FILE1 for reading.
7. Closes FILE1.
8. Attempts to open FILE1 for writing; should receive error code 38 -- "attributes prohibit request."
9. CHATR's FILE1 to write only.
10. Opens FILE1 for writing.
11. Closes FILE1.
12. Attempts to open FILE1 for reading; should receive error code 38 -- "attributes prohibit request."

ERROR MESSAGES

When an error is detected during Phase 1A, execution continues. In contrast, when an error is detected during Phase 1B or Phase 2, EXFM is terminated; a message shows the total error count.

Error messages have the following format:

text xx file

text is one of the messages in the following list.

xx System Error Code in hexadecimal.

file the name of the file associated with the error.

<u>Text</u>	<u>Notes</u>
CREATE	Phase 1A, Step 1 Phase 1B, Step 2 Phase 2, Step 2
OPNR	Phase 1A, Step 2
OPNR OF OPENED NO-SHARE	Phase 1A, Step 3
SHOULD BE ERFOP	Phase 1A, Steps 3, 21
CLOSE	Phase 1A, Steps 4,7,11,17,18,22,23 Phase 1B, Steps 4,7 Phase 2, Steps 7,10
CHANGE ATTRIBUTE	Phase 1A, Steps 5,9,13
OPNR OF READ ONLY	Phase 1A, Step 6
OPNW OF READ ONLY	Phase 1A, Step 8
SHOULD BE ERFAP	Phase 1A, Steps 8,12
OPNW OF WRITE ONLY	Phase 1A, Step 10
OPNR OF WRITE ONLY	Phase 1A, Step 12
OPNR OF SHARED	Phase 1A, Step 14
OPNW OF SHARED	Phase 1A, Step 15
OPNR ONLY OF FULL SHARE	Phase 1A, Step 16
SHOULD BE ERFSH	Phase 1A, Step 16

OPNW FULL OF FULL SHARE	Phase 1A, Steps 19,20
OPNR OF OPEN FOR WRITE SHARE	Phase 1A, Step 21
DELETE	Phase 1A, Step 24 Phase 1B, Step 8 Phase 2, Step 16
! FBF or GBF	Phase 1B, Step 1
OPNR OR OPNW	Phase 1B, Steps 2,5 Phase 2, Steps 5,8
WRITE LINE	Phase 1B, Step 3 Phase 2, Step 6
SHORT LINE	Phase 1B, Steps 3,6 Phase 2, Steps 6,9
LINE FORMAT	Phase 1B, Steps, 3,6 Phase 2, Steps 6,9
READ LINE	Phase 1B, Step 6 Phase 2, Step 9
UNEXPECTED RECORD READ	Phase 1B, Step 6
CHATR	Phase 2, Step 3
UNEXPECTED TIMEOUT RETURN	Phase 2, Steps 2,15
NO INTERLEAVING IN SHARED FILE	Phase 2, Step 9
DISK FULL	Phase 2, Step 6. This is not counted as an error. FILE1 is deleted; Phase 2 is restarted.
ONLY switches ARE ALLOWED	Unacceptable switch on command line which invoked EXFM.
ERROR: FILE IS A DEVICE	The specified FILE1 is a device. It must be a valid disk file name.
HIGH-TO-LOW STACK OVERFLOW!	This message should never be seen during normal program operation.
MESS BUF OV!	This message should never be seen during normal program operation.
TIMEOUT RETURN	This message should never be seen during normal program operation.

!

EXHSH -- Hash File Exerciser

EXHSH is an execution sharable program. Using keyboard commands, the hash file facilities of System II can be tested.

The operator must provide an existing hash file with 4-word keys; any data in the file must be ASCII characters suitable for a display. This file will be altered during the test. Note that an empty file to hold "n" records can be created with the standard utility HCREAT:

HCREAT device:name/D K=4 F=n

The execution request for EXHSH specifies the file to be used:

EXHSH device:name

EXHSH may be terminated at any time by pressing <CANCEL>.

Description

When execution starts, this layout is displayed:

HASH FILE EXERCISER

KEY

DATA

F1 = DISPLAY FILE
F2 = INQUIRY
F3 = ADD
F4 = DELETE
F5 = REPLACE
F6 = NEXT RECORD
F7 = INQUIRY WITH RECORD LOCK
F8 = NEXT RECORD WITH RECORD LOCK

The cursor will appear at the KEY field. To add a new record, enter an 8-character key, press <RETURN>, fill in the DATA field, and press <F3>. Follow similar procedures to edit the file as desired.

The test should include several executions of each function. If a problem occurs, an error message appears on the bottom line of the screen. In some cases, these will be diagnostic messages prompting correct use of the functions.

Message

Meaning

UNIMPLEMENTED FUNCTION

Function Key not recognized.

COMPLETE - PRESS F6 OR
F8 FOR NEXT RECORD

Function was completed.

RECORD NOT AVAILABLE -
RETRY OR RE-ENTER

The record requested on an
inquiry was not found in
the file.

Messages may also appear in which a numeric code is used; see the Error Messages manual for details. As an example, if the last record in the file has just been accessed with the INQUIRY function <F2>, then requesting NEXT RECORD with <F6> would cause an end-of-file error:

*** EXHSH ***** FILE ERROR 80 filename

EXPRNT -- Printer Exerciser

EXPRNT is a program which tests line printers and character printers. The test consists of outputting lines of characters to the specified printer. Each line ends with a Carriage Return; the last line on each page is followed by a Form Feed character. Page size, line length, and data patterns are user-controlled.

Initiation

To begin execution, enter this command:

EXPRNT

This message appears:

DO YOU WANT TO SET LINE LENGTH OR PAGE SIZE?

The default line length of 80 characters and page size of 55 lines will be displayed. Enter "NO" if the default values are to be used; enter "YES" if you wish to define line length and page size, or select the "Expanding Line" option. Press any function key.

With the "Expanding Line" option, the line length will start at "1" -- one character on a line. Each subsequent line will be one character longer than the one before, until the line length specified by the operator is reached. The next line will be one character long and the sequence will be repeated.

Skip the following section if a "NO" response was entered.

Setting Line Length and Page Size

1. Enter the number of characters to be printed on each line or use the default number displayed, press any function key.
2. Enter the number of lines to be printed on each page or use the default number displayed, press any function key.
3. Answer the question "Expanding Line?" either "YES" or "NO" and press any function key.

Select Output Device

! Enter the name of the printer to be tested. Examples: LPT1,
! DPR1, ASY1, NPPI, DH04.

Select Data Pattern

There are four different data formats available. The first three are "canned" and the fourth allows the operator to compose the line.

