SUPER DIAGNOSTIC SYSTEM

(SUDSY II)

VOLUME I

LINC-8

DIGITAL EQUIPMENT CORPORATION • MAYNARD, MASSACHUSETTS

DEC-L8-DSS0-D

SUPER DIAGNOSTIC SYSTEM (SUDSY II) VOLUME I OPERATOR'S MANUAL

DIGITAL EQUIPMENT CORPORATION • MAYNARD, MASSACHUSETTS

First Edition September 1967
Second Edition April 1968
Third Edition August 1968

ACKNOWLEDGEMENT

This programming system represents the combined efforts of many people both at Digital Equipment Corporation and in the field. We feel that it is a substantial addition to the LINC-8 program library and suggest that all users take full advantage of it.

Particular note is made of J. Martin Graetz, who did most of the programming and virtually all writing of this manual.

Finally, a note of thanks for the patience of the many people involved in LINC-8 Production. The many hours spent using the evolving versions of this program were invaluable in producing this final working system.

LINC-8 Engineering

SUDSY II

CONTENTS

VOLUME I

CONTENTS (continued)

CONTENTS (continued)

TABLES

SUMMARY OF OPERATING PROCEDURES

LOADING (The Monitor must be in memory)

Put 0100 in RIGHT SWITCHES
Press PDP-8 LOAD ADD

From LINCtape

RS 1	down
RS 2	up
RS 6/11	(test number)

From Punched Tape

RS 0	up	ASR reader
	down	high-speed reader
RS 1	up	
Other	RS have no effect	

Press PDP-8 START

After a previous normal halt (C(PC)=0114)

From LINCtape: set RS as above and press PDP-8 CONT.

From Punched Tape (except after Tests 7, 22, 23, 24, and 25): Place new tape in reader, set RS, and press PDP-8 CONT. Tests noted above destroy RIM and BIN Loaders.

STARTING

Loaded from LINCtape

Set LEFT SWITCHES
(see below)
Press PDP-8 CONT

Loaded from Punched Tape

Set LEFT SWITCHES
Put 1000 in RIGHT SWITCHES
Press PDP-8 LOAD ADD, then START

SWITCH SETTINGS (All switches up for effect, down for no effect)

Program Control

LS 11	Repeat this cycle
LS 10	Repeat entire test*
LS 9	Special repeat*

Error Processing

LS 0	Don't ring the bell
LS 1	Don't print data
LS 2	Don't halt on error
LS 3	Don't print INTS

*See test write-up for special use of LS 10 and LS 9; these may be used to repeat sections of a test, rather than the entire test.

PROGRAMMED HALTS AND RECOVERY

Errors

C(PC)=0257
C(AC)=Error Ident.No.

Recovery

Press PDP-8 CONT

Normal Final Halt

C(PC)=0114

Recovery

See STARTING, above

See test write-up for special programmed stops.

LOADING THE MONITOR

From LINCtape

Lift LOAD
Put 0700 in LEFT SWITCHES
Put 0013 in RIGHT SWITCHES
Lift DO
Put 4030 in RIGHT SWITCHES
Press PDP-8 LOAD ADD, then
START

From Punched Tape

Put 7777 in RIGHT SWITCHES
Press PDP-8 LOAD ADD
Put RS 0 up for ASR, down for
high-speed reader
Press PDP-8 START

From GUIDE

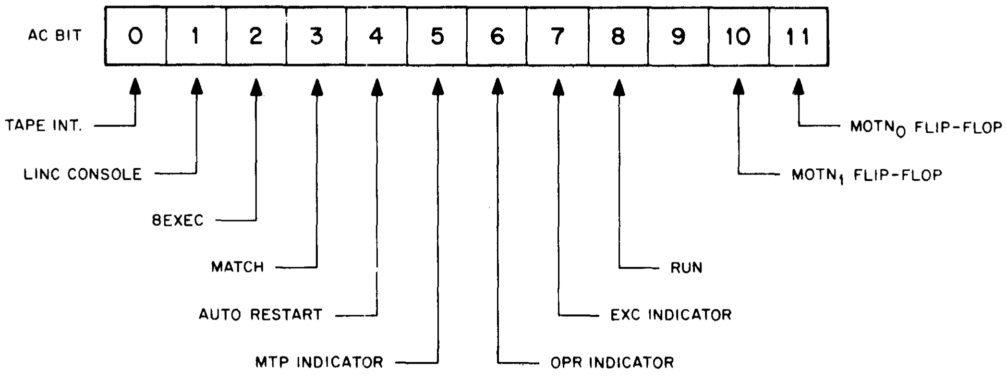
"EXECUTE THE PROGRAM
DECTST"

LOADING AND RUNNING A CONTINUOUS TEST

Load the Monitor (see above)
Put 0100 in RIGHT SWITCHES
Press PDP-8 LOAD ADD
Set LEFT SWITCHES (see above)
Set LS 6 or 8 for number of additional memory fields attached.
Set RS 3 up for SNS test and grand final halt (in epilog)
down to skip SNS test and recycle through test sequence
Press PDP-8 START.

LINC INTERRUPT STATUS BIT ASSIGNMENT

PDP-8 AC bit = 1 when condition exists



CHAPTER 1
INTRODUCTION

The LINC-8 Super Diagnostic System (SUDSY) exhaustively tests the functioning of the LINC processor subsystem of the LINC-8 computer. SUDSY consists of a series of PDP-8 programs, each testing one or more parts of the LINC processsor logic, the LINC-8 interface, the LINC processor operating mode and interrupt functions, and the LINC console.

A typical test program governs the actions of one or more LINC instructions. Using the LINC-8 interface IOTs, the program sets up all pertinent LINC registers; it then passes control to the LINC to execute a short LINC program sequence which includes the instruction being tested. At the end of this sequence, control returns to the PDP-8 program, which examines the pertinent registers for errors. The test continues until the data are exhausted or until certain control parameters reach terminating values. Various program control and error processing options are available to the operator through settings of the LEFT and RIGHT SWITCHES.

1.1 PHYSICAL ORGANIZATION

The entire set of SUDSY programs, along with a Monitor which performs housekeeping, error processing, and tape reading functions, is kept on the LINC-8 maintenance tape. With the Monitor in memory, each test may be called either individually by the operator (useful for checkout and for hunting specific errors) or in automatic sequence by the Monitor (useful for daily maintenance), passing oncethrough the entire series. The entire test sequence may also be cycled indefinitely (useful for acceptance testing).

In addition to the maintenance tape, each test of SUDSY is also available as a binary program on punched tape. This allows its use even when the LINCtape is not functioning; checkout engineers, for example, can begin to debug parts of the LINC processor before the LINCtapes have been installed.

1.2 LOGICAL ORGANIZATION

The tests are mutually independent, but each one operates with a Monitor program which remains in memory at all times. The Monitor provides the necessary facilities for processing errors, communicating with the operator, handling LINCtape, and controlling the operation of the LINC processor. The Monitor also provides a number of useful housekeeping subroutines for clearing memory, testing and clearing certain I/O flags, and setting memory bank limits.

A section of page 0 is reserved for common storage of error data, indexing parameters, pointers, and other variables which must be shared by the Monitor and the test program.

1.3 STORAGE ALLOCATION

SUDSY occupies the first quarter of LINC-8 memory (registers 0000-1777); this section is not accessible to the LINC processor. The Monitor, which includes the LINCtape handling routines, occupies registers 0100-0777. Common storage and workspace extends from 0000-0077. Each test begins in register 1000 and extends toward 1777.

During operation, the system does not encroach on any area of memory available to the LINC processor; therefore, any arrangement of the memory bank selectors is acceptable to SUDSY II, as long as it is in the same field with the LINC memory banks. (Except for the Extended Memory tests themselves, SUDSY is not able to operate across memory fields, i.e., with the test system in one field and the LINC memory banks in another.)

CHAPTER 2
THE SUDSY TEST PROGRAMS

The executable portion of each test extends from register 1000 toward 1577. In general, registers 1600-1777 (page 7) are reserved for error messages and other printed matter, though space limitations sometimes require error data to be stored elsewhere. Executable routines are never kept on page 7.

Test	Functions Tested	Description
Prolog (Test 0)	LINC-8 interface, LINC console, LINC active registers.	This test is the first to be executed. It has four parts, the first two requiring operator intervention. Part 1 tests the LINC Left Switches and the active registers, using the IOT instructions to read the switches and display them. Part 2 tests the LINC console switches in the same manner. Part 3 tests the Auto Restart Delay. Part 4 is an exhaustive test of the interface IOT instructions and their corresponding active registers.
1	"Go-Control": Interrupts, 8 Exec Class, HLT, STC, and memory address.	Tests the operation of the ICON instruction (6141) in clearing the LINC interrupt status and MOTN bits, and in selecting, starting, and deselecting the LINC processor. Test 0 checks the program interrupt from the LINC on execution of HLT (interrupt disabled), and EXC, OPR, and MTP (interrupt enabled). The STC instruction is tested for accurate storage of data, and is then used to test the LINC lower memory bank addressing.
2	MSC Class	Tests NOP, CLR, COM, ZTA, ATR, and RTA
3	JMP, JMP 0	Tests the operation of these instructions for selected settings of P and for every state of S.
4	Skip Class	Tests ZZZ, APO, AZE. Other skips are tested later.
5	ADD (FLO)	Tests ADD, first for selected operands, then for random numbers. A special repeat option allows one to remain in the random number test alone. The FLO FF and the FLO skip instruction are tested.

Test	Functions Tested	Description
6	Rotate Class	Tests ROR, ROL, and SCR, each with selected values and with random numbers in the LINC accumulator. LZE and the Z-register are tested. Numbers are "vibrated" in the A-register, by executing a string of ROL-ROR pairs in rapid succession.
7	LDA. Full memory address test.	Tests LDA for correct transfer of data. Exhaustively tests the addressing of LDA, using all combinations of i and β, thus providing a rigorous address test of both memory banks.
10	1. STA 2. SAE 3. Logic	Addressing of these instructions is not exhaustively tested, but is chosen to provide worst-case conditions. Section 3 tests BCO, BCL, and BSE for selected combinations of bits.
11	ADA	This is nearly identical to Test 4 (ADD); it also checks for accurate addressing. The operation of FLO and its flip-flop is also tested.
12	ADM	Similar to Test 11, this also checks for accurate storing of the sum.
13	LAM	Similar to the preceding tests, this also checks for correct addition and setting of the Link.
14	SET, XSK	Tests for correct transfer of data (SET) and correct indexing and skip (XSK), for worst-case addressing conditions.
15	SRO	This instruction is tested for all β-registers, using worst-case addressing conditions.
16	MUL	Logically, this test is similar to the four addition tests, in that a table of operands is used, once each as multiplicand and multiplier, against all numbers as the second operand. The arithmetic is simulated, then the operation is tested. Two addressing schemes are used, both providing worst-case conditions: MUL i 0, and MUL i 16.

Test	Functions Tested	Description
17	LDH	In addition to checking for proper manipulation of data, this program also does another complete memory address test, similar to Test 7 (LDA).
20	STH, SHD	These tests are essentially similar to their full-word counterparts, STA and SAE. Worst-case addressing is used, and both halves of the storage word are tested.
21	DIS, DSC	Both display channels are used, with worst-case addressing, to display two horizontal lines (using DIS) with the legend "TEST 21:DISPLAY!" (using DSC) between them.
22	Memory Bank Selectors	The Memory Bank Selectors and their associated LINC instructions, LMB and UMB, are tested for setting all possible values from all other possible values. Boundary protection is tested for bank 0 and all banks above the upper limit of memory, including extensions.
23	Extended Memory Addressing	Address indexing around the end of each memory bank is tested using LDH i 17. A full memory address test using LDA 0 is performed for all settings of the selectors.
24	Extended Memory Timing	Using all possible values of C(A) and C(L), the instruction ROR i 17 is executed to test the critical timing at the end of this instruction, when the next instruction is fetched.
25	EXT S01	The special case of SET and DSC, where a second lower memory bank reference is required after normal i-β address setup, is tested using critical addresses.
Epilog (Test 50)	KST, SNS Z and L.	KST is tested for skip and no-skip conditions, using IACF to set the KST FF. SNS requires operator intervention to actuate the SENSE SWITCHES whose states are continuously displayed in the RELAY lights. During a continuous run, the SNS test may be omitted. The Z- and L-registers are checked for all possible C(Z) and

Test	Functions Tested	Description
		both states of L, to see that they are not disturbed by any instruction which is not supposed to involve them.

In the SUDSY II system, there are 23 tests, numbered in octal from 0 through 25, plus the Epilog (Test 50). In addition, there are five tests (numbered 26-32) comprising the LINCtape Diagnostic Program (see separate manual). Ample space is provided on the Maintenance Tape for additions to SUDSY II.

CHAPTER 3
THE MONITOR

The Monitor performs all necessary operating housework: it processes the switch options, performs all input and output functions, and controls test program sequencing. The Monitor contains subroutines to clear a specified section of memory, set memory bank pointers, test flags, and test and clear the LINC interrupts and control flip-flops.

The Monitor occupies all of the first four pages (registers 0020-0777) of memory, except for registers 0000-0017, which are reserved for the use of the test programs. Registers 0020-0077 contain common storage, the Monitor subroutine dispatch table, and several constants shared by the Monitor and the test programs.

The functions of the programs in the Monitor are described below:

Program — Function

TED — The Test Director
This is the main control program. TED examines the Right Switches and calls each new test into memory, using the LINCtape routines or the BIN Loader as required. Before starting the test, TED initializes all pertinent registers and pointers and clears the LINC interrupts. When the test is finished, control returns to TED.

CLIFF — Clear LINC interrupts, flip-flops, and flags.
This subroutine is called by TED, before each test starts, and by SPIN, after the LINC program sequence has been executed in each test cycle, so that spurious interrupts are not left hanging.

SLIME — Set LINC memory bank pointers.
This subroutine, called by TED, sets two pointers, LMB and UMB, which are used by the test programs to determine the absolute octal addresses of locations in the two LINC memory banks.

KEFT — Keyboard Flag Test.
Immediately before a LINC program sequence is executed and again at the end of each test cycle, the keyboard flag is sensed and, if it is up, a signal register (KSIG) is set. The signal is used by CYCRO (see below) to determine whether or not the test cycle should be repeated. The KBD flag is cleared.

Program — Function

LINGO

LINC Go-Control

This program sets the P-register to the starting location of the LINC program sequence, selects the LINC, and starts the LINC processor. On return from the LINC, LINGO tests the KBD flag; if it was up during LINC operation, the AC is set to 7777, and control returns to the test program. This signal causes the entire test cycle to be ignored. If the flag was down, the AC contains 0 on return.

CYCRO

Cycle Repeat Option

This subroutine tests LS 11 at the end of a test cycle. If the switch is up, the cycle is repeated; if not, the test continues. If the KBD flag is raised, however, the repeat is suppressed until the next occurrence of an error.

TESRO

Test Repeat Option

When a test is completed, TESRO examines LS 10; if the switch is up, the entire test (or major section thereof; see each test write-up for details) is repeated. If the switch is down, control returns immediately to TED.

CLEM

Clear Memory

This subroutine clears the section of memory specified by the two arguments of the 'CLEAR' macro, which calls CLEM. The first argument is the address of the first register to be cleared; the second specifies the length, in registers, of the section of memory affected.

EROS

Error Option Selector

EROS examines Left Switches 0, 1, and 2 to determine which of the three error processing options are to be executed; if a given switch is up, the operation it controls is not performed. LS 0 controls the ringing of the bell, LS 1 the error printout, and LS 2 the error halt.

BOOK

Error Printout Control

This routine is used by EROS and SPIN (see below) to print error information, including all diagnostic messages, test identification headings, and error data.

SPIN

Spurious Interrupt Test

After the LINC sequence is executed, SPIN examines the LINC interrupt status; if any bit is set, a diagnostic, including the state of the interrupts as read into the AC, is printed. The interrupt status is cleared. The printout is suppressed if LS 3 is raised. The test does not stop after the diagnostic print.

Program	Function
Output Package	A set of five subroutines controls all printout. These are: OTY Basic character-printing subroutine. CARL Types a carriage return and a line feed. TAB Spaces to the next tab stop. PRIM Prints all messages and headings. POC Octal print routine. Types error data; suppresses leading zeros with spaces.
LOLITA	Load LINCtape This program examines the Right Switches, reading in the specified test (or reading tests in sequence) from LINCtape. Each test is loaded, relocated into its operating area, and if running continuously, begun. The tape reading routines are taken almost intact from PROGOFOP.

All subroutines of the Monitor are called via a dispatch table, stored on page 0, which contains the addresses of every subroutine.

The functions of all option switches are described in Chapter 5. The Monitor is described in detail in Volume 2, Chapter 3.

CHAPTER 4
DOCUMENTATION

For ease in reference, complete operating instructions are provided in tabular form at the front of this manual.

Following Chapter 5 of this manual is a set of write-ups, each consisting of a description of one SUDSY test, and including a tabular presentation of all test data: special switch settings, printouts, error stops, and LINC program sequences.

4.1 WRITE-UP FORMAT

Each test write-up is organized according to the following outline:

1. Description
2. Special Switch Settings
3. Printout Headings
4. Non-Error Messages
5. Special Halts
6. LINC Program Sequences
7. Execution Time
8. Error Stops

4.1.1 Notes On The Write-Up Format

4.1.1.1 Description - Is a short explanation of how the test program works. Whenever required, this includes instructions for operator intervention, as, for example, in the Prolog (Test 0).

4.1.1.2 Special Switch Settings - Are any which differ from, or are in addition to, those given in the Operating Procedures.

4.1.1.3 Printout Headings - Are shown exactly as they appear on the teleprinter when an error occurs. Each heading is surrounded by a box to set it off from the rest of the matter. An explanation of the column heads is given below the heading.

4.1.1.4 Non-Error Messages - Are shown exactly as they appear on the teleprinter.

4.1.1.5 Special Halts - Are any which are not included in the normal error-detection and final halt routines.

4.1.1.6 LINC Program Sequences - Are given with their LINC memory bank octal locations; to find the absolute addresses, add the appropriate constant, according to the setting of the Memory Bank Selectors. For the standard settings, the Lower Bank will be set at 2; the constant is thus 4000. Locations of all LINC-accessed data are also given.

4.1.1.7 Execution Time - Is for one complete pass through the test program. Where a number of repetitions are written into the program (as, for example, Test 21), the execution time includes the repetitions.

4.1.1.8 Error Stops - Are presented in tabular form. The first column shows each error message as it appears when printed. The second column gives the error identification number as it appears in the PDP-8 AC at the time of the halt. The third column lists the pertinent column heads associated with that error. Column four gives the program tag as it appears in the argument of the 'ERROR' macro in the program listing. Below each error listing, a description of probable causes and test criteria is given.