1. Ripple characters across line. Each line will consist of an ascending ASCII sequence (1234567...ABCDEFGH, and so on). Where one line leaves off, the next line will begin. The actual characters printed in this sequence will depend on the print font being used.

2. Ripple characters by line. Each line will consist of only one type of character. The character on each line will be one ASCII increment from the character on the preceding line. The sequence will repeat after the character equivalent to hex 7F is reached.

3. Repeat a specific character. Each line will consist of the character selected by the operator.

4. Line to be entered by the operator. Each line will consist of a string of characters defined by the operator.

Enter the number (1, 2, 3, or 4) of the data format you desire and press any function key.

If data format 1 or 2 was selected, printing will commence immediately, and continue until the CANCEL key is pressed. If data format 3 or 4 was selected, the operator will be asked to enter the characters to be printed.

Stopping the Printer

The print operation may be terminated at any time, to change parameters or end the program, by pressing the CANCEL key. This message will appear:

CONTINUE WITH TEST?

Enter YES to continue testing or NO to end EXPRNT.

EXSB -- Scheduler and Buffer Manager Exerciser

EXSB is an execution sharable program. It runs in the detached mode, and tests the scheduler and buffer manager portions of the operating system.

Initiation

To begin execution, enter this command:

```
EXSB[/N][/O] [K=nnnn] [L=nnnn]
```

/N An optional switch to suppress the printing of normal status messages. Error messages will still be generated.

/O An optional switch which causes the four-digit hexadecimal offset from the beginning of the program to the point of error detection to be appended to each error message. This facility is intended for use only by maintenance personnel.

K=nnnn An optional decimal number to override the default number of tenths of a second to suspend in SUSZ when the buffer management test has exhausted the system buffer pool. The default value is 100.

L=nnnn An optional decimal number to override the default number of tenths of a second to suspend in SUSZ when the buffer management test has released all its system buffers. The default value is 100.

Description

Once EXSB is initiated, the terminal is released for the performance of other functions.

EXSB performs four tests continuously. Each of these tests is described in detail below for purposes of error diagnosis.

CAUTION: In the following description, whenever the same memory location is referred to in two or more places, it is given a symbolic name and is referred to by this symbolic name. This procedure is familiar to every assembly language programmer.

BUFFER MANAGEMENT TEST

1. Set BTNW to 0.
2. Set BTLB to 0. Set BTCT to 8.
3. Set BITS to 1.
4. Set BTCT to BTCT - 1.
5. If BTCT = 0, then set BTCT to 8 and suspend.
6. Call GBF with buffer size argument bits.
7. If GBF takes error return, go to step 13.
8. Set the first word in the buffer to BTLB. Set the second word in the buffer to the number of words in the buffer. Add the number of words in the buffer to BTNW.
9. Set BTLB to the buffer address. Set all but the first 2 words of the buffer to BTLB + the task buffer address.
10. Set BITS to BITS + 1.
11. If BITS is the GBF argument for a buffer size > the maximum buffer size allowed, then go to step 3.
12. Go to step 4.
13. If the error code is not ERBNA, then output an error message.
14. If BITS is not equal to 1, then go to step 10.
15. If BTNW=0, then go to step 1.
16. Suspend in SUSZ for K tenths of a second. The default value of K is 100. This may be overridden by including K= in the command line.
17. Check that the contents of the buffer whose address is BTLB has not changed.
18. Set BTLB1 to the contents of address BTLB.
19. Zero the buffer whose address is BTLB.
20. Call FBF to free the buffer whose address is BTLB.
21. Set BTCT to BTCT - 1.
22. If BTCT=0, then set BTCT to 8 and call SUSP.

23. If BTLB1 is non-zero, set BTLB to BTLB1 and go to step 17.
24. Suspend in SUSZ for L tenths of a second. The default value of L is 100. This may be overridden by including L= in the command line.
25. Call FBF to re-release the buffer whose address is BTLB.
26. If FBF makes its normal return, then output an error message.
27. If FBF makes its error return with any error code other than ERBNJ, then output an error message.
28. Call FBF to release the buffer whose address is BTLB+1.
29. If FBF makes its normal return, then output an error message.
30. If FBF makes its error return with any error code other than ERBIA, then output an error message.
31. Call GBF with buffer size argument - 1.
32. If GBF makes its normal return, then output an error message.
33. If GBF makes its error return with any error code other than ERBIS, then output an error message.
34. Call GBF with buffer size argument 1 larger than the maximum allowed.
35. If GBF makes its normal return, then output an error message.
36. If GBF makes its error return with any error code other than ERBIS, then output an error message.
37. If the /N switch was not used, then output the message BUFFER MANAGEMENT TEST COMPLETE.
38. Go to step 1.

The following error messages might appear in conjunction with this test:

<u>Message</u>	<u>Notes</u>
CONTENTS OF A BUFFER CHANGED	Step 17.
ERROR: RELEASED BUFFER RE-RELEASED	Step 26.
ERROR: ILLEGAL BUFFER ADDRESS RELEASED	Step 29.
ERROR: GBF BUFFER SIZE -1	Step 32.
ERROR: GBF BUFFER SIZE 1 TOO BIG	Step 35.
BUFFER MANAGEMENT UNEXPECTED SYSTEM ERROR	System message next. Steps 13, 27, 30, 33, 36.
SUSZ UNEXPECTEDLY TOOK ITS NORMAL RETURN	Steps 16, 24.

SUSPEND CAPABILITY TEST

1. Call GTOD.
2. Save the time in seconds.
3. Set R1 to 11.
4. Call SUSP.
5. If SUSP returns via RTS 1, then output an error message.
6. Set R1 to the contents of address A1; set R2 to address A1.
7. Call SUSC.
8. If SUSC takes its normal return, then output an error message.
9. Call SUSZ.
10. If SUSZ takes its normal return, output an error message.
11. Set R2 to the address of word containing 0.
12. Call SUSN.
13. If SUSN takes its normal return, output an error message.
14. Set R0 to the contents of address A1; set R2 to address A1.
15. Call SUST.
16. If SUST takes its normal return, output an error message.
17. Call GTOD.
18. If less than 4 seconds elapsed between the calls to GTOD, then output an error message.
19. Set R0 to 1, R1 to -1, and R2 to the address of a word containing 0.
20. Call SUSC.
21. If SUSC takes its timeout return, output an error message.
22. Set R0 to 0 and R2 to the address of a word containing 1.
23. Call SUSC.
24. If SUSC takes its timeout return, then output an error message.
25. Set R2 to the address of a location containing 0.

26. Call SUSZ.
27. If SUSZ takes its timeout return, output an error message.
28. Set R2 to the address of a location containing 1.
29. Call SUSN.
30. If SUSN takes its timeout return, output an error message.
31. Set R0 to 6, and R2 to the address of a word containing 0.
32. Call SUST.
33. If SUST takes its timeout return, output an error message.
34. If R0 was changed, then output an error message.
35. If the address in R2 was not set to the contents of R0, then output an error message.
36. Set R0 to 100 and R1 to 20.
37. Start a task (by calling TASK) that outputs an error message if R0 does not equal 100, increments location SEX to 100, and terminates (by calling TEND).
38. Set location SEX to 100 and R2 to address SEX.
39. Call SUSC.
40. If SUSC takes its timeout return, output an error message.
41. If R0 does not equal 100, then output an error message.
42. If location SEX contains 100, then output an error message.
43. Start a task (by calling TASK) that zeros location SEX.
44. Set R2 to address SEX.
45. Call SUSZ.
46. If SUSZ takes its timeout return, output an error message.
47. If location SEX does not contain 0, output an error message.
48. Start a task that increments location SEX by 1, then terminates.
49. Set R2 to address SEX.
50. Call SUSN.
51. If SUSN takes its timeout return, output an error message.

52. If location SEX contains zero, output an error message.
53. Start a task that zeros R2 and location SEX, then terminates.
54. Set R0 to -1, R2 to address SEX and the contents of location SEX to 6.
55. Call SUST.
56. If SUST takes its timeout return, output an error message.
57. If location SEX does not contain -1, output an error message.
58. If the /N switch was not used, then output the message SUSPEND CAPABILITY TEST COMPLETE.
59. Go to step 1.

The following error messages might appear in conjunction with this test:

<u>Message</u>	<u>Notes</u>
NORMAL RETURN TAKEN WHEN TIME OUT RETURN EXPECTED	Steps 5, 8, 10, 13 or 16.
TOO LITTLE TIME ELAPSED	Step 18.
TIME OUT RETURN TAKEN WHEN NORMAL RETURN EXPECTED	Steps 21, 24, 27, 30, 33, 40, 46, 51 or 56.
SUSPEND CALL CHANGED AC0	Step 34 or 41.
SUSPEND DIDN'T SET LOCATION	Step 35, 42, 47, 52 or 57.
SUSPEND SET LOCATION	Step 33.
TASK DIDN'T SET REGISTERS CORRECTLY	Step 37.

SUSPEND INHIBIT TEST

The following steps comprise this test:

1. Set location SICT to 16,384.
2. Start a TASK that outputs an error message if location SICT is non-zero, decrements location SICT by 1 and terminates.
3. Call SUSX.
4. Decrement location SICT by 1.
5. If location SICT is non-zero, then go to step 4 -- that is, SICT is counted down to zero to generate a real-time delay.
6. Call SUSX.
7. Set R1 to 30 and R2 to address SICT.
8. Call SUSN.
9. If SUSN takes its timeout return, then output an error message.
10. If the /N switch was not used, then output the message SUSPEND INHIBIT TEST COMPLETE.
11. Suspend for 30 seconds by calling SUSZ on a non-zero location.
12. Go to step 1.

The following error message might appear in conjunction with this test:

<u>Message</u>	<u>Notes</u>
SUSX FAILED	Step 9.

SUSPEND AND TIME SLICE TEST

1. Set STFVAL to -16.
2. Pull the hardware stack 16 times (i.e., empty it); push onto the hardware stack, in order: 5, 4, 3, 2, 1.
3. Set the status flags to STFVAL.
4. Set SEL to bit 15 of STFVAL.
5. Zero location STCT.
6. Set Registers R0, R1, R2 and R3 to known values.
7. Decrement location STCT by 1.
8. If location STCT is non-zero, then go to step 7 -- that is, STCT is counted down to zero to cause a real-time delay.
9. If R0 has changed since step 6, then output an error message.
10. If the status flags do not equal STFVAL, then output an error message.
11. Call SUSP.
12. If R0 has changed since step 6, then output an error message.
13. If SEL does not equal bit 15 of STFVAL, then output an error message.
14. If R0, R1, R2 or R3 has been changed since step 6, then output an error message.
15. Pull the hardware stack 5 times, and each time the value pulled does not equal the expected value (1, 2, 3, 4, and 5, in order, should be obtained), output an error message.
16. Suspend for 5 seconds by calling SUSZ on a non-zero location.
17. If SUSZ takes its normal return, then output an error message.
18. If the /N switch was not used, then output the message SUSPEND/TIME-SLICE TEST COMPLETE.
19. Decrement STFVAL by 1.
20. Go to step 2.

The following error messages may appear in conjunction with this test:

<u>Message</u>	<u>Notes</u>
REGISTER DESTROYED	Step 9, 12 or 14.
STATUS FLAGS DESTROYED	Step 10.
SEL DESTROYED	Step 13.
HARDWARE STACK DESTROYED	Step 15.
SUSZ UNEXPECTEDLY TOOK ITS NORMAL RETURN	Step 17.

GENERAL ERROR MESSAGES

<u>Message</u>	<u>Notes</u>
HIGH-TO-LOW STACK OVERFLOW	This error message should never be observed. It indicates an error in program EXSB. Return is made to System II.
ONLY /N & /O ARE ALLOWED	The command line contained some switch other than /N or /O. Return is made to System II.

TERMINATION

EXSB will continue to run until cancelled by entering the System II command:

KILL EXSB

Upon termination, this message will be displayed:

TOTAL # OF ERRORS DETECTED = hex

NORMAL MESSAGES

If the /N switch is not entered in the command line, the following messages will appear at the initiating terminal repeatedly, in random order, as each phase of EXSB is completed.

BUFFER MANAGEMENT TEST COMPLETE xx

SUSPEND CAPABILITY TEST COMPLETE xx

SUSPEND INHIBIT TEST COMPLETE xx

SUSPEND/TIME-SLICE TEST COMPLETE xx

In each case, "xx" is a hexadecimal count of test completions.

The first message will appear less often than the others, because this test runs for a longer period of time.

EXSILA -- SILA Exerciser

EXSILA is an execution sharable program. It runs in the detached mode, and tests the Synchronous-Isochronous Line Adapter (SILA).