4.1.2 Organization of the Write-Up

The description begins on page 1 of the write-up, and continues after the tabular listings, when necessary. Items 2 through 7 appear in tabular form on page 2 of each write-up. Item 8 appears on page 3 of the write-up (continuing to page 4 when necessary), so that all pertinent test information is presented on facing pages.

4.2 SYMBOL CONVENTIONS

Throughout this manual, the following symbol conventions apply:

Symbol	Definition
R	Any register of the computer
C(R)	The contents of register R.
$C(R_j)$	The contents of bit j of register R.
$C(R_j\text{-}k)$	The contents of bits j through k, inclusive, of register R.
$\overline{C(R)}$	The complement of the contents of register R.
ICON-n	Indicates the execution of an ICON instruction (PDP-8 IOT code 6141) with <u>n</u> in the AC.

Register Designators:

AC	The PDP-8 accumulator
PC	The PDP-8 program counter
MA	The PDP-8 memory address register
MB	The PDP-8 memory buffer

Register Designators (continued)	Definition
A, B, S, P, Z	Respectively, the LINC accumulator, memory buffer, address register, program counter, and Z-register.
L	The LINC Link
LS-n	Left Switch n
RS-n	Right Switch n
LMBS, UMBS	Lower Memory Bank Selectors, Upper Memory Bank Selectors
CSW1	Console switches 1 (see drawing LINC8-0-L7)
CSW2	Console switches 2 (see drawing LINC8-0-L7)
Lnnnn	A memory register whose LINC address is nnnn. For example, register 20 in LINC lower memory is designated L20 (or L0020). The highest location of LINC upper memory is L3777. The absolute octal address, of course, depends upon the setting of the Memory Bank Selectors.

CHAPTER 5
GENERAL OPERATING PROCEDURES

This chapter describes the procedures for loading the SUDSY II system, loading and running a single test, and running a continuous test. All switch options are described. A summary of SUDSY II operations will be found in appendix 1. For the operator's convenience, a summary of operating procedures is presented in tabular form at the front of this manual.

5.1 LOADING THE MONITOR

5.1.1 From the LINC-8 Maintenance Tape

Mount the LINC-8 Maintenance Tape on unit 0.

Press LOAD

Set the Left Switches to 0700.

Set the Right Switches to 0013.

Press DO.

Set the Right Switches to 4030.

Press PDP-8 LOAD ADD, then START. The Monitor is now ready to operate.

When loading from LINCtape, pressing LOAD brings in PROGOFOP. The next steps read the contents of block 13 into quarter 0 of LINC memory. Block 13 contains a PDP-8 program which, when executed, reads the Monitor into its operating area, pages 0-3 of PDP-8 memory.

5.1.2 From Punched Tape

The BIN and RIM Loaders must be in memory.

Put the Monitor binary tape in the reader, and turn the reader on.

Set the Right Switches to 7777.

Press PDP-8 LOAD ADD.

If reading from the ASR 33 or ASR 35, press PDP-8 START.

If reading from a high-speed reader (type 750 or PC01), put RS 0 down, then press PDP-8 START.

5.1.3 From GUIDE

Call for the program "DECTST".

5.2 RUNNING AN INDIVIDUAL TEST

5.2.1 Loaded from LINCtape

If the test was loaded from LINCtape, or if a previous test has reached normal SUDSY II halt,
Set Left Switches for program control and error options.
Press PDP-8 CONT

5.2.2 Loaded from Punched Tape

If the test was loaded from punched tape,
Set the Left Switches for desired options.
Set the Right Switches to 1000.
Press PDP-8 LOAD ADD, then START.

5.3 RUNNING A CONTINUOUS TEST

5.3.1 Run Through Once, Stop After Epilog

To run through the entire SUDSY II test sequence once, stopping at the end of Epilog:
Load the Monitor.
Set the Right Switches to 0100.
Press PDP-8 LOAD ADD.
Set the Left Switches as desired for options.
Put RS 3 up, RS 1 and RS 2 down.
Press PDP-8 START.

The test sequence will run up to the last section of Epilog, which is the SNS test and requires operator intervention. At this point, striking any key on the ASR keyboard (except those which do not generate flag pulses) will cause the program to stop with the word END displayed in the LINC lights. Then, pressing PDP-8 CONT will cause the entire sequence to begin again.

5.3.2 Continuous Cycle

To cycle continuously through the test sequence without stopping;
Load the Monitor.
Set the Right Switches to 0100.
Set the Left Switches for desired options.
Press PDP-8 LOAD ADD, then START.

5.4 LOADING AN INDIVIDUAL TEST

Set the Right Switches to 0100 (the Monitor must be in memory).

Press PDP-8 LOAD ADD.

5.4.1 From LINCtape

Set the Right Switches as follows:

RS 1 down

RS 2 up

RS 6-11: Octal number of the test desired.

Press PDP-8 START

5.4.2 From Punched Tape

Set the Right Switches as follows:

RS 0 down for high-speed reader (Type 750 or PC01)
up for ASR 33 or ASR 35.

RS 1 up

Other switches have no effect.

Press PDP-8 START

RS 1 controls the source of the input; down for LINCtape and up for punched tape. RS 0 is used by the standard BIN Loader to determine its input source. RS 2 tells the LINCtape loading routines to look for the test corresponding to the number in Right Switches 6-11.

5.5 PROGRAM CONTROL OPTIONS

In normal operation, an individual test proceeds to the end and then returns to TED for the normal halt or to read in the next test. The operator may alter the program flow by means of the LEFT SWITCHES, as shown in table 5-1.

TABLE 5-1 PROGRAM CONTROL OPTIONS

Switch	Position	Function
LS 11	down	Normal operation. Proceed to the next test cycle.
	up	Repeat the last cycle executed. In general, this means that as long as the switch is up the test cycle is repeated with the same data. (See section 5.6)
LS 10	down	Normal operation. When the test is finished, return control to TED.
	up	Repeat the entire test from the beginning. Note that for certain tests, this switch is effective only for part of the program; see each write-up for details.

TABLE 5-1 PROGRAM CONTROL OPTIONS (continued)

Switch	Position	Function
LS 9	down	Normal operation.
	up	For certain tests, this switch is used as an additional test repeat control for sections of the program. It functions the same way as LS 10, except that it does not return control to TED.

5.6 CYCLE REPEATS AND SCOPING

It is often useful to be able to execute a short program loop indefinitely, in order to synchronize an oscilloscope with one or more pulses or levels in the computer. SUDSY provides the cycle repeat option, controlled by LS 11, for this purpose. No repeat occurs until the first error is detected; thereafter, the test repeats the cycle causing the error as long as LS 11 is up, regardless of how often the error occurs. This allows the operator to detect intermittent as well as gross failures of the logic.

Because many errors are data-dependent, a provision has been made to allow the operator to proceed from one error-causing cycle to another without manipulating LS 11. If, while a cycle is being repeated (LS 11 up), the operator strikes any signal-generating key on the ASR keyboard, the test program resumes normal operation, ignoringthe repeat switch until another error occurs, when the repeat option again takes effect. In this way, the operator can step from error to error.

5.7 CYCLE-IGNORE FUNCTION

If an external device flag is raised when the LINC processor is running, the current LINC instruction is completed and control returns to the PDP-8, with the LINC RUN flip-flop still on. Because it is possible that a key may be struck during LINC-mode operation, a provision has been made to ignore the cycle in which this has occurred, therefore, spurious errors resulting from the incomplete execution of a LINC program sequence will not be processed.

5.8 ERROR PROCESSING OPTIONS

When an error occurs, the operator can be informed in three ways. First, the bell rings; next, an error printout consisting of diagnostic messages and error data is typed on the Teleprinter; finally, the program halts with the error identification number in the AC lights. Each of these error processing options is controlled by one of the Left Switches, as shown in table 5-2.

TABLE 5-2 ERROR PROCESSING OPTIONS

Switch	Position	Function
LS 0	down	Ring the bell when an error occurs.
	up	Don't ring the bell.
LS 1	down	Print error messages, identification, and data.
	up	Don't print anything.
LS 2	down	Halt, with error number displayed. C(PC)=0257.
	up	Don't stop.
LS 3	down	Process interrupt error and print diagnostic.
	up	Ignore LINC interrupt errors.

The following are examples of how the error option switches can be used:

a. When scoping on a repeated cycle, the operator would normally suppress all three processing functions (all switches up).

b. When collecting data to make informed guesses about the nature of an error, the operator can suppress the bell and the halt, so that the program will print error data without interruption.

5.9 THE ERROR PRINTOUT

Printout of error messages and data is in standard tabular form, described below. Examples of all texts and messages are given in the write-up for each test.

5.9.1 The Error Heading

Error data is printed in several columns across the page. These columns are identified by headings printed when the first error occurs in the test being executed. The heading consists of a title line identifying the test or test section and short column heads identifying the data. The heading is printed only once during the execution of a given test or test section; however, if the test is restarted at memory location 1000, the heading will be printed again.

Certain tests (Test 10 is a good example) contain several logically distinct parts; for each subsection, then, a separate heading is provided. In each case, the heading is printed only once for that subsection, but if the test is repeated, the headings will appear again.

5.9.2 Error Data

For most errors, the pertinent data is printed in two rows, with a short message at the left margin to indicate the particular instruction or function being tested. The two rows themselves are identified by the letters A (actual) and C (correct) respectively, referring to the first and second rows of data. The actual data result from the operation of the LINC instruction; the correct data are determined in advance by the test program.

5.9.3 Types of Data

The information printed usually compares erroneous data with that provided for proper evaluation of the results. Data in error are identifiable by the fact that both actual and correct values are printed. Other data appear as single entries on the second (correct) line of print. Columns with no entries refer to items which are not tested or are not pertinent and are not in error.

5.9.4 Other Error Messages

In addition to the data print-out, some tests provide special messages to call the operator's attention to a particular item in the subsequent print, or to inform him of troubles not associated with the data. Test 16, MUL, has two examples of this type of message. One states simply, FLO FF SET, noting the fact that the overflow flip-flop, which should not have been disturbed by the action of MUL, was changed. The other, BETA ERROR, warns the operator to check the C(BETA) column on the forthcoming printout. The test program does not stop after such a message is printed, but continues to the next data printout.*

5.9.5 Spurious Interrupt Diagnostic

During each test, a running check is kept on the status of the LINC interrupts. After the LINC program sequence is executed, the SPIN subroutine examines the interrupts; if any INT status bit is set, the diagnostic message

INTS

is printed, followed by an octal number which represents the contents of the AC after the LINC Interrupt status has been read into it. If LS3 is up, the message is suppressed. The interrupt error diagnostic is not affected by the settings of LS0, 1 or 2.

5.10 THE SUDSY PHILOSOPHY

The Super Diagnostic System was designed to make use of the peculiar two-processor structure of the LINC-8. By using one processor, the PDP-8, to check out the other, no prior assumptions need be made about the proper working condition of any part of the LINC.

*If printout is suppressed (LS1 up), the program does halt, with the error identification number displayed in the PDP-8 AC. The message is not printed.

Because SUDSY is designed for use both as a checkout tool and as a running maintenance diagnostic routine, certain compromises were made in order to make it useful for the two purposes. The system is built around a set of subroutines, the Monitor, which control all operations; each test uses these subroutines liberally. In addition, the tests themselves are often divided into subroutines, so that no test exceeds the memory space allowed for the system. For checkout engineers, this sacrifice of clear straight-line programming is compensated for by a detailed diagnostic printout which provides all the necessary checkout information for most problems. The following suggestions are offered for instances when a single-step approach is useful.

a. Don't try to follow the program listing, instruction by instruction, unless you are a very competent programmer. Macros are liberally used and there is a lot of jumping back and forth between subroutines.

b. If you want to single-step through a test cycle, do the following:

1. Start the test (RS 1000, press PDP-8 LOAD ADD, then START) and wait until the first error halt.

2. To go through the cycle again, raise LS 11.

3. Now single-step. Ignore what happens until the number 0012 appears in the PDP-8 accumulator. This is the signal that the LINC processor is about to start operation.

4. From this point, watch the LINC processor console. The LINC program may be a single instruction or a short sequence, but in every case (except parts of the Go-Control test) the sequence ends with a HLT, and control returns to the PDP-8.

5. If you suspect a LINC interrupt failure, continue single-stepping until the PDP-8 PC contains the address 354. At this point the PDP-8 AC will contain the LINC interrupt status.

6. After this point, control returns to the test for error processing. The best procedure is to return to continuous operation until the next halt. Suppressing the printout will speed this up, unless the error data are of interest.

c. Often the cycle-repeat option (scope mode) operates over too long a program sequence to allow reliable scope synchronization. If this happens while testing a LINC instruction that does not destroy the data (examples of instructions that do are LAM, MUL, STC, LDH, STH), the engineer can make a much tighter scoping loop that does not go through the error testing program. A special register in the LINC Go-Control subroutine (LINGO) is provided for this purpose.

1. Examine register 0170. It should contain a NOP (7000). Replace this with a JMP 157 (5157). This creates a tight loop that does no more than setthe LINC P-register, select and start the LINC processor. As long as no LINC instructions or data are destroyed, this will provide a very tight scoping loop.
2. Be sure to restore the NOP to register 0170 when you are finished.

NOTE: If the working data are not relevant to what you are testing for, any LINC instruction sequence can be cycled through in the above manner.

WARNING

Don't try a tight loop with DIS; you are likely to burn out a phosphor.

TEST 0 - PROLOG

Tests LINC console, switches, and flip-flops; also AUTO RESTART delay, and PDP-8/LINC interface IOTs.

NOTE: To execute tests A and B, manually start this test in location 1001. These tests require the participation of the operator. See description below for directions.

1. DESCRIPTION

1.1 Test 0A: LINC Left Switches Test

The MEMORY BANK SELECTORS are read into the AC, using IMBS (PDP-8 IOT instruction 6155). If the selectors are set to the standard values, i.e., if C(AC)=0142, the test proceeds. The contents ofthe Left Switches are read into the AC, using ILES (6145). From there, using the appropriate IOTs, the C(AC) are transferred to the P-, S-, A-, and B-registers. A delay of about 160 msec follows. then the switches are sensed and displayed again.

1.1.1 Operator - To test the functioning of the Left Switches and of the PDP-8 IOT instructions ILES, ISSP, IACS, IACA, and IACB, actuate the switches singly and in groups. Observe the following effects:

a. When a switch is up, the corresponding bit in each of the registers (A, B, S, P, and AC) should be lit. When the switch is lowered, these lights should go out. There is a barely detectable delay before the light changes state after the switch is moved. This allows an intermittent error to appear as a flashing or dimly flickering light.

b. If the lights do not funrtion as described, the trouble can come from one of the following sources:

1. If, when a switch is up, none of the corresponding lights are lit, or if they all remain lit when the switch is down, the trouble is probably in the switch itself, its associated hardware, or in the ILES instruction.

2. If a bit failure occurs only in the A-, S-, or P-registers, and in none of the others (except, possibly, the B-register), the fault is probably in the affected register or in the corresponding interface IOT (IACA, IACS, ISSP).

3. If a bit failure appears only in the B-register, the trouble lies in the operation of IACB (6161).

4. If a bit failure appears in all the LINC registers, but not in the PDP-8 AC, the fault is in the B-register itself, the IACB instruction, or both.

Description continued on page T0-4

2. SPECIAL SWITCH SETTINGS

None

3. PRINTOUT HEADING (Test D only)

```
PROLOG: INTERFACE IOT TEST
        CONTENTS OF
        A, B,
        Z, OR
        OLD P   NEW P
```

In the first column, the contents of the tested register are printed.

OLD P Previous C(P) as read from B after ISSP.

NEW P C(P) transferred to P by ISSP. Printed for information only; tested on following cycle.

4. NON-ERROR MESSAGES

Test C, when Monitor is freshly loaded (no previous tests executed):

```
SET DELAY KNOBS: COARSE ON 4, FINE AT FULL RIGHT.
PRESS PDP-8 CONT.
```

5. SPECIAL HALTS

Test C, after message: C(AC)=0000
C(PC) = 1212

6. LINC PROGRAM SEQUENCES

None

7. EXECUTION TIME

Test C: 4-8 sec.

Test D: 2 sec.

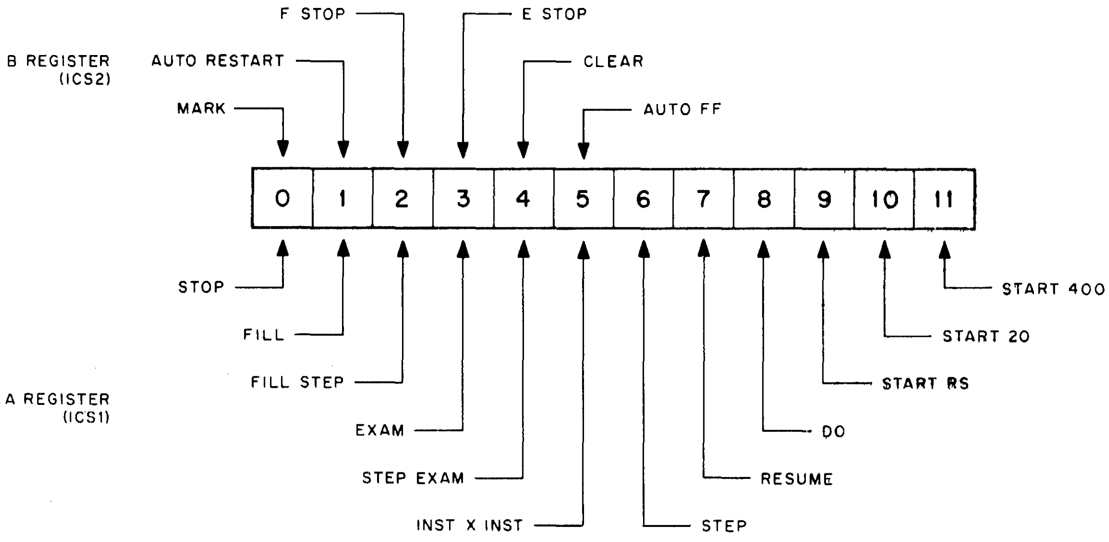


Figure T0-1 LINC Console Switch Bit Assignments (Test B)

8. ERROR STOPS AND INFORMATION

Error Message	C(AC)	Data Printed	Program Tag
None	MBS status	None	PEM
		If the Memory Bank Selectors are not set to the standard values (C(AC)=0142), the program stops with the actual status displayed in the AC. Pressing CONT causes the test to begin again; it will not proceed until the selector status is correct.	
FAST DELAY	0020	None	E20
		The AUTO INT FF was set before 458 msec had elapsed since the DELAY START trigger was fired. Check to be sure that both delay knobs are at their full right positions. If they are, the delay is firing too soon.	
NOISY RESTART	0021	None	E21
		After the AUTO INT FF was set, noise generated by the changing states of bits in the A-, B-, S-, and P-registers caused DELAY START to fire again. This is the type of error which causes more than one LINC instruction to be executed when operating in INST X INST mode.	
SLOW DELAY	0022	None	E22
		The AUTO INT FF was still not set 7.5 seconds after the DELAY START trigger was fired. Check for mis-wiring or bad enabling gates. The AUTO FF is checked during Test B; try running the Console Switch test again.	
B REG	0010	B	E10
A REG	0011	A	E11
		The number transferred to the indicated register using IACB or IACA did not match that returned to the AC using IBAC or IAAC. Check the register in question and both the associated IOTs.	
P REG	0012	OLD P, NEW P	E12
		Using ISSP, the P-register was set from the AC. The previous C(P), read into the AC from B, is compared with the correct value; if the two do not match, the data are printed under "OLD P". The new C(P) is also printed; this value is tested on the following cycle.	
Z REG	0013	Z	E13
		a. ICON-14 may have failed to clear Z. b. ICON-15 may have failed to transfer C(B) to Z correctly. c. IZSA may have failed to jam-transfer C(Z) to A correctly. d. A Z-register flip-flop may be defective.	