Initiation

A loop-back plug must be installed at the end of the device cable. For specifications, see the COMTST write-up. Then enter this command:

```
EXSILA[/A] [device] [S=sync] [F=freq] [T=delay]
          [L=length]
```

- /A Indicates that option switches on the SILA are set to run in asynchronous mode. If /A is not entered, EXSILA assumes synchronous operation.
- device Name of communications line to be used; the default is COM1. The device name entered (or COM1 if the default is assumed) must have been defined in SYSGEN in response to the question: GENERAL SILA COMMUNICATION LINES? The device must currently be set for internal clocking.
- sync Synchronization character, in hexadecimal, which must be less than hex 20; the default is hex 12. On a J500, a value of 16 will cause spurious errors if used with 5-bit characters.
- freq The SILA device is closed and opened after every n complete tests, where n = the open frequency. If n = -1, the SILA remains open indefinitely. The default value of n is +1.
- delay Length of time, in tenths of a second, that the exerciser is suspended after each error message. This feature prevents error messages from being displayed so fast as to be unreadable. The default delay is 10.
- length The length, in characters, of the test message sent and received by the exerciser (not including the synchronization character). This value must be less than or equal to 512. The default is 256.

All parameters are optional. All values are specified in decimal except the sync character.

Once EXSILA is initiated, the terminal is released for the performance of other functions. EXSILA will run continuously until terminated by entering this System II command:

KILL EXSILA

DESCRIPTION

EXSILA tests the following three functions continuously:

1. Simultaneous transmission and reception from two separate tasks using TMS and RMS, respectively.
2. Simultaneous transmission and reception from a single task making calls to IRMS, TMS and RMS in that order.
3. An RMS timeout test which is a single call to RMS that times out.

ERROR MESSAGES

All errors are non-fatal except where noted below. The general format of all error messages is:

*****EXSILA name - n message

name	The name of the SILA board being used.
n	The error code, as noted below.
message	The text of the message.

COMMAND PARSER ERROR	An error return was taken from the system call to the command parser with error code n (fatal).
OPEN SILA ERROR	An error return was taken from the system call to OPEN (to open the SILA) with error code n (fatal).
CLOSE SILA ERROR	An error return was taken from the system call to CLOS (to close the SILA) with error code n (fatal).
ERROR ON TASK CREATION	An error return was taken from the system call to TASK (in Test No. 1) to create a second task; n was the error code returned.
TEST 1 RMS ERROR RETURN	The RMS system call in Test No. 1 took an error return with error code n.
TEST 2 IRMS ERROR RETURN	The IRMS system call in Test No. 2 took an error return with error code n.
TEST t TMS ERROR RETURN	The TMS system call in Test No. t took an error return with error code n.

RMS TIMEOUT TEST
RETURN

The RMS system call in the RMS timeout test either took the normal return (n = 00) or took an error return but the error code was not ERCMTO (n = error code not returned).

n CHARS EXPECTED,

A test message received by the m CHARS RECEIVED exerciser was not the length expected; n = length expected and m = length received (including synchronization characters).

n BAD CHARACTER
RECEIVED, EXPECTING m

A character (n) in a SILA message received was not equal to its expected value (m). Only the first such character in a given message elicits this error display.

J5DMEM -- Memory Diagnostic (Stand-Alone)

J5DMEM is a stand-alone memory diagnostic. It can be used to test all of memory above address Hex 00FF. The addresses below Hex 00FF are used by the program and must not be included in the range of memory addresses to be tested.

1. Load System II from a floppy containing both J5DMEM and J5LDAB, the J500 Absolute Loader.

2. Enter this System II command line:

 \$J5LDAB \$J5DMEM

3. J5DMEM will begin execution, still under System II.

4. This will be displayed:

 DYNAMIC STAND-ALONE MEMORY DIAGNOSTIC

FIRST ADDRESS	LAST ADDRESS	DATA PATTERN	
LOOP DIAG?	TEST ALL?	TEST NO.	PRINT?

5. FIRST ADDRESS

The cursor will indicate the entry for the first address to be tested in the range selected by the operator. The address displayed at this time is absolutely the lowest that should be tested, as this is the first address after the program. Enter the first address to be tested, or if the address currently displayed is the one desired. Press <RETURN>.

6. LAST ADDRESS

Now enter the last address to be tested. This address should not exceed the last address available in memory. In no case should this address be defined above Hex FB00. This is the last memory address before the CRT memory. Press <RETURN>.

7. DATA PATTERN

Now enter the Data Pattern. This will be four Hex digits that will be written into all memory addresses (that were specified in first and last memory address entries) during test 6.

8. LOOP DIAG

If the operator wishes the test to run continuously, enter "Y".
If the test is to run one pass only, enter "N".

9. TEST ALL

There are eleven different tests (data patterns stored) making up one "pass" of the diagnostic program. The tests are identified as Test 0 through Test A. The data patterns in each test are:

Test 0 - The Hex word 0000 is written into the addresses to be tested. Then, these addresses are read and any errors are reported.

Test 1 - The Hex word FFFF is written into the addresses to be tested. Then, these addresses are read.

Test 2 - The Hex word 0000 is written into the first address to be tested and the Hex word FFFF is written into second address. These two words are alternately written in this manner throughout the memory range to be tested. A read follows the write operation.

Test 3 - The Hex word F0F0 is written and read in each address to be tested.

Test 4 - The Hex word 0F0F is written and read in each address to be tested.

Test 5 - The Hex word AAAA is written into the first address to be tested and the Hex word 5555 is written into the next address. These two patterns are alternately written throughout the memory range to be tested.

Test 6 - The four Hex digits entered as "Data Pattern" will be written into and read from the memory range to be tested.

Test 7 - A Hex word that is equal to the address of the memory location to be tested will be written into and read from each address to be tested.

Test 8 - First, the Hex word 0001 will be written and read from each memory address to be tested. Then the word will be rotated left one bit to make the Hex word 0002 and this pattern is written and read. This process will be repeated sixteen (16) times until the Hex word 8000 has been used as the test pattern. Please note that while the previous tests made one write/read pass through the memory range to be tested, this test will make 16 passes. This test therefore will take 16 times longer to run.

Test 9 - This test will start with a Hex word of FFFE and will proceed just as in test 8. The zero in the extreme right bit position will be shifted left after each write/read sequence ending with a pattern of Hex 7FFF. This test will make 16 write/read passes.

Test A - The data pattern used by this test is a sequence of 31 Hex words. The 31 words are written, starting with the first address to be tested. The sequence will be repeated every 31 memory locations. When the end of the range to be tested is reached, the data will be read and verified. The above sequence will then be repeated but will start with the second address instead of the first. This will have the effect of displacing the sequence of 31 words forward by one memory address throughout the range of memory to be tested. This process will continue until the data sequence has been written and read 31 times. The sequence of Hex words used in this test is:

```
0000 FFFF 0000 0000 FFFF 0000 0000 0000
FFFF 0000 FFFF FFFF 0000 FFFF FFFF 0000
0000 0000 FFFF 0000 FFFF 0000 FFFF 0000
FFFF 0000 FFFF 0000 FFFF 0000 FFFF
```

If it is desired to run all of the above tests enter "Y". If one particular test pattern is to be selected, enter "N". Press the <RETURN> key.