Description continued from page T0-1

To continue to Test B, strike any signal generating key on the keyboard (see NOTE at the end of this section).

1.2 Test 0B: LINC Console Switch Test

The LINC console switch states are read into the PDP-8 AC using the two IOTs, ICS1 and ICS2. The first set of switches is displayed in the A-register, the second set in the B-register (see figure T0-1 for bit assignments).

When any of the switches labeled AUTO RESTART, INST X INST, E STOP, F STOP, or CLEAR is actuated, the corresponding Flip-Flop light, immediately to the right of the switch, is turned on using the IOT instruction IACF. The MARK FF is not set, so that information on a mounted tape will not risk being destroyed. The MARK switch light (B_0) operates normally.

With one exception, the light corresponding to a switch is on until the switch is actuated, at which time it goes out. As in Test A, a 160 msec delay between changes of state allows an intermittent error to become apparent. The single exception is the AUTO FF bit (B_5), which goes on when the AUTO RESTART switch is raised; at the same time, the AUTO RESTART switch light (B_1) goes out, and the AUTO RESTART flip-flop light goes on.

1.2.1 Operator - Actuate the console switches one by one, examining the corresponding light in the A- or B-register for an indication of correct operation.

WARNING

DO NOT ACTUATE THE LOAD SWITCH

To continue to Test C, press any signal-generating key on the keyboard.

1.3 Test 0C: DELAY RESTART Test

1.3.1 Operator - Action is required only when the Monitor has been freshly loaded into memory. At that time, the following message is printed:

```
SET DELAY KNOBS: COARSE ON 4, FINE AT FULL RIGHT.
PRESS PDP-8 CONT
```

The program halts. Set the delay knobs as directed, and press CONT.

On subsequent passes through the Prolog, this message and halt do not occur.

When the test resumes, the AUTO FF is set, the LINC interrupts are cleared, and the DELAY RESTART is triggered. The program then waits for 458.5 msec. If the AUTO INT FF is set within that time, the diagnostic message

FAST DELAY

is printed. If the flip-flop is set during the 0.5 to 7.5 sec interval after the delay is triggered, it is considered a normal delay. If the flip-flop is not set by the end of the 7.5 sec interval, the diagnostic message

SLOW DELAY

is printed.

As soon as the AUTO INT FF is set, the program tests the noise rejection ability of the delay trigger. The flip-flop is cleared, and every bit of the LINC-, A-, B-, S-, and P-registers changes state. If the AUTO INT FF is set as a result of noise generated by the bit-changing pulses, the message

NOISY RESTART

is printed.

The test is performed eight times; if the delay is operating normally and the knobs are set correctly, the test should take between 4 and 8 sec.

The cycle repeat option (LS 11) causes all of Test C to be repeated.

1.4 Test 0D: PDP-8/LINC Interface Test

Each LINC active register (except S) and its associated IOT instruction are tested for accurate transmission of data to and from the PDP-8 AC. All possible values of C(AC) are used for each register. The registers are tested in the following order:

B-Register	(IACB, IBAC)
A-Register	(IACA, IAAC)
P-Register	(ISSP)
Z-Register	(ICON-14, ICON-15, IZSA)

1.5 General

Any of the IOT instructions tested can be the focus of a tight loop of no more than three or four PDP-8 instructions, to allow scope synchronizing. Scattered throughout the test program (see listing in Volume 2) are NOP instructions which can be replaced by JMPs when a scope loop is desired. The locations are identified by the instruction mnemonic 'Q'; the associated comment gives a suggested JMP to use in making the loop.

NOTE: All keys on the ASR 33 keyboard generate keyboard flag signals except CTRL, SHIFT, REPT, and BREAK.

TEST 1 - GO-CONTROL

Tests HLT, 8EXEC class, STC, and lower memory bank addressing.

1. DESCRIPTION

Test 1 has two logically distinct parts. The first part tests the LINC processor operating mode (Go-Control); the second tests the STC instruction, then uses that to perform an address test of the LINC lower memory bank.

1.1 Test 1A: Go-Control

When the test begins, the LINC interrupts are cleared, using the PDP-8 IOT, ICON-7. The interrupt status is then read into the AC, using INTS, and tested to see if the LINC interrupts were in fact cleared by ICON-7. (The 8EXEC indicator bits are masked out for this test.) If the interrupts are not clear, the error message,

LINC INTS NOT CLEAR

is printed, followed by an octal number which represents the C(AC) after the LINC interrupt status has been read into it.

Next, the Go-Control is tested with interrupt disabled. The LINC is selected (ICON-10) and started (ICON-12). A HLT instruction in register L1 is executed and control returns directly to the PDP-8. The C(P) are tested; if the LINC processor did start, the occurrence of a GNI pulse will have incremented P. The HLT test is performed 4096 times.

After the HLT test, the LINC Deselect is tested. First, the LINC flip-flops are set, using IACF with 7773 in the AC (the MARK FF is not set). Next, the LINC is deselected (ICON-11). The LINC AUTO FF is then tested (ICS2 to read the status) to see if the PWR CLR pulse was generated at the time of the deselect. If the bit is still set, the message,

PWR CLR FAILED

is printed. If the bit is not set, an attempt is made to restart the LINC with the processor deselected. The test is made by examining the C(P); if the P-register was indexed (indicating that a GNI pulse occurred), the program assumes that the LINC processor started, and the error message,

DESELECT FAILED

is printed. The Deselect test is also performed 4096 times.

Finally, the LINC Go-Control is tested with the interrupt enabled. Each of the three 8EXEC class instructions, EXC, OPR, and MTP, is tested to see that each one causes an interrupt request, followed by a program break to register 0000. The indicator corresponding to each instruction is tested to see that it is set correctly. Each 8EXEC class instruction is tested 4096 times.

Description continued on page T1-5

2. SPECIAL SWITCH SETTINGS

LS 10	up	repeat STC and Memory Address tests only
LS 9	up	repeat Go-Control test only

3. PRINTOUT HEADINGS

3.1 Go-Control Test

```
GO-CONTROL TEST
     C(P)     INDIC       INTS
```

C(P)	Contents of P
INDIC	State of the 8EXEC indicators and 8EXEC INT flip-flop.
INTS	State of LINC interrupts

3.2 STC and Memory Address Tests

```
STC AND LOWER MEMORY ADDRESS TEST
     C(Y)    Y            C(A)
```

C(Y)	Contents of Y
Y	Address of operand
C(A)	Contents of A after execution of STC

4. NON-ERROR MESSAGES

None

5. SPECIAL HALTS

None

6. LINC PROGRAM SEQUENCES

6.1 Go-Control Test

```
L1,     *                              /*In succession:  HLT, EXC, OPR, MTP
L2,     HLT
```

6.2 STC and Address Tests

```
L1 or L1775,          STC Y
L2 or L1776,          HLT
```

7. EXECUTION TIME

4 sec

8. ERROR STOPS AND INFORMATION

Error Message	C(AC)	Data Printed	Program Tag
INTS nnnn	No halt	INTS status	None
	A spurious LINC interrupt occurred. The octal number nnnn represents the interrupt status as read into the PDP-8 AC. See table at the front of this manual to determine the cause of the interrupt. The program does not stop.		
LINC INTS NOT CLEAR	0101	INTS status	E101
	An attempt to clear the LINC interrupt status failed. See table at the front of this manual to determine which interrupts are causing trouble.		
HLT	0110	C(P)	E110
	a. If the actual C(P)=0001, the LINC processor probably did not start. b. If the execution of HLT fails to stop the LINC and cause an interrupt request, the LINC RUN light will remain on, while the string of HLTs cycles endlessly.		
PWR CLR FAILED	No halt (0111)	None	E111
	After the LINC deselect, the PWR CLR pulse should have cleared the LINC flip-flops. If not, the AUTO FF will remain on; it was preset before the test. No data is printed and the test does not stop.		
DESELECT FAILED	No halt (0112)	None	E112
	An attempt to restart the LINC processor after DESELECT was successful. The error is detected by testing for an indexed P-register, which indicates that a GNI pulse has occurred.		

(continued)

Error Message	C(AC)	Data Printed	Program Tag
EXC OPR MTP	0120 0121 0122	C(P), INDIC, INTS	EGC
	Each of these 8EXEC class instructions is tested with the interrupt enabled. a. The RUN flip-flop was cleared (it should remain set after the break). The INTS status is printed. b. More than one instruction was executed. This indicates that a second GNI occurred; C(P) are printed. c. Wrong 8EXEC indicator bit set, or 8EXEC class status bit AC_2) not set. INDIC printed.		
TRAP FAILED	No halt (0123)	See preceding error	E123
	The processor failed to trap to PDP-8 location 0001 when an 8EXEC class instruction was executed.		
STC	0130	C(Y), C(A)	E130
	Data was stored incorrectly in register L0003, or C(A) was not cleared, or both.		
ADRS	0140	C(Y), Y, C(A)	E140
	The register designated by Y contained the wrong information, or the wrong register was addressed, or both.		

Description continued from page T1-1

1.2 Test 1B: STC and Memory Address Test

The STC instruction is tested for its ability to store all possible numbers in a given location. The instruction is placed in register L1, followed by HLT. The instruction stores data in L3. The A-register is tested to see if it is cleared by STC.

The Memory Address test determines whether STC can store a given datum in every register of LINC lower memory. The instruction is executed first from L1, storing information in registers L3 through L1777. Next, STC is executed from L1775, storing data in L0 through L1774.

In each cycle of the test, the number stored in a register is equal to the absolute octal address of that register. For standard settings of the selectors, the absolute address of any LINC register is 4000+LINC address. Thus, the number 4003 is stored in L0003, 5003 is stored in L1003, 5774 in L1774, and so on.

The two sections of Test 1 are repeatable separately. Switch 9 causes the Go-Control test to be repeated; if this switch is down, the test proceeds to Test 1B. Switch 10 controls the repetition of the Memory Address test, including the STC 3 test.

TEST 2 - MSC CLASS

Tests NOP, CLR, COM, ZTA, ATR and RTA.

1. DESCRIPTION

1.1 NOP Test

A string of 15 NOP instructions, terminated by HLT, is stored in locations L1-L20. To determine that NOP has no effect on the A-register, the string is executed once for every possible number placed in A. The program counter is also tested for possible anomalies; the HLT is in L20, therefore, C(P) after execution should equal 0021.

1.2 CLR Test

The instruction is tested once for all possible numbers in A. Each time, the Z-register is set to 7777; after execution, both the A- and Z-registers should be clear.

1.3 COM Test

This test has two parts.

First, COM is tested once for every possible number in the A-register.

Second, a string of 63 COM instructions, terminated by HLT, is stored in registers L1-L100. The number 5252 is placed in A, and the string is executed. Because there are an odd number of COMs in the string, the resulting C(A) should be the complement of the original value, or 2525. The string is executed only once, but the cycle repeat option will causes the String test to be repeated.

1.4 ZTA Test

The Z-register is loaded consecutively with all possible even numbers, and ZTA is tested for each. The C(Z) should be placed in the A-register, shifted right one place. Since the low-order bit is lost each time, only even numbers need be tested.

1.5 ATR-RTA Test

The A-register is loaded with one of a sequence of numbers selected to turn on the relays in succession, starting with the rightmost relay. Each relay is added until all are lit, then, starting again at the right, each relay is turned off.

The LINC program sequence is ATR, CLR, RTA. The number returned to the A-register should match that placed there at the start of the test. Each number is used 2000 times, so that the operator can watch the relay lights to observe the progress of the test. The impression, when the instructions operate correctly, is that of a train of lights traveling slowly in from the right and out at the left.

2. SPECIAL SWITCH SETTINGS

None

3. PRINTOUT HEADING

```
TEST 2:  NOP, CLR, COM, ZTA, ATR/RTA.
         C(A)       C(P)       C(Z)
```

(CA)	Contents of LINC accumulator
C(P)	Contents of LINC program counter
C(Z)	Contents of LINC Z-register

4. NON-ERROR MESSAGES

None

5. SPECIAL HALTS

None

6. LINC PROGRAM SEQUENCES

6.1 NOP Test

```
L1-L17,          NOP
L20,             HLT
```

6.2 CLR and COM Tests

```
L1,  CLR         COM
L2,  HLT         HLT
```

6.3 COM String Test

```
L1-L77,          COM
L100,            HLT
```

6.4 ZTA Test

```
L1,  ZTA
L2,  HLT
```

6.5 Relays Test

```
L1,  ATR
L2,  CLR
L3,  RTA
L4,  HLT
```

7. EXECUTION TIME

7.5 sec

8. ERROR STOPS AND INFORMATION

Error Message	C(AC)	Data Printed	Program Tag
INTS nnnn	No halt	INTS status	None
	A spurious LINC interrupt occurred. The octal number nnnn represents the interrupt status as read into the PDP-8 AC. See table at the front of this manual to determine the cause of the interrupt. The program does not stop.		
NOP	0200	C(A)	E200
	The contents of A were disturbed during the execution of a string of 15 NOPs.		
P	0201	C(P)	E201
	The LINC processor stopped in the wrong place while executing the NOP string. Correct C(P)=0021.		
CLR	0210	C(A), C(Z)	E210
	CLR failed to clear the A-register or the Z-register, or both.		
COM	0220	C(A)	E220
	C(A) was not complemented correctly.		
STRNG	0221	C(A)	E221
	C(A) not correct after a string of 63 COMs was executed. The starting value of C(A) was 5252; the resulting C(A) should be 2525.		
ZTA	0230	C(A), C(Z)	E230
	C(Z) was not transferred correctly to the A-register. Check for operation of the jam-transfer from Z to A, and for correct ASHR PLS at T3, after the transfer.		
RELAY	0240	C(A)	E240
	Either ATR or RTA, or both, failed to transfer data correctly, so that the contents of the relays as read into A do not match the original C(A). The failure can be identified by examining the relay lights. If the lights match the correct C(A), then RTA failed. If C(relays) are not correct, then ATR failed, or one of the relays is defective.		

TEST 3 - JMP AND JMP 0

1. DESCRIPTION

The JMP test is in two parts.

1.1 JMP X, Where X ≠ 0000

During this part of the test, the JMP is executed to every location in memory between L0001 and L1775, inclusive, from each of a selected set of origins. The target location, to which control is transferred, contains NOP; the next location contains HLT. All other registers in memory are clear. The origin registers are shown in section 1.1.1 below.

To prevent infinite loops resulting from a JMP situation, the program forces the target to leapfrog over the JMP, so that for any given origin, the JMP is not tested for the register immediately preceding the JMP or, obviously, for the JMP location itself. Because there is a two instruction sequence (NOP followed by HLT) at the target, the JMP cannot be tested for a target in L1777, since register L0 is involved.

1.1.1 Origin Locations For JMP - Each of the addresses is a LINC location, not an absolute octal address.

0001
0002
0003
0004
0006
0007
0010
0016
0017
0020
0036
0037
0040
0076
0077
0100
0176
0177
0200
0376
0377
0400
0776
0777
1000
1776
1777

After the JMP is executed, the P-register is tested for correct addressing, L0 is checked to see that the return address was stored properly (as a JMP to the location following the origin), and the A-register is examined to see if it was disturbed during JMP.

1.2 JMP 0 Test

The instruction 'JMP 0' is executed from every location. As in the preceding test, the target location (L0) contains NOP, the following location HLT. L0 is tested to see that it was not changed, P is tested for proper addressing, and A is tested for possible disturbance.

Description continued on page T3-4

2. SPECIAL SWITCH SETTINGS

None

3. PRINTOUT HEADING

```
TEST 3:  JMP,  JMP 0.
    C(P)     C(L0)    ORIGIN  C(A)
```

C(P)	Contents of P after the jump
C(L0)	Contents of L0 after the jump
ORIGIN	Location of the JMP instruction
C(A)	Contents of A

4. NON-ERROR MESSAGES

None

5. SPECIAL HALTS

None

6. LINC PROGRAM SEQUENCES

6.1 JMP Test

```
Target register,     NOP
Target+1,            HLT
```

JMP is stored in each of the selected registers listed in section 1.1.1, and is executed to every register from L1 through L1776, inclusive.

6.2 JMP 0 Test

For origin L1:

```
L0,      JMP 10
L1,      JMP 0
L10,     NOP
L11,     HLT
```

For all other origins:

```
L0,      NOP
L1,      HLT
```

7. EXECUTION TIME

8 sec

8. ERROR STOPS AND INFORMATION

Error Message	C(AC)	Data Printed	Program Tag
INTS nnnn	No halt	INTS status	None
	A spurious LINC interrupt occurred. The octal number nnnn represents the interrupt status as read into the PDP-8 AC. See table at the front of this manual to determine the cause of the interrupt. The program does not stop.		
JMP	0300	C(P), C(L0), ORIGIN, C(A)	E300
	A JMP to a location other than L0 caused one or more of the following errors: a. The target address was incorrect. If JMP failed to transfer control to the correct register, or if NOP in the target location was not executed properly, C(P) will be incorrect. b. The return address was incorrect. If, on execution of JMP, the return JMP was not stored correctly, C(L0) will be in error. L0 should contain a JMP to the register immediately following ORIGIN, which is printed for comparison. c. The LINC accumulator was disturbed during the test. The correct C(A)= 2634; the A-register is set before the test.		
JMP 0	0310	All entries	E310
	JMP to L0 caused one or both of errors (a) and (c), described above. In addition, C(L0) might have been disturbed; for obvious reasons, JMP 0 should not change C(L0). Because NOP is stored in L0 for this test, C(L0) should always be 0016, except for the special case of an origin of L1. For this case, C(L0) should be 6010.		

Description continued from page T3-1

The special case of JMP 0 from location L1 is treated as follows:

JMP 0 is stored in L1.
JMP 10 is stored in L0.
NOP is stored in L10, and HLT in L11.

The final contents of L0 should be 6001, after executing the JMP to register L10. The final C(P) should be 0012 after executing the HLT in register L11.

TEST 4 - SKIP CLASS

Tests ZZZ, APO, AZE

1. DESCRIPTION

Each of these instructions is tested using a worst-case configuration for the P-register. The skip instruction is in L1775, followed by two HLTs. This is designed to catch timing troubles whenever a skip actually occurs, because the skip is effected by $p + 1 \longrightarrow P$ at t_3 (GNI), followed immediately by another one at t_L (GO). In forcing the longest possible carry chain at t_L (from 1777 to 0000), a severe test of the timing is accomplished.

1.1 ZZZ, ZZZ I

For these instructions, the Z-register is loaded consecutively with all possible numbers; the instruction is executed, and the results tested. For ZZZ, the skip should occur whenever C(Z) is even; never when C(Z) is odd. The converse is true for ZZZ I.

1.2 APO, APO I

Again, all possible numbers are tested in sequence. The LINC accumulator is loaded, the instruction is executed, and the results are tested. For APO, the skip should occur as long as C(A) is positive, that is, while $C(A_0)=0$. While $C(A_0)=1$, there should be no skip. Conversely for APO I.

1.3 AZE, AZE I

For each of these instructions, two tests are performed. First, C(A) is alternated between 0000 and 7777, and the result tested. For each of these values, AZE should skip, AZE I should not. The alternation is performed and tested 4096 times. Next, the A-register is loaded with all values in succession from 7776 through 0001; for these values, AZE should not skip, AZE I should.

2. SPECIAL SWITCH SETTINGS

None

3. PRINTOUT HEADING

```
TEST 4:  ZZZ, APO, AZE.
    C(A)      C(P)       C(Z)
```

C(A)	Contents of LINC accumulator
C(P)	Contents of LINC program counter
C(Z)	Contents of the Z-register

4. NON-ERROR MESSAGES

None

5. SPECIAL HALTS

None

6. LINC PROGRAM SEQUENCES

```
L1775,              (ZZZ, APO, or AZE)
L1776,              HLT
L1777,              HLT
```

7. EXECUTION TIME

7 sec

8. ERROR STOPS AND INFORMATION

Error Message	C(AC)	Data Printed	Program Tag
INTS nnnn	No halt	INTS status	None
	A spurious LINC interrupt occurred. The octal number nnnn represents the interrupt status as read into the PDP-8 AC. See table at the front of this manual to determine the cause of the interrupt. The program does not stop.		
ZZZ	0400	C(A), C(P), C(Z)	ZER
ZZZ I	0401	C(A), C(P), C(Z)	ZER
APO	0410	C(A), C(P)	E410
APO I	0411	C(A), C(P)	E411
AZE	0420	C(A), C(P)	E420
AZE I	0421	C(A), C(P)	E421
	Each of these errors is caused by a failure of the instruction being tested, or of the register bits sensed. In every case, if a skip is expected, the 'C' entry under C(P) will be 1; if a skip is not expected, the entry will be 0. C(A) are tested for all cases; if disturbed, the data are printed under C(A). For ZZZ and ZZZ I tests, C(Z) are tested. If any instruction fails to skip when it should, the 'A' entry under C(P) will be 0; if a skip occurred at the wrong time, the entry will be 1. If either HLT following the skip instruction is not executed (as for instance, if an interrupt request is waiting at the time the LINC processor is started), C(P) will be either 1776 or 1777.		

TEST 5 - ADD AND FLO

1. DESCRIPTION

The ADD instruction is tested using, first, a selected set of operands (see section 1.1.1), then a sequence of pseudo-random numbers. Each number is used as the A operand, with all possible numbers added to it, then as the Y operand, adding it to all possible numbers in A. Before the LINC sequence is executed, the PDP-8 simulates the 1's complement addition and the action of FLO. The operands are placed in Y and in A, and the LINC program is executed. Depending upon the action of FLO, the LINC program will halt with C(P)=4 or 5.

The Random Number test uses operands generated separately by the following PDP-8 program sequence:

```
TAD CV          /Previous random operand
CLL RAL
SZL
TAD (3
DCA CV          /Store new operand
```

A similar sequence generates a new operand in CV+1, for use as the Y OP.

The A operand is seeded with the number 5326; the Y operand with 2203. The algorithm implemented by the program sequence described above is as follows:

$I = 2\ (RN_0)$

$RN_1 = I + 3$ if $L = 1$; I if $L = 0$.

L:	Link Bit
RN_0:	Previous Random Number
I:	Intermediate Number
RN_1:	New Random Number

1.1.1 Add Test Operands

0000	4000	5252	7700
7777	3777	2525	0077

2. SPECIAL SWITCH SETTINGS

LS 9	up	Continue the Random Number Test

3. PRINTOUT HEADING

```
TEST 5:  ADD Y, FLO.
     A OP    Y OP    SUM    FLO    C(P)
```

A OP	Operand in A before execution
Y OP	Operand in Y
SUM	Sum left in A after execution
FLO	State of the FLO flip-flop after execution
C(P)	Contents of P after execution

4. NON-ERROR MESSAGES

None

5. SPECIAL HALTS

None

6. LINC PROGRAM SEQUENCE

```
L1,         ADD 50               /Add C(Y) to C(A)
L2,         FLO                  /Is FLO FF set?
L3,         HLT                  /No—stop here; C(P) = 0004.
L4,         HLT                  /Yes—stop here; C(P) =0005.

L50,        Y operand
```

7. EXECUTION TIME

19 sec

8. ERROR STOPS AND INFORMATION

<table>
<tr><th>Error Message</th><th>C(AC)</th><th>Data Printed</th><th>Program Tag</th></tr>
<tr><td>INTS nnnn</td><td>No halt</td><td>INTS status</td><td>None</td></tr>
<tr><td></td><td colspan="3">A spurious LINC interrupt occurred. The octal number nnnn represents the interrupt status as read into the PDP-8 AC. See table at the front of this manual to determine the cause of the interrupt. The program does not stop.</td></tr>
<tr><td>FLO ERROR</td><td>No halt
(0510)</td><td>None</td><td>E510</td></tr>
<tr><td></td><td colspan="3">If the state of the FLO flip-flop is not correct, or if the FLO instruction skipped improperly, this message is printed to call the operator's attention to the FLO and C(P) columns of the data printout, which follows immediately.</td></tr>
<tr><td>ADD</td><td>0500</td><td>All entries</td><td>E500</td></tr>
<tr><td></td><td colspan="3">If the SUM is incorrect, or if the operand in Y is disturbed, the corresponding data are printed. In addition, the original A OP is printed for information.
If a FLO ERROR has occurred, the FLO and C(P) columns will contain the following entries:
a. If the FLO FF was not set at the proper time, or if the FLO instruction failed to skip when the FLO FF was set, the entries are
FLO C(P)
(A) 0 4
(C) 1 5
b. If the FLO FF was set incorrectly, or if the FLO instruction skipped erroneously, the entries are
FLO C(P)
(A) 1 5
(C) 0 4</td></tr>
</table>

TEST 7 - LDA

Tests full memory address.

NOTE: Because this test uses all of LINC memory, the BIN and RIM Loaders are destroyed. They must be read back into memory before the next test can be read from punched tape. The LINCtape routines are not affected.

1. DESCRIPTION

The program tests the action of LDA in loading the LINC accumulator accurately. It is also a rigorous test of LINC memory addressing, using all possible combinations of I and β, for all memory locations. The test falls logically into four parts.

1.1 Test 7A: LDA I 0

This addressing combination causes the contents of the location immediately following the LDA instruction to be placed in the accumulator. The LDA is located at L1771; the operand in L1772. The instruction is tested for loading all possible numbers into the A-register. The result in A is tested, and the contents of memory are tested to see that they were not changed.

For the following test, the instruction is stored in L1771, and the effective address in L1772. All other LINC memory registers are set to contain their own addresses. Thus, for the standard memory bank settings, L0000 contains 4000, L0100 contains 4100, etc. All memory registers from L0000 through L1770 and L2000 through L3777, are tested.

1.2 Test 7B: LDA 0

With this addressing combination, the location immediately following the LDA contains the effective address of the operation. For each memory location, C(A) and C(Y) are tested for accurate transfer. In addition, the contents of L1772 are examined for accurate addressing. An indirect test for correct BETA reference is performed if C(A) = 0; this assumes that the wrong β-register was referenced; each β not being used contains 0, referencing register 0, which also contains 0.

For the remaining tests, LINC memory is set up as before, except that the LINC program is kept in registers L20-123. All β-registers not being tested contain 0000. All memory registers from L0030 through L37777 are tested.

1.3 Test 7C: LDA β. $1 \leq \beta \leq 17$.

This test is the same as the preceding one except that each β-register is tested in turn for indirect addressing. Those not being used contain 0.

Description continued on page T7-4

2. SPECIAL SWITCH SETTINGS

None

3. PRINTOUT HEADING

```
TEST 7:  LDA.  FULL MEMORY ADDRESS TEST.
     C(A)    BETA    C(BETA)    C(Y)
```

C(A)	Contents of A
BETA	Octal address of β-register being tested
C(BETA)	Contents of β-register. Same as the LINC address of Y, except during LDA I 0 test, when C(BETA) and C(Y) are the same.
C(Y)	Contents of the addressed register; should be identical to C(A).

4. NON-ERROR MESSAGES

None

5. SPECIAL HALTS

None

6. LINC PROGRAM SEQUENCES

6.1 Tests 7A and 7B

```
              Test 7A       Test 7B

L1771,        LDA I 0       LDA 0
L1772,        operand       Y
L1773,        HLT           HLT
```

6.2 Tests 7C and 7D

```
L20,          (LDA β or LDA I β)
L21,          HLT
L22,          HLT
```

7. EXECUTION TIME

14 sec

8. ERROR STOPS AND INFORMATION

Error Message	C(AC)	Data Printed	Program Tag
INTS nnnn	No halt	INTS status	None
	A spurious LINC interrupt occurred. The octal number nnnn represents the interrupt status as read into the PDP-8 AC. See table at the front of this manual to determine the cause of the interrupt. The program does not stop.		
LDA I	0700	All entries	E700
LDA	0701	All entries	E701
	a. C(A) is incorrect. This could result from a failure in the B to A transfer, or from accessing the wrong Y-register. b. C(Y) was disturbed. c. C(BETA) was incorrect. This is more likely to occur during the LDA I β test, and will probably appear as an indexing failure (NDXB).		
WRONG BETA?	No halt (0710)	None	E710
	If C(A)=0000 after execution, this message is printed to call the operator's attention to the possibility that the wrong β-register has been accessed.		

Description continued from page T7-1

1.4 Test 7D: LDA I β. $1 \leq \beta \leq 17$

In this configuration, the contents of the β-register are incremented by 1 before being used as the effective address. Indexing is done only over 10 bits, however; for example, when C(BETA)= 1777, the result of the indexing is 0000. Similarly, when C(BETA)= 3777, the result of indexing is 2000. The program makes a special test for the case of the 10-bit index; register L0 is preset with its own octal address so that the test will be correct.

TEST 10 - STA, SAE, BCO, BSE, BCL

1. DESCRIPTION

This test examines five of the full-index class instructions: STA, SAE, and the three logic operations, BCO, BSE, and BCL.

1.1 Test 10A: STA

This instruction is tested for correct transfer of data from A to memory, using the addressing scheme, STA 0. The instruction is located in L20, so that L21 contains the effective address, which is 0000. Thus the operand is always stored in L0. In addition, this insures that the instruction immediately following the STA is equivalent to HLT, so that if the P-register fails to index properly in order to skip L21, the LINC will stop with C(P)=0022. Since C(P) is tested, the indexing failure will be caught, and C(P) value printed. STA is tested for the ability to store every possible number from the accumulator, for leaving C(A) unchanged, and for correct addressing. C(Y) is preset to 7777 before each cycle.

1.2 Test 10B: SAE

This instruction also uses L21 as the β-register, so that the P-register is required to work as hard as possible during the execution of the instruction. Both skip and non-skip situations are tested.

First, the skip situation, C(A)=C(Y), is tested for all possible numbers in both A and memory. If the skip fails, the skip indicators in the printout column labeled "SKIP?" will be 0 and 1, respectively, for the A and C entries.

Next, the non-skip situation is tested in two ways, as follows:

a. The number N is compared with N+1, first with N in the A-register, then with the operands switched, so that N is in memory. All values of N are tested.

b. C(A) is made to differ from C(Y) by a single bit. This is done by the "sliding 1s and 0s" method. A value of all 0s is compared with a number which contains all 0s execept one bit, which is 1. The comparison is performed with the differing bit in each position, and for each position with the operands in both A and Y. The complementary case, with all 1s compared against all 1s except a single bit=0, is tested the same way.

In every case, the SAE instruction should not skip. If it does, the entries in the "SKIP?" column will be 1 and 0, when the error data is printed.

1.3 Test 10C: Logic Operations

The three instructions, BCO, BSE, and BCL, are tested in turn; each uses the same addressing scheme as 7A and 7B: effective address of 0000 in L21, and Y operand in L0. For each instruction, the A and Y operands are the same, 4631 and 6314, respectively.

Description continued on page T10-4

2. SPECIAL SWITCH SETTINGS

None

3. PRINTOUT HEADINGS

3.1 Test 10A: STA

```
TEST 10A:  STA 0.
    C(Y)    C(A)    C(P)
```

C(Y)	Contents of memory register
C(A)	Contents of A
C(P)	Contents of P

3.2 Test 10B: SAE

```
TEST 10B:  SAE 0.
    SKIP?    C(P)    C(A)    C(Y)
```

SKIP?	Indicates whether or not a skip occurred
C(P),C(A), C(Y)	Same as Test 10A.

3.3 Test 10C: Logical Operations

```
TEST 10C:  BCO, BSE, BCL.
    RESULT    A OP    Y OP
```

RESULT	C(A) after operation
A OP	Operand in A before execution
Y OP	Operand in memory

4. NON-ERROR MESSAGES

None

5. SPECIAL HALTS

None

6. LINC PROGRAM SEQUENCES

```
L0021,         (STA 0, SAE 0, BCO 0, BSE 0, BCL 0)
                              /Each instruction indirectly addresses L0000
                              /through L0022
L0022,         0000           /Points to L0000, and acts as a HLT for
                              /checking operation of P.
L0023,         HLT
L0024,         HLT            /Required for testing SAE.

L0000,         operand
```

7. EXECUTION TIME

14 sec

8. ERROR STOPS AND INFORMATION

Error Message	C(AC)	Data Printed	Program Tag
INTS nnnn	No halt	INTS status	None
	A spurious LINC interrupt occurred. The octal number nnnn represents the interrupt status as read into the PDP-8 AC. See table at the front of this manual to determine the cause of the interrupt. The program does not stop.		
STA	1000	All entries (10A)	E1000
	This error occurs if any of the following failures are detected: a. C(A) not stored correctly b. C(A) altered during execution c. C(P) not indexed properly.		
SAE	1010	All entries (10B)	E1010
	a. C(A) altered during execution b. C(Y) altered during execution c. Skip failed to operate correctly. If the skip was expected, and did not occur, the A and C entries under SKIP? will be 0 and 1, respectively. If an unexpected skip occurred, the entries will be 1 and 0.		
BCO	1020	All entries (10C)	E1020
BSE	1021	All entries (10C)	E1021
BCL	1022	All entries (10C)	E1022
	For each of these logic operations, an error will be detected if the RESULT of the logical comparison is incorrect, or if the contents of Y are altered. The original A OP is printed for information.		

Description continued from page T10-1

The test makes the following comparisons using the two operands:

$C(A)$ with $C(Y)$
$\overline{C(A)}$ with $C(Y)$
$\overline{C(A)}$ with $\overline{C(Y)}$
$C(A)$ with $\overline{C(Y)}$

In this way, all possible states of each bit of A and of Y are compared with all possible states of the corresponding bit in the other register. Each comparison is performed 4096 times.

TEST 11 - ADA AND FLO

1. DESCRIPTION

NOTE: This test is used as the example for detailed discussion in Vol. 2, chapter 2.

The arithmetic properties of ADA are identical to those of ADD, and the methods used in this test are identical, even to the coding, to those used in Test 5, ADD. For details, please refer to the write-up for that test.

Because ADA is a full-index class instruction, Test 11 also checks for correct addressing. In the course of executing the instruction, the contents of the S-register are changed three times: once when the instruction is fetched from memory, again when the address of the β-register is extracted, and finally when the effective address is obtained from the β-register. In this test, as in most of the other tests of full-index class instructions, the LINC program and data locations have been chosen to insure that most of the bits of S, and especially S8-11, change state three times. To achieve this worst-case addressing situation, the ADA instruction is stored in L20; the β-register is L17, and the effective address, Y, is L1760, where the Y operand is stored.

When the data is tested, the contents of the β-register are examined to see that they have not be changed.

2. SPECIAL SWITCH SETTINGS

LS 9	up	Continue the Random Number Test

3. PRINTOUT HEADING

```
TEST 11: ADA 17.
    A OP  YOP  SUM  FLO  C(BETA)
```

A OP	Operand in A before execution
Y OP	Operand in memory
SUM	Sum left in A
FLO	State of the FLO flip-flop
C(BETA)	Contents of β-register 17

4. NON-ERROR MESSAGES

None

5. SPECIAL HALTS

None

6. LINC PROGRAM SEQUENCE

```
L0020,        ADA 17        /L17 is the β-register
L0021,        FLO           /Is FLO set?
L0022,        HLT           /No.  C(P)=0023.
L0023,        HLT           /Yes.  C(P)=0024

L0017,        1760          /Effective address of Y
L1760,        Y operand
```

7. EXECUTION TIME

21 sec

8. ERROR STOPS AND INFORMATION

Error Message	C(AC)	Data Printed	Program Tag
INTS nnnn	No halt	INTS status	None
	A spurious LINC interrupt occurred. The octal number nnnn represents the interrupt status as read into the PDP-8 AC. See table at the front of this manual to determine the cause of the interrupt. The program does not stop.		
FLO ERROR	No halt (1110)	None	E1110
	If the FLO FF is not correct, this message is printed to call the operator's attention to the FLO column of the subsequent printout.		
ADA	1100	All entries	E1100
	The corresponding error data are printed if a. The SUM is incorrect, b. The Y OP or the contents of β are disturbed, c. A FLO error has occurred. FLO Errors: If FLO FF is not set when it should be, the A and C entries under FLO will be 0 and 1, respectively. If the flip-flop is set incorrectly, the entries are 1 and 0, respectively.		

TEST 12 - ADM AND FLO

1. DESCRIPTION

Please refer to Test 5, ADD, for a discussion of the methods used to test the arithmetic functions of ADM.