10. TEST NO.

Enter the desired test number. This entry is allowed only if the response to the Prompt "Test All" was "N". Otherwise the cursor will skip over this entry. Press the <RETURN> key.

11. PRINT?

If a Diablo Character Printer is available, enter "C", if not enter "N". Press the <RETURN> key.

The Diagnostic Program will now proceed to perform the selected test(s) on the selected range of memory addresses. When the last test in the sequence is finished, the program will move itself from starting address Hex 100 to starting address Hex 1100. The Diagnostic will then perform the same test(s) on address range Hex 100 to Hex 10F0 (this is where the program was located when the range of memory addresses defined by the operator was being tested).

The above operation consists of one test "pass". The pass count on the Master Screen indicates the number of the above sequences that have been performed. If during a test an error is detected, the Error Address, Stored Data, Read Data, and the test in which the error occurred will be displayed.

TIME DISPLAY

The time, relative to starting the Diagnostic, is displayed on the left side of the Master Screen.

STOPPING THE TEST

To interrupt the test at any time press the "CANCEL" key. This will return to the start of the Diagnostic where new test parameters may be entered.

FLOPPY DIAGNOSTIC

While the Memory Diagnostic is running, device FP01 -- which must be a single-density floppy -- is being tested. This is to introduce DMA activity that may bring out memory problems. THE FLOPPY IN FP01 MUST BE A SCRATCH DISK; WRITING WILL BE DONE.

If no diskette is placed in FP01, the message "DISK MUST BE FLOPPY" will appear on the screen when the Memory Diagnostic is initiated. This is only an indication that the floppy will not be tested and should not be considered an error. If an error is detected on the floppy, the Cylinder and Sector where the error occurred will appear on the Master Screen.

As with Memory errors, the Floppy errors will be reported on the Character Printer if "C" was the response to the "PRINT ?" prompt.

J5LDAB -- J500 Absolute Loader

On a J500 only, J5LDAB performs the loading and execution of a stand-alone program -- that is, a program which does not run under the control of System II. Such programs, typically, are field service aids.

Command Line Format

J5LDAB[/D] pgmfn [S=addr]

/D loads the System II Debugger (\$DEBUGX.SB) into high memory after "pgmfn" is loaded, and gives control to the Debugger. The DEBUGX.SB file must be on a floppy disk which is currently the primary disk.

pgmfn is the file name of the program to be loaded and (if "/D" does not appear) executed immediately. Only a file on a floppy disk is accepted.

S=addr overrides any start address in the program file. A hex number is expected; a decimal prefix -- as in D'256 -- is allowed.

If there are any other jobs, or any disk activity, an appropriate error message is displayed; the command is rejected. Otherwise, a REMOVE command is implicitly issued for every disk device (that is, the Sector Management Table is flushed). Then this message is displayed:

SYSTEM II DISABLED

Subsequent use of System II will require reloading it from disk.

The specified program is loaded with BSECT reference address 0 and TSECT reference address X'100. ASECT code is loaded as appropriate.

Abort Messages

If one of the following messages appears, System II is not disabled, and the utility is abnormally terminated.

OTHER JOBS EXECUTING

Jobs other than the utility are still executing.

DISK ACTIVE

Some disk device still has a non-zero user count.

DISK MUST BE FLOPPY

The program to be loaded must reside on a floppy drive. If the "/D" switch appears, the primary device must be a floppy drive.

ONLY /D ALLOWED

An unacceptable switch has been used in the command line.

TSECT OVERLAP

There is insufficient memory for the program being loaded.

BSECT OVERLAP

More than 256 BSECT words were required.

MUST BE INVOKED FROM A CRT

The utility can not be invoked from a remote asynch terminal.

Total Abort Messages

If one of the following messages appears, the utility failed after System II was disabled. The system must be reloaded.

DISK ERROR

A disk error occurred; 5 retries could not overcome it.

WRONG BUFFER

An invalid buffer was released. This message is never seen in normal operation. It may indicate a malfunctioning memory.

WRONG SIZE

A buffer of invalid size was requested. This message is never seen in normal operation. It may indicate a malfunctioning memory.

NO START ADDRESS

The utility command had neither a "/D" switch nor an "S=" field. The specified program had no start address in its end block.

J5UDD -- Universal Disk Diagnostic

J5UDD is a diagnostic program for any floppy or rigid disk drive attached to a J500. Under operator control, a single execution of J5UDD can test all drives of a given device type for I/O operation and for general reliability. The program may be "released" -- detached from the initiating terminal -- allowing the concurrent execution of another program, including a J5UDD test for a different device type.

Initiation

For each drive to be tested, a disk medium must be inserted before executing J5UDD.

Each floppy disk must already have "soft" sector headers, whether initialized by the manufacturer, by the FLOPIZ utility, or by the FORMAT utility.

Headers may be written onto a factory-fresh cartridge disk with FLOPIZ, with FORMAT, or with J5UDD itself, using the "HI=1" parameter and the "WA" test described later.

During J5UDD execution, a drive under test is not available to any other job. Furthermore, once J5UDD has written onto a disk, a subsequent run of FORMAT will be necessary before any other program can safely use that disk again.

To initiate execution, enter this System II command:

```
J5UDD dsk [L=fn]
```

- dsk Any attached disk of the device type to be tested. For example, naming FP00 is sufficient to prepare this execution of J5UDD for testing all single-density floppy drives, but no other device type. A J5UDD command of "D=" is used to specify the actual drive, as described later.
- L=fn This optional field names a printer (or a disk file not on a drive to be tested). J5UDD screen output, except for certain disk address messages, will then be recorded for later review.

Description

When the System II command line is executed, the program displays its name and last revision date, followed by a blank line containing the cursor. At this point, the operator may provide information by keying in command lines. A command line may contain any combination of commands, followed by a <RETURN> key.

If J5UDD is invoked from a command file, it reads its command lines from that file rather than from the keyboard. An end-of-file is treated as a Q (quit) command and returns to be returned to System II.

Pressing the <CANCEL> key terminates the current operation and leaves the program waiting for the next operator command.

Examples

The following command line seeks first to cylinder 40, then to cylinder 195, exactly once:

```
C=40 SC C=195 SC
```

The "*" command causes the current command line to be re-executed from its beginning. Hence, the following command line seeks cylinder 40, then cylinder 195, repeatedly until the <CANCEL> key is pressed:

```
C=40 SC C=195 SC *
```

Control Commands

D Display parameters.