The addressing of ADM is designed for worst-case conditions, that is, for maximum bit change of the S-register when the instruction is executed. To achieve this, the ADM instruction is stored in L20, the β-register is L17, and the Y operand is stored in L2760, which is in the LINC upper memory bank.

Although C(BETA) is not examined, C(Y) is checked to determine whether the sum was stored properly or not.

2. SPECIAL SWITCH SETTINGS

LS 9	up	Continue the Random Number Test

3. PRINTOUT HEADING

```
TEST 12:  ADM 17.
     A OP   Y OP   SUM(A)   SUM (Y)   FLO
```

A OP	Operand in A before execution
Y OP	Operand in Y before execution
SUM(A)	Sum left in A
SUM(Y)	Sum left in memory
FLO	State of the FLO flip-flop after execution

4. NON-ERROR MESSAGES

None

5. SPECIAL HALTS

None

6. LINC PROGRAM SEQUENCE

```
L0020,          ADM 17
L0021,          FLO
L0022,          HLT          /Here if FLO not set
L0023,          HLT          /Here if FLO set

L0017,          2760         /Effective address of Y
L2760,          *            /*Y Operand before execution,
                             /SUM(Y) after execution
```

7. EXECUTION TIME

20 sec

8. ERROR STOPS AND INFORMATION

Error Message	C(AC)	Data Printed	Program Tag
INTS nnnn	No halt	INTS status	None
	A spurious LINC interrupt occurred. The octal number nnnn represents the interrupt status as read into the PDP-8 AC. See table at the front of this manual to determine the cause of the interrupt. The program does not stop.		
FLO ERROR	No halt (1210)	None	E1210
	If the state of the FLO flip-flop is not correct, this message is printed to call the operator's attention to the FLO column of the subsequent printout.		
ADM	1200	All entries	E1200
	The pertinent data are printed if a. Either sum, in A or in Y, is incorrect, or b. A FLO error occurs. A OP and Y OP are also printed. FLO errors: If the FLO FF is not set when it should be, the A and C entries under FLO will be 0 and 1, respectively. If the FLO FF is set incorrectly, the entries will be 1 and 0, respectively.		

TEST 13 - LAM

Also tests FLO I and LZE I.

1. DESCRIPTION

The methods used for this test are similar to those used for ADD, ADA, and ADM. However, because the Link is involved in the addition (and also because the addition is now 2's complement), the simulation is much more complicated. The LINC operands require some setting up, so that a longer LINC sequence is required for this test than for the others mentioned. This sequence is given in section 6.

The combination of FLO I, LZE I, and HLT instructions provides four distinct stopping places, and thus four possible values for C(P). Each value of C(P) corresponds to one of the four possible combinations of overflow and carry conditions that can exist after executing LAM.

2. SPECIAL SWITCH SETTINGS

LS 9 up Continue the Random Number Test

3. PRINTOUT HEADING

```
TEST 13: LAM 17.
     SUM(A)  A OP  Y OP  LINK  SUM(Y)  FLO
                         B  A
```

SUM(A)	Sum left in A
A OP	Operand in A before execution
Y OP	Operand in memory before execution
LINK B A	State of the Link before (B) and after (A) execution
SUM(Y)	Sum left in memory
FLO	State of the FLO flip-flop after execution

4. NON-ERROR MESSAGES

None

5. SPECIAL HALTS

None

6. LINC PROGRAM SEQUENCE

```
L20,        CLR         /Clear A, Z, L
L21,        ADD 0       /Put Link state in A (0 or 1)
L22,        ROR I 1     /Set Link from A 11
L23,        ADD 35      /Put A operand in A
L24,        LAM 17
L25,        FLO I       /Did overflow occur?
L26,        JMP 32      /Yes.
L27,        LZE I       /No.  Did carry occur?
L30,        HLT         /No.  C(P)=0031
L31,        HLT         /Yes.  C(P)=0032
L32,        LZE I       /FLO did skip.  Was there carry also?
L33,        HLT         /No.  C(P)=0034
L34,        HLT         /Yes.  C(P)=0035

L35,        (A-operand before test)
L00,        (State of Link,to be set before execution)
L17,        1760        /Effective address of Y
L1760,      *           /*Y-operand before test, SUM(Y) after.
```

7. EXECUTION TIME

40 sec

8. ERROR STOPS AND INFORMATION

<table>
<tr><th>Error Message</th><th>C(AC)</th><th>Data Printed</th><th>Program Tag</th></tr>
<tr><td>INTS nnnn</td><td>No halt</td><td>INTS status</td><td>None</td></tr>
<tr><td></td><td colspan="3">A spurious LINC interrupt occurred. The octal number nnnn represents the interrupt status as read into the PDP-8 AC. See table at the front of this manual to determine the cause of the interrupt. The program does not stop.</td></tr>
<tr><td>FLO ERROR</td><td>No halt
(1310)</td><td>None</td><td>E1310</td></tr>
<tr><td></td><td colspan="3">If the state of FLO FF is not correct, this message is printed to call the operator's attention to the FLO column of the printout.</td></tr>
<tr><td>CARRY ERROR</td><td>No halt
(1320)</td><td>None</td><td>E1320</td></tr>
<tr><td></td><td colspan="3">If the state of the Link is not correct, this message calls the operator's attention to the LINK column of the printout.</td></tr>
<tr><td>LAM</td><td>1300</td><td>All entries</td><td>E1300</td></tr>
<tr><td></td><td colspan="3">If either sum is incorrect, the pertinent data are printed.
If a CARRY out of A_0 fails to set the Link, or if the Link is not left clear when no carry occurs, the pertinent data are printed under LINK. The state of the Link before addition is printed as the first digit of the C entry, unless the state is 0. The actual state of the Link after addition is printed as the A entry; the correct final state appears as the last digit of the C entry. The column heads B and A identify the Link states before and after, respectively.
FLO Errors:
a. If the FLO FF was not set, and should have been, the A and C entries will be 0 and 1, respectively.
b. If the flip-flop was set and should not have been, the entries are 1 and 0, respectively.</td></tr>
</table>

TEST 14 - SET, XSK

1. DESCRIPTION

This test examines SET and XSK, two of the four ALPHA-class instructions.

1.1 Test 14A SET and SET I

First, the instruction SET I is tested for every α-register. The instruction is stored in register L1760, with the operand in L1761. This provides a worst-case addressing situation for the S-register, with most of its bits changing state between the reading of Y and the storing of C(Y) in the α-register. SET I is tested for the transfer of all possible numbers from L1761 to each of the α-registers (L0-L17). During the transfer of data, the LINC accumulator contains the number 2634. C(A) are tested after execution to see if they have been disturbed during the transfer of data.

The test of SET provides another worst-case addressing situation. This time the instruction is in L36; L37 contains the effective address of the operand, which is in L1740. The instruction is tested for all values of C(Y) from 1777 through 7777, for each α-register. The previous C(ALF) are not cleared.

1.2 Test 14B XSK

XSK is tested for correct operation of the skip. All possible values of C(ALF) are tested; whenever the low-order 10 bits of ALPHA contain 1777, the instruction should skip, otherwise it should not.

XSK I is tested for correct indexing of C(ALF). Since the indexing is performed modulo 2^{10}, a special testing scheme is used to investigate the action of XSK I whenever the low-order 10 bits of ALPHA contain 1777.

2. SPECIAL SWITCH SETTINGS

None

3. PRINTOUT HEADINGS

3.1 Test 14A SET

```
TEST 14A:  SET, SET I.
    C(ALF)  ALFA  C(Y)  Y  C(A)  C(P+1)
```

C(ALF)	Contents of the α-register (L0-L17)
ALFA	Address of the α-register being tested
C(Y)	Contents of Y, i.e., the operand being moved
Y	Octal address of Y
C(A)	Contents of A
C(P+1)	Contents of L0037, i.e., the effective address of Y

3.2 Test 14B XSK

```
TEST 14B:  XSK 17, XSK I 17.
    C(ALF)  C(P)  C(A)
```

C(ALF)	Contents of α-register
C(P)	Contents of P
C(A)	Contents of A

4. NON-ERROR MESSAGES

None

5. SPECIAL HALTS

None

6. LINC PROGRAM SEQUENCES

6.1 SET I Test

```
L1760,          SET I           /All α-registers tested
L1761,          Y operand
L1762,          HLT
```

6.2 SET Test

```
L0036,          SET
L0037,          1740            /Effective address of operand
L0040,          HLT

L1740,          Y operand
```

6.3 XSK, XSK I Test. α-register is L17.

```
L0020,          (XSK 17 or XSK I 17)
L0021,          HLT
L0023,          HLT

L0017,          α-register
```

7. EXECUTION TIME

24 sec

8. ERROR STOPS AND INFORMATION

Error Message	C(AC)	Data Printed	Program Tag
INTS nnnn	No halt	INTS status	None
	A spurious LINC interrupt occurred. The octal number nnnn represents the interrupt status as read into the PDP-8 AC. See table at the front of this manual to determine the cause of the interrupt. The program does not stop.		
SET I	1400	All except C(P+1)	E1400
	The corresponding error data are printed if a. Information is not transferred correctly. C(ALF) in error. ALFA, C(Y), Y printed for information. b. C(Y) altered during execution. c. C(A) disturbed during transfer.		
SET	1410	All entries	E1410
	All errors listed above apply. In addition if effective address is changed, C(P+1) are printed.		
XSK	1420	All entries (14B)	E1420
	If the contents of ALFA or of A are disturbed, the pertinent data are printed. If a skip error occurs, the data under C(P) are interpreted as follows: If the skip occurred when it was not expected, the A and C entries are 23 and 22, respectively. If the skip failed to occur at the proper time, the entries are 22 and 23, respectively.		
XSK I	1430	C(ALF), C(A)	E1430
	The skip is not tested. If C(ALF) or C(A) are incorrect, the data are printed. Watch for indexing errors, indicated by a difference of 1 between the A and C entries of C(ALF).		

TEST 15 - SRO

1. DESCRIPTION

The SRO instruction is tested for accurate functioning, using all β-registers, with L2760 as Y, the location of the operand. The number 2634 is placed in Y and the instruction is executed 4096 times, the number being rotated right each time. After each execution, the contents of Y and A are tested for correct values, and C(P) is tested to detect a skip error, if any. Before each execution, the LINC A-register is loaded with the number 2634, and afterwards tested to see that it was not changed.

2. SPECIAL SWITCH SETTINGS

None

3. PRINTOUT HEADING

```
TEST 15:  SRO.  ALL BETAS;  Y=L2760
     C(Y)   C(P)   BETA   C(A)
```

C(Y)	Contents of Y (operand) after execution
C(P)	Contents of P
BETA	Octal address of β-register being used
C(A)	Contents of A

4. NON-ERROR MESSAGES

None

5. SPECIAL HALTS

None

6. LINC PROGRAM SEQUENCE

```
L0020,          SRO β          /All β-registers used
L0021,          HLT            /Here if no skip
L0022,          HLT            /Here if skip

L2760,          (operand)
```

7. EXECUTION TIME

14 sec

8. ERROR STOPS AND INFORMATION

Error Message	C(AC)	Data Printed	Program Tag
INTS nnnn	No halt	INTS status	None
	A spurious LINC interrupt occurred. The octal number nnnn represents the interrupt status as read into the PDP-8 AC. See table at the front of this manual to determine the cause of the interrupt. The program does not stop.		
SRO	1500	All entries	E1500
	a. Rotate failure. The operand in Y was not rotated right properly. C(Y) in error. b. Skip error. If the skip failed to occur when $C(Y_{11})=0$, the A and C entries under C(P) will be 22 and 23, respectively. If the skip occurred when $C(Y_{11})=1$, the entries will be 23 and 22, respectively. c. C(A) disturbed. Corresponding data are printed. In every case, the absolute octal address of the β-register in use is printed for information.		

TEST 16 - MUL

1. DESCRIPTION

Logically, the MUL test is similar to the Addition tests, in that a table of operands is tested, each against all possible numbers. The table is the same as that for the Add tests, there being eight selected numbers including 0000. For this test, however, the table is split into two parts: the numbers 0000, 7777, 4000, and 3777 are tested using the addressing scheme MUL I 0; the numbers 5252, 2525, 7700, 0077 are tested using the scheme MUL I 16. Each number is used as a multiplicand for all possible multipliers, then as a multiplier for all possible multiplicands.

A random number test is also performed, using the MUL I 16 addressing scheme. The random number generator is the same as for ADD, ADA, ADM, and LAM, but the seeds are 5163 and 0666.

A test cycle proceeds as follows:

The FLO FF is cleared, by executing ADD 1761 to a cleared A-register. The operands are each tested for sign, and the absolute values are stored in intermediate registers. The state of the Link, representing the sign of the product, is stored. Each absolute value is then tested for magnitude; if either value is 0, the multiplication is not simulated. The product registers are cleared, unless the Link indicator is 1, in which case APROD is set to 7777.

If neither operand is 0, the multiplication is simulated, using a subroutine which follows the LINC hardware algorithm rather closely. In fact, the simulator resembles the standard PDP-8 Multiply Subroutine, with the necessary changes to conform to 1's complement rather than 2's complement arithmetic.

After the simulation, the LINC program sequence is executed. This sequence is given in Sect.6.

After execution, the results are compared with the results of the simulation, and errors are identified and printed. The tests are performed in the following order:

BETA (when applicable)
FLO
Link
Product (A and Z)
Y operand

The locations of the LINC program and the Y operand are designed to provide worst-case conditions for the MUL I 16 addressing scheme. For this test, the MUL instruction is in L1760, the β-register is L16, and the Y operand is in L2761. This results in a maximum bit change in S when the addressing is set up.

2. SPECIAL SWITCH SETTINGS

None

3. PRINTOUT HEADING

```
TEST 16:  MUL.
ADDR        APROD        ZPROD        A OP        Y OP        H  L        C(BETA)
```

ADDR Addressing scheme being used. This constitutes the error message for a printout (see facing page).
APROD That part of the product in A after execution
ZPROD That part of the product in Z after execution
A OP The multiplicand
Y OP The multiplier
H The state of the H-bit before execution
L The state of the Link after execution
C(BETA) The contents of the β-register during MUL I 16 test.

4. NON-ERROR MESSAGES

None

5. SPECIAL HALTS

None

6. LINC PROGRAM SEQUENCE

```
L1760,          (MUL I 0 or MUL I 16)
L1761,          *               /*Operand for  MUL I 0 , NOP for  MUL I 16
L1762,          STC 1771        /Store product and clear A
L1763,          LZE             /Is Link set?
L1764,          ADD 1770        /Yes.  Put indicator in A.
L1765,          FLO             /Is FLO FF set?
L1766,          HLT             /No.  C(P)=1767
L1767,          HLT             /Yes.  C(P)=1770

L1770,          0001            /Link set indicator
L1771           ----            /Holds product after execution

L0016,          2761            /β-register for MUL I 16
L2761,          operand for MUL I 16
```

7. EXECUTION TIME

50 sec

8. ERROR STOPS AND INFORMATION

Error Message	C(AC)	Data Printed	Program Tag
INTS nnnn	No halt	INTS status	None
	A spurious LINC interrupt occurred. The octal number nnnn represents the interrupt status as read into the PDP-8 AC. See table at the front of this manual to determine the cause of the interrupt. The program does not stop.		
BETA ERROR	No halt (1630)	None	E1630
	This message is printed if the β-register was not indexed properly, or contained the wrong effective address for any reason. Check C(BETA) entries of printout.		
FLO FF SET	No halt (1620)	None	E1620
	The MUL instruction should not disturb the FLO flip-flop, which is cleared each cycle before the test is performed. If the flip-flop is set, this message is printed. Unless there are other errors, there will be no data printout; the message will be printed repeatedly until the condition is corrected or the operator stops the program manually (see Chapter 5, section 6.8).		
I 0	1600	All except C(BETA)	E1600
	The instruction is MUL i 0. No β-register is involved, and only integer arithmetic may be performed. Causes of error: a. Either or both halves of the product are incorrect. These are APROD and ZPROD, which are C(A) and C(Z), respectively. b. C(Link) is incorrect. L entry printed. c. Multiplier was altered. (Y OP)		
I 16	1610	All entries	E1610
	The instruction is MUL i 16. All errors noted above apply; in addition: If the effective address of the multiplier was changed, the data are printed under C(BETA). This is also signaled by the message BETA ERROR. Fractional arithmetic may be specified; the state of the H-bit is printed for information.		

TEST 17 - LDH

Includes full memory address test.

1. DESCRIPTION

The basic approach of this test is similar to that for Test 7, LDA, which also tests the entire memory addressing scheme. All possible combinations of I and β are used to access all except the last 200_8 registers of the upper bank of LINC memory (this is to insure that the BIN Loader is not destroyed).

1.1 Test A LDH I 0

This addressing scheme causes the contents of the left half of the register immediately following the instruction to be placed in the right half of the accumulator. The test performs this operation for every possible value of $C(Y_L)$; the right half of Y is set to 77 to catch possible half-index errors.

1.2 Test B LDH 0. Right Half-Word Test

This test uses the register following the instruction as an indirect address register. All possible values of the contents of the right half of Y are loaded into the accumulator. The instruction is stored in L21; the effective address is in L22, and the operand is in L30. Register L31 is set to 7777, to provide a check on half-indexing errors.

1.3 Test C LDH 0. All registers.

This and the next two tests use all except the last 200_8 registers of the upper bank of LINC memory. For every register, the contents of Y are 4422, and the instruction is tested for loading each half of the register into the AC. For this test, L22 contains the effective address, and the first Y-register tested is L30. In every case, half-index errors are checked for.

1.4 Test D LDH β

Every register from L30 through L3577 is tested, using each β-register in turn. At any time, all unused β-registers are cleared. C(Y) is always 4422, and half-index errors are tested for by setting the next succeeding register (Y + 1) to 7777.

1.5 Test E LDH I β

All β-registers and all LINC memory registers from L30 through L3577 are tested, using 4422 as the operand. In this test, the β-register is automatically half-indexed, so failures are detected by presetting the register preceding Y with 7777.

2. SPECIAL SWITCH SETTINGS

None

3. PRINTOUT HEADING

```
TEST 17:  LDH. FULL MEMORY ADDRESS TEST.
     C(A)   C(Y)   C(BETA)   BETA
```

C(A)	Contents of A after execution
C(Y)	Contents of Y after execution
C(BETA)	Contents of β-register after execution
BETA	Number of β-registers in use

4. NON-ERROR MESSAGES

None

5. SPECIAL HALTS

None

6. LINC PROGRAM SEQUENCES

6.1 Test A

```
L21,           LDH I 0
L22,           (operand)
L23,           HLT
```

6.2 Test B

```
L21,           LDH 0
L22,           0030           /effective address of operand
L23,           HLT

L30,           (operand)
```

6.3 Test C

```
L21,           LDH 0
L22,           Y              /effective address varies
L23,           HLT

L30-L3577                     /registers tested
```

6.4 Tests D and E

```
L21,           LDH (I) β      /Test E uses 'I' bit
L22,           HLT
```

All β-registers, and all addresses from L30-L3577, inclusive.

7. EXECUTION TIME

35 sec

8. ERROR STOPS AND INFORMATION

<table>
<tr><th>Error Message</th><th>C(AC)</th><th>Data Printed</th><th>Program Tag</th></tr>
<tr><td>INTS nnnn</td><td>No halt</td><td>INTS status</td><td>None</td></tr>
<tr><td></td><td colspan="3">A spurious LINC interrupt occurred. The octal number nnnn represents the interrupt status as read into the PDP-8 AC. See table at the front of this manual to determine the cause of the interrupt. The program does not stop.</td></tr>
<tr><td>HNDX B</td><td>No halt
(1760)</td><td>None</td><td>E1760</td></tr>
<tr><td></td><td colspan="3">During Tests A through D, this error occurs if the LINC B-register is half-indexed incorrectly. The error is inferred from the contents of A; if C(A) match the contents of the next successive half-word, an HNDX B error is assumed. Check C(A) entries of the printout.
During Test E, this message is printed if the half-index of B fails to occur when executing LDH I β. In this case, the comparison should be made between C(A) and the preceding half-word (see description of numerical values used).</td></tr>
<tr><td>HNDX MB</td><td>No halt
(1770)</td><td>None</td><td>E1770</td></tr>
<tr><td></td><td colspan="3">A half-index error occurred in the PDP-8 MB. The error is inferred from the evidence that C(BETA) is incorrect. In tests A through D, the MB should not be indexed; in test E, it should be. Check C(BETA) entries of the printout.</td></tr>
<tr><td colspan="4">In each of the following diagnostics, C(A), C(Y), and C(BETA) are all possible sources of trouble. C(Y) should never be altered during execution. In test E, C(BETA) should be indexed according to the LINC half-indexing scheme. In Tests D and E, the octal address of the β-register is printed for information.</td></tr>
<tr><td>TST A</td><td>1700</td><td>All entries</td><td>E1700</td></tr>
<tr><td></td><td colspan="3">LDH I 0. Loads all values from left half of L22.</td></tr>
<tr><td>TST B</td><td>1710</td><td>All entries</td><td>E1710</td></tr>
<tr><td></td><td colspan="3">LDH 0. Loads all values from right half of L30.</td></tr>
<tr><td>TST C</td><td>1720</td><td>All entries</td><td>E1720</td></tr>
<tr><td></td><td colspan="3">LDH 0. Loads A from L30-L3577, inclusive.</td></tr>
<tr><td>TST D</td><td>1730</td><td>All entries</td><td>E1730</td></tr>
<tr><td></td><td colspan="3">LDH. Loads A from L30-L3577; all β's.</td></tr>
<tr><td>TST E</td><td>1740</td><td>All entries</td><td>E1740</td></tr>
<tr><td></td><td colspan="3">LDH I β. All β's; L30-L3577.</td></tr>
</table>

TEST 20 - STH, SHD

1. DESCRIPTION

The tests of STH and SHD are similar to their full-word counterparts, STA and SAE.

1.1 Test A STH 17

The instruction is tested for the ability to store any number from the right half of the AC into the left and then right halves of a memory register. To insure that addressing conditions are as rigorous as possible, the instruction is stored in register L1761, the effective address in β-register L0017, and the operand, Y, in L1760. For each half of Y, the instruction is executed for all values of C(A), so that half-word values of 00 through 77 are stored for each value of the other half-word of the A-register. Before each cycle is executed, the Y-register is loaded with the number 6633, to provide a check on what should remain the undisturbed half of the word. For example, in storing the number 56 in the right half of Y, the resulting C(Y) should be 6656, since the left half should remain undisturbed.

For each cycle of the test, C(A) and C(BETA) are tested to see that they have been left unchanged during the execution of STH.

1.2 Test B SHD 17, halves equal

The addressing scheme for this test is the same as for Test A. The comparison is made for all values of C(Y) and C(A), with the tested halves equal. At no time are C(A_L) equal to the contents of the untested half of Y.

1.3 Test C SHD 17, halves unequal

Again, the addressing scheme is the same as that of the preceding tests. For each value of the content of the tested half of Y, all possible unequal values of C(A_L) are used. The untested halves of both registers are preloaded with the number 66, so that these halves are always equal to each other.

After each cycle of Tests B and C, the contents of A, Y, and BETA are tested for possible disturbance.

2. SPECIAL SWITCH SETTINGS

None

3. PRINTOUT HEADINGS

3.1 Test A STH

```
TEST 20A:  STH 17.  Y=L2760.
     C(Y)    C(A)    C(BETA)
```

C(Y)	Contents of Y
C(A)	Contents of A
C(BETA)	Effective address: contents of β-register.

3.2 Test B and C SHD

```
TEST 20, B AND C:  SHD 17.
     SKIP?   C(P)   C(A)   C(Y)   C(BETA)
```

SKIP?	Indicates occurrence or non-occurrence of a skip.
C(P)	Contents of P
C(A)	Contents of A
C(Y)	Contents of Y
C(BETA)	Effective address: contents of β-register.

4. NON-ERROR MESSAGES

None

5. SPECIAL HALTS

None

6. LINC PROGRAM SEQUENCES

6.1 Test A

```
L1761,          STH 17
L1762,          HLT

L0017,          *               /*2760 for left half, 6760 for right.
L2760,          operand
```

6.2 Tests B and C

```
L1761,          SHD 17
L1762,          HLT
L1763,          HLT

L0017,          *               /*2760 for left half, 6760 for right.
L2760,          operand
```

7. EXECUTION TIME

7 sec

8. ERROR STOPS AND INFORMATION

Error Message	C(AC)	Data Printed	Program Tag
INTS nnnn	No halt	INTS status	None
	A spurious LINC interrupt occurred. The octal number nnnn represents the interrupt status as read into the PDP-8 AC. See table at the front of this manual to determine the cause of the interrupt. The program does not stop.		
LEFT	2000	C(Y), C(A), C(BETA)	E2000
RIGHT	2010	C(Y), C(A), C(BETA)	E2010
	STH into LEFT or RIGHT half of memory register failed. Probable causes: a. C(A) disturbed b. Datum stored incorrectly. C(Y) printed. c. Effective address incorrect. C(BETA) printed.		
L EQ	2020	All entries	E2020
R EQ	2030	All entries	E2030
	One or more of the following errors occurred during a test of SHD with left (L EQ) or right (R EQ) half of memory word equal to $C(A_R)$: a. The instruction skipped. C(P) printed, along with indicators of 1 and 0 as A and C entries under SKIP? b. C(A), C(Y), or C(BETA) were disturbed. The pertinent data are printed.		
L NEQ	2040	All entries	E2040
R NEQ	2050	All entries	E2050
	One or more of the following errors occurred during a test of SHD with the designated half (left: L NEQ; right: R NEQ) of memory word not equal to $C(A_R)$: a. The instruction failed to skip. C(P) printed, and skip indicators of 0 and 1, respectively, for A and C entries under SKIP? b. C(A), C(Y), or C(BETA) disturbed. Pertinent data are printed.		

TEST 21 - DIS, DSC

1. DESCRIPTION

1.1 DIS 1 and DIS I 1

The non-indexing instruction, DIS 1, displays a horizontal line from the left edge of the screen to the right, at vertical coordinate V=050. This line is displayed on channel 0.

The indexing instruction, DIS I 1, displays a horizontal line from left to right at V=010, on channel 1.

After each point is displayed, the pertinent registers are tested.

1.2 DSC and DSC I 17

DSC displays the characters "TEST 21" on channel 0 between the horizontal lines displayed by DIS. Starting co-ordinates are H = 154, V = 030.

DSC I 17 displays the characters "DISPLAY!" on channel 1, between the lines. Starting co-ordinates are H = 414, V = 030.

After each half-character is displayed, the pertinent registers are tested.

When operating correctly, the display should appear as in figure T21-1.

The entire pattern is displayed 16 times.

2. SPECIAL SWITCH SETTINGS

None

3. PRINTOUT HEADING

```
TEST 21:  DIS, DSC.
     HORIZ   VERT   C(Z)   PATRN   C(BETA)
```

HORIZ	Horizontal coordinate: C(L1)
VERT	Vertical coordinate: C(A)
C(Z)	Contents of Z (should be clear after DSC)
PATRN	Pattern word for DSC
C(BETA)	Effective address for DSC

4. NON-ERROR MESSAGES

None

5. SPECIAL HALTS

None

6. LINC PROGRAM SEQUENCE

```
L1756,          CLR              /Clear A, Z, L
L1757,          ADD 1762         /V-coordinate to A
L1760,          (DIS 1, DIS i 1; DSC 17, DSC i 17)
L1761,          HLT

L0001,          H-coordinate
L0017,          Effective address of pattern for DSC
L2760,          Pattern word for DSC
L1762,          V-coordinate storage
```

7. EXECUTION TIME

5 sec

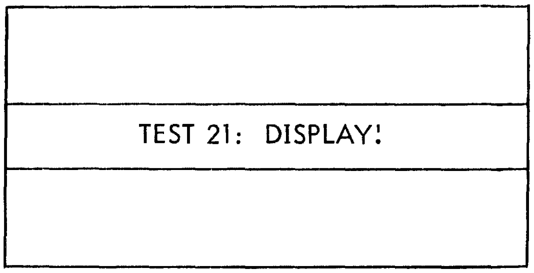


Figure T21-1 Display Pattern

8. ERROR STOPS AND INFORMATION

Error Message	C(AC)	Data Printed	Program Tag
INTS nnnn	No halt	INTS status	None
	A spurious LINC interrupt occurred. The octal number nnnn represents the interrupt status as read into the PDP-8 AC. See table at the front of this manual to determine the cause of the interrupt. The program does not stop.		
DIS DIS I	2100 2101	HORIZ, VERT, C(Z) HORIZ, VERT, C(Z)	E2100 E2101
	Internal operation error in DIS. Possible sources: a. Either co-ordinate may be incorrect. HORIZ represents C(L1); VERT represents C(A). b. Z-register may not be clear. c. An indexing failure may have occurred, which would result in a difference of 1 between A and C entries under HORIZ. This may also be visible as a blank spot in one of the two horizontal lines displayed.		
DSC DSC I	2110 2111	All entries All entries	E2110 E2111
	Internal operation error in DSC. Possible sources: a. The coordinates may be left incorrect. HORIZ and VERT represent C(L1) and C(A), respectively. b. The Z-register, which contains the pattern word during execution, was not left clear. c. The effective address, C(BETA), was incorrect. Watch for possible indexing failures of DSC i 17. The pattern word for the displayed half-character is printed for information.		

TEST 22 - MEMORY BANK SELECTORS

1. DESCRIPTION

The functions of the Memory Bank Selectors and their associated LINC instructions, LMB and UMB, are tested.

Before the test starts, Left Switches 6, 7, and 8 must be set to the number of additional memory fields (4096 words each), if any, that are attached to the computer. The program reads the switches, and prints the following message:

UPPER LIMIT: BANK nn

where nn is the number of the highest memory bank available to the LINC processor. Table T22-1 gives the value of nn for each configuration of memory.

TABLE T22-1 MEMORY BANK LIMITS FOR EXTENDED MEMORY

No. of Additional Fields	Number of Highest Bank
0	3
1	7
2	13
3	17
4	23
5	27
6	33
7	37

The two sets of selectors are tested separately, the UMBS being examined first. For each set, the following test sequence is performed:

a. From the existing setting, the selectors are set to 01.

b. An attempt is made to set the selectors to 00.

c. The selectors are set to each legitimate value, from 01 through the limit specified by nn. After each setting, the selectors are returned to the original value.

d. An attempt is made to set the selectors to each of the illegal values from nn+1 through 37.

e. Steps b, c, and d are repeated using successive legitimate settings of the selectors as a starting point.

f. The selectors are reset to their standard value.

Table T22-2 illustrates this testing sequence.

Description continued on page T22-4

2. SPECIAL SWITCH SETTINGS

LS 6, 7, 8	Number of additional PDP-8 memory fields

3. PRINTOUT HEADING

```
TEST 22:  MEMORY BANK SELECTORS
      SELECTORS
    BEFORE   AFTER      JMP TO
```

BEFORE	C(Selectors) before test
AFTER	C(Selectors) after test
JMP TO	Number of memory bank to which control was transferred after setting LMBS.

4. NON-ERROR MESSAGE

At beginning of test, LS 6, 7, and 8 are read and the number of LINC banks is calculated. The following message is then printed:

```
UPPER LIMIT:  BANK nn
```

where nn is the octal number of the highest memory bank available.

5. SPECIAL HALT

For Error E2230 (see next page): C(PC)=1140.

Recovery: Set RS to 1000 (restart test), or
0100 (return to Monitor)
Press PDP-8 START

6. LINC PROGRAM SEQUENCES

6.1 UMBS Test

```
L1,            UMB n
L2,            HLT
```

6.2 LMBS Test

Part 1

```
L1,            LMB m
L2,            HLT
```

Part 2

```
L1,            JMP 3          /This effects the change of selectors
L2,            (HLT)
L3,            ADD 5          /Puts bank number in A
L4,            HLT
L5,            bank number m
```

7. EXECUTION TIME

Varies with number of added fields.

8. ERROR STOPS AND INFORMATION

Error Message	C(AC)	Data Printed	Program Tag
INTS nnnn	No halt	INTS status	None
	A spurious LINC interrupt occurred. The octal number nnnn represents the interrupt status as read into the PDP-8 AC. See table at the front of this manual to determine the cause of the interrupt. The program does not stop.		
OUT OF BOUNDS	No halt (2201)	None	E2201
	The selectors were set to 00 or to a value beyond the upper limit.		
TOO SOON! NO JMP	No halt (2211)	None	E2211
	LMBS were set to new value after execution of LMB, but before JMP.		
JMP WRONG	No halt (2212)	None	E2212
	Program control was transferred to the wrong bank when LMBS were set. See JMP TO column in subsequent printout.		
UMBS	2200	AFTER	E2200
	The UMBS were set incorrectly. The state of the selectors BEFORE the test is printed for information.		
LMBS	2210	AFTER, JMP TO	E2210
	The LMBS were set incorrectly, or control was transferred to the wrong bank. The state of the selectors BEFORE the test is printed for information.		
MANUAL START, PLEASE	0000	None	E2230
	This message is printed at the end of the test if: a. LMBS or UMBS are set to 00. b. LMBS or UMBS are set beyond the upper limit. c. LMBS and UMBS are set to same bank. The operator must manually restart the PDP-8 (see preceding page, section 5), so that the selectors are reset to the standard values.		

Description continued from page T22-1

1.1 LMBS Test

Each cycle of the LMBS test is in two parts.

First, the LMB instruction is set up and executed. The selectors are checked to see if any change has occurred. If it has, the diagnostic message

TOO SOON! NO JMP

is printed.

If the LMBS are unchanged, LMB n is replaced by JMP 3 in the LINC program, and the sequence is executed. At the end of the LINC program, the A-register contains the number of the bank to which control was actually transferred when the JMP was executed. If this is incorrect, the message

WRONG BANK

is printed. The data column, JMP TO, shows the actual and correct bank numbers.

TABLE T22-2 UMBS TEST SEQUENCE

		Original Setting: 03	
	State of UMBS Before Test	Attempted Setting	
a.	03	01	
b.	01	00	
c.	01	01	
	01	01	(return)
	01	02	
	02	01	(return)
	01	03	
	03	01	(return)
	....	....	
	01	nn	
	nn	01	(return)
d.	01	nn+1	
	01	nn+2	
	....	....	
	01	37	
e.	01	02	(new starting value)
(b)	02	00	
(c)	02	01	
	01	02	(return)
	....	....	
(d)	nn	37	
f.	nn	03	(standard setting)

1.2 Cycle Repeats

If an error occurs at any point in the test sequence, the cycle repeat option can be put into effect in the usual manner. If it is not in effect, or upon continuation, the remainder of that part of the test is bypassed. Thus, if an error occurs during the UMBS test, the program proceeds directly to the LMBS test.

1.3 Exit Sequence

At the end of the entire test (which is repeated 30 times if there are no errors), the selectors are examined. If they are set to the standard values (LMBS=02, UMBS=03), no error has occurred, and the diagnostic proceeds to the next test in sequence.

If either set of selectors is set to 00, to a bank beyond memory field 0 (including illegal settings), or to the same bank as the other set of selectors, the following message is printed:

MANUAL START, PLEASE.

and the program stops with C(PC)=1140, C(AC)=0000. The operator may restart the test or return control to the Monitor, but he must do so by pressing PDP-8 START, in order to return the selectors to the standard values.

If these fatal-error conditions do not apply, but if the selectors are not set to standard values, the manual start is not required, and testing continues. However, in case of any error, fatal or not, the tests of extended memory (Tests 23-25) are not executed.

TEST 23 - EXTENDED MEMORY ADDRESS

1. DESCRIPTION

The i-β class addressing is tested for all possible combinations of Memory Bank Selector settings. The test is in two parts.

1.1 Auto-Indexing

Using the LINC instruction, LDH i 17, a worst-case addressing scheme provides a test of half-word indexing when operating across memory field boundaries. All legal combinations of UMBS and LMBS settings are used, except those when C(UMBS)=C(LMBS). The addressing scheme is as follows:

```
L20,          LDH I 17
L17,          7777          /address of right half of L3777
L3777,        0001          /provides HNDX B error check
L2000,        nn77          /datum.  nn varies from 00-77.
```

When the LINC sequence is executed, C(BETA) is incremented by 1, using LINC half-indexing. If the indexing is correct, the resulting effective address is 2000, thus pointing to the left half of the first word in the LINC upper memory bank. If HNDX B fails, the contents of the right half of L3777 will be loaded in the A-register. If HNDX MB fails, C(BETA) will be incorrect.

1.2 Addressing

This test is similar to its counterpart in Test 7. Access to every possible address is tested, from both lower and higher addresses in the lower bank. The instruction used is LDA 0. The test attempts to load the LINC accumulator from every register in the upper memory bank. The test is performed with LDA 0 first in L20, then in L1770.

During the test, each register of the lower memory bank (except those containing the program sequence) contains its own LINC address, augmented by 2000. Thus, L0000 contains 2000, L0001 contains 2001, and so on. The upper memory bank registers each contain the address augmented by 4000, so that L2000 contains 6000, L2001 contains 6001, and so on. All other memory registers are cleared.

After the execution of the LINC sequence, the pertinent data are checked: C(A) and C(Y) for correct data handling, C(BETA) for correct effective address. If the lower bank is accessed, the message

LOWER BANK

is printed. Otherwise, the data printout provides all necessary information.