M Display error switches:

ON: C1 C2 C3 ... Cn
OFF: D1 D2 D3 ... Dn

By default all error switches are on. To turn off an error switch, set its corresponding error code to 0. To turn on an error switch, set its corresponding error code to any non-zero value. When the error switch corresponding to an error code is off, the error code is never reported. Example:

BZ=0 (turn off BZ)
BZ=1 (turn on BZ)

Q Quit; return to System II.

R Release the initiating terminal; continue executing the command line (and/or command file) in the background. After the terminal is released, only error messages are sent to it. For example, after the terminal is released, commands D, Y and PS will not generate any output to the terminal (if no errors occur) but will generate output to the listing file (if any).

U Update initial random number.

Y Display current status.

* Execute the current command line from its beginning.

Input Parameters

The following input parameters may be entered; each consists of a letter, an equal sign, and a value. Where a fixed default exists, it is shown here after the equal sign. All values are assumed to be decimal unless a hexadecimal prefix (X') is used, as X'FF.

B= Buffer address. Default value is whatever partition System II assigns, as determined by adding 16 to the "W" parameter described below. The operator may set B to the address of any W+16 word buffer. However, care should be exercised, since no checks are made to prevent the destruction of the system or of application programs. When B is set, the current I/O partition (if any) is released.

C=0 Cylinder number. Range allowed: 0 to 408.

D= Drive number. Default value is determined from System II tables for the device specified in the command line. Range allowed: 0 to 7.

E=1 Error print interval. Setting E=0 suppresses all error messages. Setting E=n causes every n'th error message to be output.

F=0 Number of CE errors per command line which are to cause the listing of the whole actual and expected sector contents.

Note that F applies to each command line, not to each command line execution. Hence, for the following two command lines, only the first two CE errors cause the listing of the whole actual and expected sector contents, though the second command line is repeatedly re-executed until the CANCEL key is pressed:

F=2
TA *

H=0 Head number. Range allowed: 0 to 4.

G0=0 These are running totals of the retries which have
G1=0 occurred on drives 0 through 3. The operator can
G2=0 clear any of them to zero (and no other value).
G3=0

I= Maximum number of milliseconds for an I/O operation. The range allowed is 0 to 2047. (Twice the I number is allowed for operation G -- write with hardware verify.) Defaults: cartridge, 47ms; floppy, 173 ms.

J= Maximum number of milliseconds allowable for a seek operation, from 0 to 2047. Defaults: cartridge, 60ms; floppy, 173 ms.

K= Distance between successively accessed sectors during full track I/O, from 1 to 12. Defaults: cartridge, 9; floppy, 6.

L=1 Number of platters per drive to test, from 1 to 4.

N=1 Number of sectors, tracks or cylinders to test, from 1 to 65535.

P= Platter number, from 0 to 1. The default is determined from System II tables for the device specified in the command line.

Q=1 Number of drives to operate on, from 1 to 65535.

R=0 Number of retries to be attempted for an error.

S=0 Sector number, from 0 to 31.

T= Number of sectors per track, from 1 to 48. Defaults: cartridge: 12; single density floppy, 26; double density floppy, 15.

U= Number of heads per cylinder, from 1 to 8. Defaults: cartridge, 2; single sided floppy, 1; double sided floppy, 2.

V=12 Value code parameter, from 1 to 12.

Parameter	Value for 1st Word of Sector	Value for Subsequent Words
0	X'0000	X'0000
1	X'FFFF	X'FFFF
2	X'5555	X'5555
3	X'3333	X'3333
4	X'F0F0	X'F0F0
5	Sector address. Value for preceding word + X'1111.	

Structure of the sector address for parameter 5 (or 7):
Bit 0 = high bit of drive #. Bit 1 = bit of drive # ORed with platter #. Bit 2 = low bit of drive # ORed with high bit of cyl #. Bits 3-10 = low 8 bits of cyl #. Bit 11 = head #. Bits 12-15 = sector number.

6	X'0000	Value for preceding word + 1.
7	Current random number + sector address as in 5.	Random number based on previous word.
8	X'5555	Complement of previous word -- either X'AAAA or X'5555.
9	X'0000	Complement of previous word -- either X'FFFF or X'0000.
10	X'0001	The value for the preceding word rotated one bit right.
11	X'FFFE	Same as for V=10.
12	The sector is divided into 8 equal sections (up to 7 words at the end of the sector are ignored) and "V=" codes are applied, in this order: 8, 0, 11, 10, 1, 8, 9 and 7.	

W=256 Number of words per sector, from 1 to 8048.

Y= Number of cylinders per platter, from 1 to 406.
Default from system tables; standard value is 203.

Z=8 Number of drives, from 1 to 8.

CAUTION: Extreme care must be exercised in setting parameters B, D, and P. The drive number may be 0, 1, 2 or 3. P may be 0 or 1. If any of these parameters are changed, nothing else should be run concurrently with J5UDD, and System II should be reloaded after J5UDD has been terminated.

Two-Valued Parameters

The following conditions are enabled by setting the parameter to 1, and disabled by setting the parameter to 0-- for example, DS=1 or DS=0. This list indicates the conditions, the parameter values, and the default settings.

DD	0	Not double density floppy.
	1	Double density floppy.
DS	0	Do not display every status-busy-done state. Default.
	1	Display every status-busy-done state.
HI	0	Check cartridge sector headers during each I/O operation. Default.
	1	Do not check cartridge sector headers.
OG	0	Normal cartridge read amplifier gain. Default.
	1	Low cartridge read amplifier gain.
OM	0	No cartridge offset minus. Default.
	1	Cartridge offset minus.
OP	0	No cartridge offset plus. Default.
	1	Cartridge offset plus.
RT	0	No restore when seeking.
	1	Restore when "S" command performed.
SU	0	No suspend during I/O. Default.
	1	Suspend during I/O.
WP	0	Write protect switch on drive is expected to be off. Default.
	1	Write protect switch on drive is expected to be on.

CAUTION: Only one of cartridge parameters OG, OM and OP should be 1 at any given time.

Tests to be Performed

A test to be performed is indicated by entering a pair of letters, followed by <RETURN> -- for example, WA <RETURN>. The first letter of the pair indicates the type of test:

- C Read and check read data.
- P Print contents.
- R Read.
- S Seek only.
- T Write, read and check read data.
- W Write.