Description continued on page T23-4

2. SPECIAL SWITCH SETTINGS

LS 6, 7, 8	Number of additional PDP-8 memory fields attached

3. PRINTOUT HEADINGS

3.1 Part 1 Indexing

```
TEST 23A:  INDEXING.  LDH I 17
    C(A)   C(Y)   C(BETA)
```

C(A)	Contents of LINC accumulator
C(Y)	Contents of Y
C(BETA)	Effective address

3.2 Part 2 Addressing

```
TEST 23B: ADDRESSING. LDA 0
FROM     C(A)   C(Y)   C(BETA)
```

FROM	Address of LDA 0
C(A)	Contents of A
C(Y)	Contents of Y
C(BETA)	Effective address

4. NON-ERROR MESSAGES

None

5. SPECIAL HALTS

None

6. LINC PROGRAM SEQUENCES

6.1 Part 1 Indexing

```
L0020,          LDH I 17
L0021,          HLT

L0017,          7777          /Points to L3777; half-indexed to 2000.
L3777,          0001          /HNDX B error detector
L2000,          nn77          /Datum.  nn varies from 00 through 77.
```

6.2 Part 2 Addressing

```
L20 or L1770,   LDA 0
L21 or L1771,   Y             /Effective address
L22 or L1772,   HLT
```

7. EXECUTION TIME

Varies with size of memory extension.
With one extension: 40 sec

8. ERROR STOPS AND INFORMATION

Error Message	C(AC)	Data Printed	Program Tag
INTS nnnn	No halt	INTS status	None
	A spurious LINC interrupt occurred. The octal number nnnn represents the interrupt status as read into the PDP-8 AC. See table at the front of this manual to determine the cause of the interrupt. The program does not stop.		
HNDX B	No halt (2301)	None	E2301
	HNDX B failed to occur during execution of LDH i 17. The wrong register was accessed, and the LINC accumulator probably contains 0001, this being C(L3777).		
HNDX MB	No halt (2302)		
	HNDX MB failed to occur during execution of LDH i 17. The contents of L17, which is the β-register, are incorrect.		
LDH I	2300	All entries, 23A	E2300
	C(A) was not correct, C(Y) was altered, or an HNDX error (see above) occurred.		
LOWER BANK	No halt (2303)	None	E2303
	This message is printed if the LDA 0 instruction accesses a register in the lower memory bank.		
L20	2310	All entries, 23B	E2310
L1770	2320	All entries, 23B	E2320
	An addressing error occurred during execution of LDA 0. The error message gives the LINC address in which the LDA is stored. C(A), C(Y), and C(BETA) are all possible sources of error. If C(A)=0000, the wrong bank was addressed.		

Description continued from page T23-1

The test proceeds through all combinations of UMBS and LMBS (except when they are equal) with the program sequence stored in L20-122. It then repeats the test with the LINC sequence in L1770-L1772.

At the completion of the test, the Memory Bank Selectors are reset to their standard values.

Although this is primarily a test of addressing across data fields, the test is performed even if no extensions are attached. Before the test begins, Left Switches 6, 7, and 8 must be set to the number of additional memory fields provided.

TEST 24 - EXTENDED MEMORY TIMING

1. DESCRIPTION

NOTE: This test is performed only if a memory extension is provided. The operator must set Left Switches 6, 7, and 8 before the test begins.

This test examines the critical timing conditions at GNI and GO following a Rotate-class instruction, when that instruction is executed in an extended memory field. To provide the tightest possible timing situation, the instruction ROR i 17 is executed, for each of all possible values of C(A). C(L) and C(Z) are clear before execution. After execution, C(A), C(L), and C(Z) are tested.

The test is performed from the first bank of each added field; thus, in field 1, bank 4 is used, in field 2, bank 10, and so on. If no additional fields are attached (that is, if LS 6, 7, 8=0), the test is not executed, but returns control immediately to the Monitor.

The entire test is executed eight times.

2. SPECIAL SWITCH SETTINGS

LS 6, 7, 8 number of additional memory fields

3. PRINTOUT HEADING

```
TEST 24:  EXT.MEM.TIMING.
    C(A)  C(Z)  LINK  A ORIG
```

C(A)	Contents of A after rotation
C(Z)	Contents of Z after rotation
LINK	Contents of Link after rotation
A ORIG	Contents of A before rotation

4. NON-ERROR MESSAGES

None

5. SPECIAL HALTS

None

6. LINC PROGRAM SEQUENCE

```
L0,           datum

L20,          CLR          /Clear L, Z, A
L21,          ADD 0        /Put data in A
L22,          ROR I 17     /Rotate 17 (octal) places
L23,          LZE          /Is Link set?
L24,          HLT          /Yes.  C(P)=0025
L25,          HLT          /No.  C(P)=0026
```

7. EXECUTION TIME

Varies with number of added memory fields.
With 1 field: 11 sec

8. ERROR STOPS AND INFORMATION

Error Message	C(AC)	Data Printed	Program Tag
INTS nnnn	No halt	INTS status	None
	A spurious LINC interrupt occurred. The octal number nnnn represents the interrupt status as read into the PDP-8 AC. See table at the front of this manual to determine the cause of the interrupt. The program does not stop.		
ROR I	2400	All entries	E2400
	If any of the three registers involved in the rotation are incorrect, the pertinent data are printed. The original C(A) are also provided.		

TEST 25 - EXT S01 USING SET AND DSC

NOTE: The operator must set Left Switches 6, 7, and 8 befo e this test begins. The test is not performed if no memory extension is attached.

1. DESCRIPTION

When an extended memory is provided, more time is required for setting up LINC address gates than when only one memory field is available. In particular, knowledge of which memory bank is being accessed must be available at T3 in order to set up the proper address selectors. For this purpose, a special flip-flop, EXT S01, is provided. When the state of this flip-flop is 0, the LINC lower memory bank will be accessed; when the state is 1, the upper bank will be accessed. For most i-β class instructions, EXT S01 can be switched only once during the execution of the instruction, when the operand is addressed. For two instructions, however, a subsequent access to lower bank is required. These instructions are SET, in which an operand is transferred from an upper memory register to one of the α-registers in lower memory, and DSC, in which the addressing gates must return to L1 after the pattern word has been read.

For both of these instructions, then, it is possible to make EXT S01 change from 0 (when the instruction itself is read and the indirect address register is determined) to 1 (when the operand is fetched from upper memory), and back to 0 (when the operand is stored or when the horizontal co-ordinate is accessed).

The conditions just described are tested using worst-case addressing schemes to provide the most rigorous conditions possible. The two instructions, SET and DSC, are tested separately.

1.1 Part 1 SET

Using the instruction, SET 1, data is transferred from upper memory bank register L3756 to the α-register L1. All combinations of memory fields are used; in every field, the lower bank is the third one, and the upper bank is the fourth (see table T25-1). All possible values of the operand are used. The instruction is executed from L21. L22 contains the effective address (3756) of the operand. After the LINC program is executed, C(Y), C(ALPHA), and C(L22) are tested.

1.2 Part 2 DSC

Using the instruction, DSC i 17, the same bank combinations as for Part 1 are tested. In every case, the pattern word is kept in upper memory register L3760; the β-register is L17. During the test, a bar is displayed across the middle of the scope on channel 0. This bar consists of the repeated display of the pattern 7777, starting at co-ordinates H=000, V=000. The vertical co-ordinate remains

Description continued on page T25-4

2. SPECIAL SWITCH SETTINGS

LS 6, 7, 8	Number of additional PDP-8 memory fields

3. PRINTOUT HEADINGS

3.1 Part 1

```
TEST 25A:  EXT S01 (SET).
     C(ALF)   C(Y)   C(L22)
```

C(ALF)	Contents of L1 after execution
C(Y)	Contents of L3760
C(L22)	Effective address

3.2 Part 2

```
TEST 25B:  EXT S01 (DSC).
     HORIZ   VERT   PATRN   C(BETA)
```

HORIZ	C(L1) after display
VERT	C(A) after display
PATRN	C(Y) after display
C(BETA)	Effective address

4. NON-ERROR MESSAGES

None

5. SPECIAL HALTS

None

6. LINC PROGRAM SEQUENCES

6.1 Part 1 SET

```
L21,            SET 1
L22,            3756            /Points to operands
L23,            HLT

L0001,          n               /α-register holds operand after test
L3756,          n               /Operand stored here
```

6.2 Part 2 DSC

```
L1760,          DSC i 17
L1761,          HLT

L0001,          H-coordinate
L0017,          3757            /Effective address before indexing becomes 3760
                                /afterward
L3760,          7777            /Display pattern
```

7. EXECUTION TIME

Varies with number of additional fields.
With 1 field, 4 sec

8. ERROR STOPS AND INFORMATION

Error Message	C(AC)	Data Printed	Program Tag
INTS nnnn	No halt	INTS status	None
	A spurious LINC interrupt occurred. The octal number nnnn represents the interrupt status as read into the PDP-8 AC. See table at the front of this manual to determine the cause of the interrupt. The program does not stop.		
SET	2500	All entries, 25A	E2500
	The test looks for bad addressing, which can result in an incorrect transfer of data, or in failure to access the operand. In either case, C(ALF) will be wrong. The usual checks for correct effective address (C(L22)) and source register (C(Y)) are made.		
DSC I	2510	All entries, 25B	E2510
	The same data are tested here as in Test 21, part B, which thoroughly examines the DSC instruction. Incorrect addressing can have several effects: a. Wrong memory register accessed. Since all memory registers not containing program or data contain 0000, this will probably result in a pattern of 0000 being displayed, which will appear very briefly as a dropped character in the displayed bar. b. Wrong H-co-ordinate accessed. This will also result in a value of 0000, thus causing an extra-bright character to appear at the left of the screen. Because the tested items are all detectable internally, no attempt has been made to make them easily discernible on the scope face.		

Description continued from page T25-1

the same; the horizontal co-ordinate continues to be incremented. For each combination of memory banks, 128 half-character patterns are displayed (this is equivalent to two sweeps across the face of the scope). The entire DSC test is performed eight times.

After each execution of the LINC program, the contents of L1 (HORIZ), L17 (C(BETA)), L3760 (PATRN), and A (VERT), are examined.

TABLE T25-1 MEMORY BANKS USED IN TEST 25

Lower Bank	Upper Bank
02	03
06	07
12	13
16	17
22	23
26	27
32	33
36	37

TEST 26 - LINCTAPE A

Motion Flip-Flops and Mark Window

1. DESCRIPTION

At each point in the program where a new LINCtape IOT is executed, a Q loop is provided for tight scoping. The Q represents a NOP which can be replaced by a JMP to a point just before the ICON is executed, thus providing a very short program loop.

1.1 MOTN Flip-Flops

An attempt is made to clear the MOTN flip-flops by executing the PDP-8 IOT instruction, ICON-0. The states of MOTN1 and MOTN0 are then read into the AC using INTS (IOT 6147), and tested to see that both have been set to zero.

Each MOTN flip-flop is now tested to see if it can be set to 1, using ICON-1. Each flip-flop is tested individually; then both are set to 1, using ICON-1 twice in succession (first with AC_0=0, then with AC_0=1), to see if the turn-around configuration can be set properly.

Next, the program attempts to clear the MOTN bits by deselecting the LINC processor, thus generating a PWR CLR pulse. Again, the MOTN flip-flops are read and examined using INTS.

Finally, the action of ICON-2 in changing the state of the unit flip-flop, U0, is tested. The flip-flop is set, then MOTN1 and MOTN0 are both set to 1. Then U0 is changed, and the MOTN bits are tested to see that they have been cleared by the execution of ICON-2. The same test of the MOTN bits is performed, this time with U0 changing from 0 to 1.

1.2 Mark Window Movement Test

This test is intended to ascertain only that the window moves when the tape is in motion; except for detecting the End Mark, no attempt is made to see that the window shifts correctly.

The LINC is deselected to clear the tape control. After waiting until the tape drive has had a chance to come to a full stop, the program sets the tape into motion, moving forward. The End Mark status bit, Mark Window, and MOTN flip-flops are now monitored continuously. If at any time both MOTN bits become 0, the message

MOTN CLEARED

is printed.

The window is examined for the first occurrence of a shift left. The time of this shift is checked. If it is less than 80 msec, the message

ACIP

is printed.

Description continued on page T26-4

2. SPECIAL SWITCH SETTINGS

None

3. PRINTOUT HEADINGS

3.1 Motion Bits Test

```
TEST 26A:  LINCTAPE MOTN
     MOTN1      MOTN0      U0
```

MOTN1 MOTN0	State of the LINCtape motion flip-flops
U0	State of the tape unit selector flip-flop

3.2 Mark Window Test

```
TEST 26B:  MARK WINDOW
     TIME      WNDO
```

TIME	Time in msec between AC⟶MOTN and first movement of the Mark Window
WNDO	Contents of the Mark Window flip-flops

4. NON-ERROR MESSAGES

None

5. SPECIAL HALTS

For errors 2611 and 2612, C(PC)=1524. Press PDP-8 CONT to recover.

6. LINC PROGRAM SEQUENCES

None

7. EXECUTION TIME

about 6 seconds.

8. ERROR STOPS AND INFORMATION

Error Message	C(AC)	Data Printed	Program Tag
INTS nnnn	No halt	INTS status	None
		A spurious LINC interrupt occurred. The octal number nnnn represents the interrupt status as read into the PDP-8 AC. See table at the front of this manual to determine the cause of the interrupt. The program does not stop.	
ZERO	2601	MOTN1, MOTN0	
		An attempt to clear the LINCtape MOTN flip-flops by executing ICON-0 failed. Actual and correct states of both flip-flops are printed.	
AC	2602	MOTN1, MOTN0	
		An attempt to set the LINCtape MOTN flip-flops by executing ICON-1 failed.	
PWRCL	2603	MOTN1, MOTN0	
		An attempt to clear the MOTN flip-flops by generating a PWR CLR pulse failed.	
U0	2604	MOTN1, MOTN0, U0	
		Changing the state of the unit flip-flop, U0, failed to clear the MOTN bits. The test is performed with U0 changing from 1 to 0, then from 0 to 1.	
ENDMK	2610	WNDO	
		The End Mark status bit was set, indicating that the Mark Window was decoded incorrectly. The contents of the window are printed.	
WINDOW DIDN'T MOVE	2611	None	
		After setting the tape in motion, no change was detected in the contents of the Mark Window within 500 msec.	
MOTN CLEARED	2612	None	
		After executing a command to start the tape drive (ICON-1), the MOTN bits remained clear, indicating that the drive failed to start moving.	
ACIP	2620	TIME	
		The timing between the starting of the LINCtape and the first change appearing in the Mark Window was not correct. Use cycle repeat and synchronize on ACIP to adjust the timing.	

Description continued from page T26-1

If the contents of the Mark Window do not change within about 500 msec, the message

WINDOW DIDN'T MOVE

is printed.

If, while the tape is moving, the End Mark status bit becomes set to 1, the message

ENDMK

is printed, followed by the contents of the Mark Window. This is an indication of false decoding of the window.

After testing in the forward direction, the program reverses the tape and performs the same tests in the reverse direction.

TEST 27 - LINCTAPE B

The Mark Window

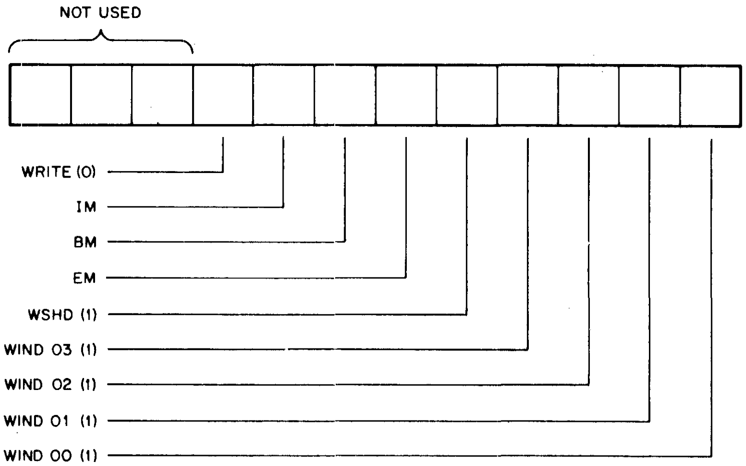
1. DESCRIPTION

1.1 Test 27A: CLOSE Pulse

This test checks to see that PWR CLR and AC ⟶ MOTN inputs each generate a CLOSE pulse, which should clear the Window Shade (WSHD) flip-flop, preset the Mark Window to the state 1110, and clear the Key Mark decoder outputs.

With tape stopped, a PWR CLR pulse is generated by deselecting the LINC (ICON-11). The Mark Window, shade, and Key Marks are read into the AC using ITAC. (The WRITE flip-flop, having been disabled by the PWR CLR, should be represented by a 1 in AC3.) The contents of the AC are masked to isolate the pertinent bits, and examined for correct state. The bit correspondence is shown in table T27-1.

TABLE T27-1 LINCTAPE CONTROL BIT ASSIGNMENTS



The states of these bits are read into the PDP-8 AC using the interface IOT instruction ITAC, operation code 6157.

The tape is then set in motion forward. The window is monitored, and after 200 changes of its contents (indicating that the tape is up to speed) a second ICON-1 is given. The window and status bits are examined, to see if the AC⟶MOTN pulse generated by ICON-1 also generated a CLOSE pulse.

Description continued on page T27-4

2. SPECIAL SWITCH SETTINGS

None

3. PRINTOUT HEADINGS

3.1 CLOSE Pulse Test

```
TEST 27A:  CLOSING THE WINDOW
     WNDO      PWR CLR     AC>MOTN
```

WNDO	Contents of the Mark Window
PWR CLR	Indicates whether this pulse was given
AC>MOTN	Indicates whether this pulse was given

3.2 Window, Shade, and Key Marks

```
TEST 27B:  WNDO, WSHD, KEY MARKS
     WNDO      EM      IM      BM
```

WNDO	Contents of the Mark Window
EM, IM, BM	Respectively, the states of end mark, interblock mark, and block number mark status bits.

4. NON-ERROR MESSAGES

None

5. SPECIAL HALTS

None

6. LINC PROGRAM SEQUENCES

None

7. EXECUTION TIME

about 45 seconds.