The second letter of the pair indicates which portion of the disk is to be tested:

- A Each sector on each of L platters on each of Q drives beginning with track 0,0 on each drive and proceeding sequentially through the tracks. The platters are done sequentially beginning with platter D,P.
- C N cylinders on each of P platters on each of Q drives beginning with track C,0 on each platter and proceeding sequentially through the tracks. The platters are done sequentially beginning with platter D,P.
- R Each sector on each platter P and $P+L-1 \text{ MOD } 2$, on each drive $D \text{ MOD } Z$, $(D+1) \text{ MOD } Z$, ..., $(D+Q-1) \text{ MOD } Z$ in random order.
- S N sectors on each of L platters on each of Q drives beginning with sector C,H,S on each platter and proceeding sequentially through the sectors. The platters are done sequentially beginning with platter D,P.
- T N tracks on each of L platters on each of Q drives beginning with track C,H on each platter and proceeding sequentially through the tracks. The platters are done sequentially beginning with platter D,P.
- Z No seek - sector H,S on each of L platters on each of Q drives beginning with platter D,P.

Sector Address Messages

Format: A HI OG OM OP RX WX VX NX DX PX CX HX SX DD RT

A Operation Code:

N	NOP
R	Read
S	Seek
W	Write

HI Header inhibit; only for parameter HI=1.
OG Read amplifier gain low; only for parameter OG=1.
OM Offset minus; only for parameter OM=1.
OP Offset plus; only for parameter OP=1.
RX Actual retries, in decimal, if non-zero.
WX Words per sector, in decimal.
VX Value code, in decimal.
NX Number of sectors, tracks or cylinders to test.
DX Drive number, in decimal.
PX Platter number, in decimal.
CX Cylinder number, in decimal.
HX Head number, in decimal.
SX Sector number, in decimal.
DD Double density floppy.
RT Restore when seeking.

If the bottom line of the initiating terminal is not a sector address message, then the screen is rolled up one line and the sector address message is written to the bottom line. If the bottom line is a sector address message, the new message is written over the old one. Each time an I/O operation is initiated, a sector address message is output. Just before allowing operator input, a message with a blank operation code appears.

Whenever a status message (with or without error codes), a CE error message, or sector contents are output, the appropriate sector address is written to the listing file, if any. These are the only cases in which a sector address message is written to the listing file.

Status Messages

Format: W EEEE BX

W Timing indicator:

P Preceding the operation (before starting the device).

D During the operation (after starting the operation and before the interrupt or lost interrupt detected).

A After the operation (after the interrupt).

L Lost interrupt (after the lost interrupt detected).

U After an unexpected interrupt.

BX Device busy status bit.

EEEE Error status register; read with a DIA instruction.
Any sum of these hex values may appear:

0001 Some device on daisy chain is busy.

0002 Not used; the corresponding bit is always 0.

0004 Write protected.

0008 CRC error, data field or header.

0010 Deleted data mark was read (floppy only).

0020 Drive status is "not ready."

0040 Heads unloaded (floppy only).

0080 Program error; commands overlapped.

0100 Format error; sector illegal or overrun.

0200 Drive went "not ready" since last command (floppy only).

0400 Drive status is "seek error." Or, illegal cylinder number.

0800 Sector not found; illegal sector number, header CRC error, or sector length mismatch.

1000 DMA overrun error.

2000 Header sector length mismatch (floppy only).

4000 Double-sided floppy disk.

8000 Not used; the corresponding bit is always 0.

Status Error Messages

Status error messages have the same format as described above, but they begin with several asterisks, and end with one or more error codes.

The possible error codes are listed below along with the corresponding bit of the controller error register (where applicable) and the check which produces the error condition. Note the following definitions:

1. Preceding an operation -- after the device is cleared and before it is started.
2. During an operation -- after the device is started and before the interrupt or lost interrupt detection.
3. After an operation -- after the interrupt or lost interrupt detection.
4. Transfer operation -- scan for equality, scan high, scan low, read, verify, or write.
5. I/O operation -- NOP, read, seek, or write.

Code:	Meaning:	Check:
AV	Address verify error.	Bit 6 of the error register is non-zero after an I/O operation.
BF	Buffer error.	(1) A read with hardware verify or a write operation changes the first or last word of the I/O buffer; or (2) a read operation changes the word preceding or following the I/O buffer.
BN	Device busy status = 1.	The device busy status, bit 15, is on (1) preceding an I/O operation; (2) during a seek operation; or (3) after an I/O operation.
BZ	Device busy status = 0.	The device busy status, bit 15, is off.
CE	Compare error.	Data word read from the disk has an unexpected value.
CS	Checksum error.	Bit 12 is non-zero after an I/O operation.

CY	Cylinder address error.	Bit 5 is non-zero after an I/O operation.
! DC	Drive not ready.	Bit 6 is non-zero after an I/O operation.
!		
DF	Drive fault error.	Bit 10 is zero after an I/O operation.
FO	Format error.	Bit 7 is non-zero after an I/O operation.
LI	Lost interrupt.	The check for this condition is that the maximum time constraint results in the detection of a lost interrupt (see the section on interrupt handling).
OV	DMA overrun.	Bit 3 is non-zero after an I/O operation.
PG	Program error.	Bit 8 is non-zero after an I/O operation.
PR	File protect error.	Bit 13 differs from the input parameter WP after an I/O operation.
TM	Software detected timing error.	SU=0 and the interrupt does not occur within the maximum allowable time after starting the I/O operation.
UIx	Unexpected interrupts; "x" of them occurred.	One or more unexpected interrupts passed from System II to program's interrupt handler.

Interrupt Handling

SU=0 After starting the I/O operation, the J5UDD task waits for the interrupt in a loop, checking the busy device status flag, the done device status flag, the device error register, the device control register and the drive seek status register field. If the interrupt does not occur before the maximum allowable time, then the TM error flag is set. If the interrupt does not occur before four times the maximum allowable time, then the LI error flag is set and the I/O operation is terminated.

SU=1 After starting the I/O operation, the J5UDD task suspends for four times the maximum allowable time. If the interrupt occurs before the suspend times out, the interrupt handler (part of D3000) sets the suspend message location and the suspend takes its normal return. If the suspend takes its timeout return, the LI error flag is set and the operation is terminated.

Compare Error Message

If a data word read from the disk contains an unexpected value, an error message is this format appears:

--- CE WWWW XXXX YYYY

WWW is the word offset (in hex) from the beginning of the sector to the first unexpected value.

XXXX is the hex value actually read.

YYYY is the hex value expected.

When the CE error is caused by reading fewer words than expected, XXXX is shown as four blanks.

Other Messages

QUIT

J5UDD is terminating; control returns to System II. This is a normal status message.

** MORE THAN 4 STATUS STATES DURING I/O

The program detected more than four changes of status state during an I/O operation but only had space to save the first four. Execution of the last command line continues.

ERROR: DEVICE NOT FOUND.

The device to be tested as specified in the command line was not found in the system tables. Execution terminates.

ERROR: NOT A DISK

The device specified for testing is not a disk. Execution terminates.

ERROR: LISTING DEVICE = TESTED DEVICE

The J5UDD listing cannot be directed to the device under test. Execution terminates.

BAD INPUT!

An entry in the last command line was unacceptable. The rest of the line is ignored. J5UDD waits for the next command.

UNKNOWN ERROR CODE FROM GCHN

A system call of GCHN (Get Channel Number) returned an error code other than ERCHNA (Channel Not Available). Execution terminates.

HIGH-TO-LOW STACK OVERFLOW! (or) MESS. BUF. OV.

Internal failure in J5UDD. Execution terminated.

MEMTST -- Memory Diagnostic

MEMTST is an execution sharable program for testing memory. It runs in the detached mode -- once initiated, the terminal is released for the performance of other functions.

Initiation

To initiate execution, enter one of these two commands:

MEMTST [xxxx]

MEMTST www yyy

xxxx The size (in hexadecimal words) of the test partition. If omitted, MEMTST obtains 128 words.

www A memory range (in hexadecimal). If this range is
yyy specified, then the test partition will consist of all locations from www rounded up to a multiple of 16 to yyy, truncated to a multiple of 16.

Extreme caution should be exercised in specifying a memory range to be tested since MEMTST makes no checks to prevent destroying System II, another program or itself.

The MEMTST command may be entered a number of times to test more than one area of memory. Each execution of MEMTST will run continuously until terminated by entering this System II command:

KILL MEMTST

After termination, the following message will be displayed:

***** MEMTST ***** # OF ERRORS DETECTED nnnn

Here, "nnnn" is the number of errors detected, in hexadecimal.

Description

1. Test 1 - Indirect Address Test

This test stores the address of the next word in each word. This maximizes the number of memory references per unit time.

2. Test 2 - Double Word Ripple Test

One pass of this test sets every even word to the test value, and every odd word to the complement of the test value. After each pass, the test value is complemented. The number of memory and register bit transitions are maximized.

Error Messages

If a fatal error is detected, one of the following messages will be displayed and control will be returned to System II:

ERROR IN SPECIFICATION OF TEST AREA	The command line specification of the test buffer was in error.
HIGH-TO-LOW STACK OVERFLOW	In normal operation, this message is never seen. It probably indicates a bug in MEMTST.

If any other error is detected, the following message will be displayed and MEMTST will continue executing:

```
***** MEMTST ***** TEST n   www www xxxx yyyy
```

n The number of the test; 1 or 2.

www The hex address of the error.

xxxx The expected hex value.

yyyy The actual hex value.

SKPTST-- Skip Function Diagnostic

This diagnostic program tests skip instructions to determine if there is a memory or CPU problem related to the skip function.

Initiation

To begin execution, enter the command:

SKPTST [P=xxxx]

P=xxxx Optional. A hexadecimal number of words to override the default work area of X'60 words. If fewer than X'20 words are specified, the minimum test partition size X'20 is used.

Description

SKPTST executes an internal program called MAINPR continuously in the allotted work area. The command may be entered any number of times to test other areas of memory.

Termination

To terminate execution of SKPTST, enter this System II command:

KILL SKPTST

Error Messages

If a problem is detected in the CPU or memory, the following message will appear on the terminal screen, ten times for each memory location.

***SKPTST**REL ADD nnnn ACT ADD nnnn

REL ADD nnnn The relative address of a JSR instruction within MAINPR.

ACT ADD nnnn The current memory address within the work area. This address is updated dynamically as it changes.

The relative address may be used to locate the JSR instruction within MAINPR (listed below) which executed following the error.

The following statements comprise the program MAINPR:

```
          .BSECT
0000 0000      B1  .WORD      0
0001 FFFF      B2  .WORD     -1
0002 0001      B3  .WORD      1
          .TSECT
0000 4C00 A     L3:  LI        0,0
0001 4FFF A     LI        3,-1
0002 1101 A     BZR        .+2
0003 2A00 A     JSR        0(2)
0004 F301 B     SKNE       0,B1+1(3)
0005 E302 B     SKG        0,B2+1(3)
0006 2A00 A     JSR        0(2)
0007 1501 A     BNZ        .+2
0008 7302 B     SKAZ       0,B2+1(3)
0009 2A00 A     JSR        0(2)
000A 4801 A     AISZ       0,1
000B 1501 A     BNZ        .+2
000C 2A00 A     JSR        0(2)
000D E303 B     SKG        0,B3+1(3)
000E F301 B     SKNE       0,B1+1(3)
000F 2A00 A     JSR        0(2)
0010 1102 A     BZR        .+3
0011 7303 B     SKAZ       0,B3+1(3)
0012 48FF A     AISZ       0,-1
0013 2A00 A     JSR        0(2)
0014 0200 A     RTS
          0000      .END
```

SPY -- Register Monitor

SPY displays the Videocomputer's hardware Program Counter and CPU registers, updating this information at each clock interrupt. The following format is used, starting in column 40 of line 24 on the invoking terminal:

PC=nnnn R0=nnnn R1=nnnn R2=nnnn R3=nnnn

Initiation

To begin execution of SPY, enter this System II command:

SPY

After SPY is initiated, it detaches from the terminal. It is then possible to enter command lines on the terminal, as long as the command does not extend into the area used by SPY.

Termination

To terminate execution of SPY, enter this System II command at any terminal --

KILL device:SPY

-- in which "device" is the disk from which SPY was loaded.

Error Message

SPY ALREADY RUNNING Only one SPY can be in the system.

COMMENT SHEET



FROM

Name _____

Business Address _____

Does this publication meet your requirements? yes no

If not, please explain. _____

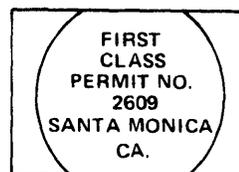
Do you wish a reply? yes no

COMMENTS

Describe any errors and/or suggested changes. Please include page number.

CUT ALONG DOTTED LINE

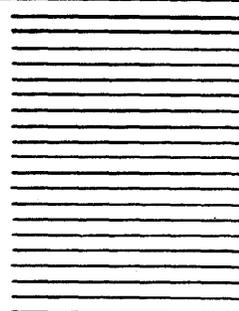
(FOLD)



BUSINESS REPLY MAIL

NO POSTAGE STAMP NECESSARY IF MAILED IN UNITED STATES

POSTAGE WILL BE PAID BY:



 Jacquard Systems
3340 Ocean Park Boulevard
Santa Monica, California 90405

Customer Service Department

(FOLD)

CUT ALONG DOTTED LINE

**3340 Ocean Park Boulevard
Santa Monica, California 90405**



**3340 Ocean Park Boulevard
Santa Monica, California 90405**