8. ERROR STOPS AND INFORMATION

Error Message	C(AC)	Data Printed	Program Tag
INTS nnnn	No halt	INTS status	None
	A spurious LINC interrupt occurred. The octal number nnnn represents the interrupt status as read into the PDP-8 AC. See table at the front of this manual to determine the cause of the interrupt. The program does not stop.		
TTOK FAILED	No halt (2701)	None	
	After executing 0→MOTN, the Mark Window was not preset to the state 1110 within 500 msec. Check $\overline{\text{TTOK}}$ input to the CLOSE pulse generator.		
TT1'S CONTINUE	No halt (2702)	None	
	After executing 0→MOTN, the contents of the Mark Window continue to shift, indicating that TT1 pulses have not been inhibited. Check the MOTN gates on the TT1 enabler.		
CLOSE	2700	WNDO, PWR CLR, AC>MOTN	
	An attempt was made to close the Mark Window, but the window was not correctly preset to the state 1110. The contents of the Window, Shade, and Key Marks are printed. (See table T27-1 for PDP-8 AC bit assignments.) If the CLOSE was attempted by means of PWR CLR, the "C" entry under PWR CLR will be 1. Similarly, if the attempt was made by executing AC→MOTN, that column's "C" entry will be 1.		
SHADE	2711	WNDO	
	The tape is set into motion. After the Mark Window has had a chance to shift contents 200 times, the SHADE flip-flop is tested to see if it remained set during the changes of the window. If the shade failed to be set, or if it was cleared while the tape was in motion, this message is printed. C(Window) printed.		
SHIFT	2712	WNDO	
	The window was read, the next succeeding value of the contents of the window was computed, and the next contents were read. If the calculated and actual values failed to match, this message is printed.		
KEYMK	2710	All entries, 27B	
	The window was read, but the corresponding Key Mark status bit was not set. The contents of the Mark Window and the state of the particular Key Mark being tested are printed.		

Description continued from page T27-1

With tape in motion, PWR CLR is generated again by deselecting the LINC, and the action of CLOSE is examined once more. Thus, the action of CLOSE is tested under three conditions, as follows:

a. Tape stopped, PWR CLR given

b. Tape stopped, AC→MOTN given

c. Tape moving, PWR CLR given

During the third part of the test, after the PWR CLR has been given, but while the drive is still in motion, the contents of the Mark Window are continuously monitored. If the bits continue to shift, it is assumed that the TT1 pulse was not inhibited, perhaps because the MOTN flip-flops were not cleared. In this case, the error message

TT1'S CONTINUE

is printed.

If the Mark Window is not preset to the state 1110 within 500 msec of the time when PWR CLR was given, it is assumed that the $\overline{\text{TTOK}}$ level (which exists whenever the tape drive is not at the proper speed) has failed to trigger the CLOSE pulse generator. The error message

TTOK FAILED

is printed.

1.2 Test 27B: Mark Window, Shade, and Key Marks

As the tape passes over the read head, bits from the mark track are transferred to the least significant bit of the Mark Window (WND00). At the same time, the previous contents of the four window bits are shifted left one place to make room for the new bit. A bit shifted out of WND03 will cause the Window Shade (WSHD) flip-flop to be set to one. Once WSHD has been set, it should remain in the 1-state until cleared by a CLOSE pulse.

For this test, the tape is set in motion. The Mark Window is monitored, and after 200 shifts (to insure that the tape is fully up to speed), the shade is examined to see if it has been set. From this point, the window and shade are monitored continuously.

If at any point, the contents of WSHD become zero, the error diagnostic "SHADE" is printed, with the contents of the Mark Window at that point.

At the same time that WSHD is checked, the contents of the window are examined to see that the shift is working properly. C(WND) are read, the value shifted left once, and stored. At the next window change, C(WND) are read again. The value of the new low-order bit, that is, C(WND00), is added to the (shifted) previous C(WND). The resulting value is then compared with the new C(WND); if they do not match, the diagnostic, "SHIFT", is printed, followed by the actual and correct values of C(WND).

The contents of the Mark Window are continuously compared with the states of the Key Mark status bits (EM, IM, and BM) to see that the window decoders are working properly. The decoding is as follows:

Key Mark	Window
EM	0000
IM	1111
BM	1110

TEST 30 - LINCTAPE C

Motion Control and Tape Interrupt

1. DESCRIPTION

1.1 Motion Control Test

With all modes disabled (no mode), the tape is set into motion going backwards. When the Mark Window indicates that the end zone has been reached (at the front of the tape), the MOTN flip-flops are tested to see that they have been cleared.

With both MOTN bits set to 1, that is, with the tape in turn-around mode, the window is monitored until an interblock mark is encountered, at which time the MOTN flip-flops are examined to see if they have been cleared. This provides a check on the condition, IM·TT2·SRCH(0).

Next, the tape is set into motion backwards in search mode. When the end zone at the front of tape is encountered, the MOTN flip-flops are tested to see if they have been complemented, thus reversing the motion of the tape.

1.2 Tape Interrupt Test

With all modes disabled, the tape is set into motion. As the window changes, the tape interrupt status is monitored; there should be no interrupts while tape is operating in this manner.

Next, the tape is placed in search mode, and the interrupt status checked to see that a tape interrupt occurs once at each block mark, and at no time between block marks.

Finally, the tape is placed in block mode, and the interrupts examined to see that one occurs every time a data word is assembled, and at each block mark. For any block, there should be 414_8 interrupts from block mark to block mark, inclusive.

1.3 Interrupt Flip-Flop Test

When the last interrupt is received while counting interrupts, PWR CLR is generated, and the interrupt flip-flop checked to see if it was cleared.

2. SPECIAL SWITCH SETTINGS

None

3. PRINTOUT HEADINGS

3.1 Motion Control

```
TEST 30A:  MOTION CONTROL
     SRCH   IM   EM   MOTN1   MOTN0
```

SRCH	Contents of the SRCH flip-flop
IM, EM	Key mark status bits (see section 3.2)
MOTN1, MOTN0	Contents of the tape motion flip-flops

3.2 Interrupt Check

```
TEST 30B:  INTERRUPT CHECK
     BM      SRCH      INTRPTS
```

BM	Block mark status bit
SRCH	Contents of the SRCH flip-flop
INTRPTS	Number of interrupts between block marks.

3.3 Interrupt Clear

```
TEST 30C:  INTERRUPT CLEAR
     PWR CLR
```

4. NON-ERROR MESSAGES

None

5. SPECIAL HALTS

None

6. LINC PROGRAM SEQUENCES

None

7. EXECUTION TIME

about 45 seconds.

8. ERROR STOPS AND INFORMATION

Error Message	C(AC)	Data Printed	Program Tag
INTS nnnn	No halt	INTS status	None
		A spurious LINC interrupt occurred. The octal number nnnn represents the interrupt status as read into the PDP-8 AC. See table at the front of this manual to determine the cause of the interrupt. The program does not stop.	
CTRL	3000	All entries, 30A	
		Motion Control is failing in one of the following ways: a) With tape moving backwards and SRCH not set, the MOTN flip-flops were not cleared when the front end zone was encountered. b) With tape moving backwards, and SRCH set to 1, the MOTN flip-flops were not complemented (thus reversing the tape direction) when the front end zone was encountered. In both cases, the states of the SRCH flip-flop and EM and IM key marks are tested.	
TPINT	3010	All entries, 30B	
		The number of tape interrupts occurring between the two block marks of a given block is in error. Three conditions are tested. a) With no mode, the tape is started. No interrupts should occur between block marks. b) In search mode, there should be two interrupts between block marks. c) In block mode (Read), there should be 414_8 interrupts. In every case, the block mark status and the contents of SRCH are printed.	
PWR CLR	3011	PWR CLR	
		Power clear doesn't reset the interrupt flip-flop.	

TEST 31 - LINCTAPE D

Search and Block Modes

1. DESCRIPTION

The SRCH and BLOCK flip-flops are tested for their ability to clear (after being set) when the appropriate pulses are generated.

1.1 Search Mode

After the tape is set in motion, the search mode is established by executing ICON-2 (SET SEARCH). Immediately after this, an attempt is made to clear the SRCH flip-flop by generating an OFFSRCH pulse, using ICON-4. The effectiveness of the pulse is tested by waiting for tape interrupts; if the SRCH flip-flop was cleared successfully, there will be no interrupts as the tape continues to move. If an interrupt does occur, the error message

SRCH

is printed, followed by the state of the SRCH flip-flop. The "C" entry under OFFSRCH will be 1 .

Next, the ability of ONBLK to clear SRCH is tested. This can only be done indirectly, as follows:

SET SRCH is given (ICON-2)
ON BLOCK is given (ICON-3)
OFF WRITE is given (ICON-6)

If the control is working properly, SET SRCH will have set the SRCH flip-flop, and placed the tape in a condition whereby an interrupr would be caused by the appearance of a block number. ON BLOCK will have cleared the SRCH flip-flop, and set up the condition for a tape interrupt every word. Finally, OFF WRITE will have cleared the BLOCK flip-flop, thus leaving tape in motion, but now with no interrupts to be expected. The program tests for this last condition; if an interrupt does occur, the program infers that the SRCH flip-flop was not in fact cleared.

If an error occurs, the diagnostic message is the same as for OFF-SRCH but the ONBLK column will have 1 as the "C" entry.

1.2 Block Mode

The BLOCK flip-flop is tested in a similar fashion. It is set (after tape has been started) using ICON-3; the program then attempts to clear the flip-flop using, successively, SETSRCH and

Description continued on page T31-4

2. SPECIAL SWITCH SETTINGS

None

3. PRINTOUT HEADINGS

3.1 Search Mode Test

```
TEST 31A:  SEARCH MODE
    ONBLK      OFFSRCH
```

ONBLK, OFFSRCH   Pulses used in test

3.2 Block Mode Test

```
TEST 31B:  BLOCK MODE
    OFFWRT      SETSRCH
```

OFFWRT, SETSRCH   Pulses used in test

3.3 Block Number Reading Test

```
TEST 31C:  BLOCK NUMBERS
```

The six columns contain six consecutive block numbers from tape.

4. NON-ERROR MESSAGES

None

5. SPECIAL HALTS

None

6. LINC PROGRAM SEQUENCES

None

7. EXECUTION TIME

about 35 seconds.

8. ERROR STOPS AND INFORMATION

Error Message	C(AC)	Data Printed	Program Tag
INTS nnnn	No halt	INTS status	None
		A spurious LINC interrupt occurred. The octal number nnnn represents the interrupt status as read into the PDP-8 AC. See table at the front of this manual to determine the cause of the interrupt. The program does not stop.	
SRCH	3100	All entries, 31A	
		An attempt to clear the SRCH flip-flop by means of OFFSRCH or ONBLK pulses failed. A "C" entry of 1 is printed under the pulse used. The error is inferred from the occurrence of a tape interrupt when a block mark is read.	
BLOCK	3110	All entries, 31B	
		An attempt to clear the BLOCK flip-flop failed. A "C" entry of 1 is printed under the pulse used (in this case, either SETSRCH or OFFWRT) to clear the flip-flop. The error is inferred from the occurrence of tape interrupts while the data track is passing the tape heads.	
BLKNO	3120	All columns, 31C	
		An error occurred while reading block numbers from consecutive blocks. Six numbers, including those in error, are printed.	

Description continued from page T31-1

OFFWRT pulses, which are generated by executing ICON-2 and ICON-6, respectively. In case of an error, the diagnostic message

BLOCK

is printed, with 1 as the "C" entry under the pertinent pulse.

1.3 Block Number Reading Test

With tape moving forward in search mode, six consecutive block numbers are read and compared with correct values calculated in advance. If an error in reading occurs for any given set of six, the six correct numbers are printed, along with the actual values of the block or blocks in error. The diagnostic is identified by the message

BLKNO

Approximately 100 block numbers are read.

The same test is then performed with the tape moving backwards, to check the ability of the processor to read reverse block numbers. The diagnostic and printout are the same, except that numbers will appear in decreasing value.

TEST 32 - LINCTAPE E

Write Mode

1. DESCRIPTION

The WRITE flip-flop is tested for its ability to be set and then cleared by various pulses. The entire test is conducted in the block number and check mark area of the first block on tape. This is block 0010(-10), one of the ten blocks which precede the first usable block on tape.

The tape is put in motion in search mode, and the program attempts to position the tape at the beginning of block -10. If the block number cannot be found, the error message

BLOCK 0010 MISSING

is printed. The remainder of the test is bypassed, and control returns directly to the Monitor. (If either repeat option is in effect, the position test is repeated.)

If the tape is positioned successfully, it is set moving forward, still in search mode. When the block number is once again assembled in the buffer, a tape interrupt should occur, at which time the write mode is enabled by executing ICON-5. The WRITE flip-flop is examined (using ITAC to read its status into the AC). If it has not been set, the error message

WRITE NOT SET

is printed. As before, the remainder of the test is bypassed, and control returns to the Monitor unless one of the repeat options is in effect.

If the WRITE flip-flop is set successfully, attempts are made to clear it using various pulses. After each successful attempt, the flip-flop is set again (and tested) and a new pulse is tested. In order, the following pulses are checked for their obility to clear the WRITE flip-flop:

PWR CLR	(using ICON-11, DESELECT LINC)
OFFWRT	(using ICON-6, OFF WRITE)
SEARCH	(using ICON-2, SET SEARCH)
0 → MOTN	(using ICON-0)
CLOSE	(using ICON-1, AC → MOTN)

Finally, the check mark is tested to see if it will clear the WRITE flip-flop when the mark appears in the window. With the tape moving forward in write mode, the window is monitored until the check mark appears. For insurance, the mark is read twice to be sure that it is really there; the WRITE flip-flop status is then examined to see if the flip-flop was cleared.

If any of the above pulses fail to clear the WRITE flip-flop, the error message

NTCLR

is printed; the "C" entry under the pertinent pulse will be 1.

2. SPECIAL SWITCH SETTINGS

None

3. PRINTOUT HEADING

```
TEST 32:  WRITE FLIP-FLOP
     PWR CLR    OFFWRT    SEARCH     0  MOTN    CLOSE    CKMK
```

Each column head identifies a pulse used to clear the WRITE flip-flop

4. NON-ERROR MESSAGES

None

5. SPECIAL HALTS

None

6. LINC PROGRAM SEQUENCES

None

7. EXECUTION TIME

Execution time of test 32 = 25 seconds.

8. ERROR STOPS AND INFORMATION

Error Message	C(AC)	Data Printed	Program Tag
INTS nnnn	No halt	INTS status	None
	A spurious LINC interrupt occurred. The octal number nnnn represents the interrupt status as read into the PDP-8 AC. See table at the front of this manual to determine the cause of the interrupt. The program does not stop.		
BLOCK 0010 MISSING	No halt (3201)	None	
	If the program cannot position the tape at the beginning of block 0010, this message is printed. There is no halt; control returns to the Monitor.		
WRITE NOT SET	No halt (3202)	None	
	If the WRITE flip-flop cannot be set, this message is printed. There is no halt; control returns to the Monitor.		
NTCLR	3200	All entries	
	If a given pulse does not clear the WRITE flip-flop, the "C" entry under the column head identifying that pulse will be 1.		

TEST 33 - LINCTAPE F

Read/Write and Timing

1. DESCRIPTION

A data pattern is generated in a block-sized buffer. The program then searches for a free tape block; while doing so, it also monitors tape speed by measuring the time it takes to pass from the front block mark to the reverse one, and examines consecutive block marks to be certain that they are in the correct sequence.

When the desired block is found, the buffer of data is written, then read into a second buffer. The contents of the original and final buffers are compared word by word, and errors are accumulated in the Monitor's data tables. When six errors have been detected, the data are printed, identified by the diagnostic message

CMPAR

When the test resumes, the buffer examination continues until the end of the block is reached. If fewer than six errors are detected, the data will be printed when the buffer comparison is completed.

If the cycle repeat option is in effect, the same data pattern will be written and compared. Otherwise, the program will generate a new pattern. In all, four data patterns are used for testing:

a. All zeros

b. All ones

c. An incrementing pattern, in which the successive words of the block are 0101, 0202, 0303, 0404, etc.

In one complete block, the pattern is repeated four times.

d. A random sequence of numbers.

The generated (correct) data may be found in a block starting in PDP-8 register 2000. The data read from tape (actual) may be found in a block starting in PDP-8 register 3000.

If, while searching for a block, the tape requires longer than 250 msec to travel from the front to the reverse block mark, the error message

TOO SLOW

is printed. There is no halt, and no data is printed.

If, while searching for a block, two consecutive block numbers are not in the correct sequence, the diagnostic message

BNSEQ

is printed, followed by the two numbers in question.

2. SPECIAL SWITCH SETTINGS

None

3. PRINTOUT HEADINGS

```
TEST 33:  BLOCK MODE
     1ST   2ND   3RD   4TH   5TH   6TH
```

1ST	Previous block number, or first data word
2ND	This block number, or second data word
3RD, 4TH, 5TH, 6TH	Successive data words in error

4. NON-ERROR MESSAGES

None

5. SPECIAL HALTS

None

6. LINC PROGRAM SEQUENCES

None

7. EXECUTION TIME

5–25 seconds.

8. ERROR STOPS AND INFORMATION

<table>
<tr><th>Error Message</th><th>C(AC)</th><th>Data Printed</th><th>Program Tag</th></tr>
<tr><td>INTS nnnn</td><td>No halt</td><td>INTS status</td><td>None</td></tr>
<tr><td></td><td colspan="3">A spurious LINC interrupt occurred. The octal number nnnn represents the interrupt status as read into the PDP-8 AC. See table at the front of this manual to determine the cause of the interrupt. The program does not stop.</td></tr>
<tr><td>TOO SLOW</td><td>No halt
(3301)</td><td>None</td><td></td></tr>
<tr><td></td><td colspan="3">The tape slowed down or stopped during the reading of a block. The diagnostic is printed if, after reading a block number, more than 250 msec passes before the appearance of the same number at the other end of the block.</td></tr>
<tr><td>BNSEQ</td><td>3300</td><td>1ST, 2ND</td><td></td></tr>
<tr><td></td><td colspan="3">Two consecutive block numbers were read, but the numbers were out of sequence.</td></tr>
<tr><td>CMPAR</td><td>3310</td><td>Six data items</td><td></td></tr>
<tr><td></td><td colspan="3">Six words from tape are compared with the corresponding six words in memory. Those values which do not match are printed.</td></tr>
</table>

TEST 50 - EPILOG

Tests KST, Z-L non-disturbance, and SNS.

NOTE: The participation of the operator is required for the SNS test. See description for instructions. RS 3 must be up for this test to be executed.

1. DESCRIPTION

1.1 KST Test

The KST instruction is tested 4096 times for correct skipping when the KST flip-flop is not set. Using the IACF instruction, the flip-flop is then set, and KST is tested another 4096 times. In all cases, the P-register is examined after each execution to determine the state of the skip indicators; if an error occurs, actual and correct values of P are printed along with the skip indicators as described in section 6.

1.2 Z-L Non-Disturbance Test

This is a general test designed to pick up random errors in the operation of those LINC instructions which should not disturb the Z-register or Link during execution. Each instruction is tested for all possible values of C(Z) and both possible states of the Link. To set the Link, C(Z) are placed in A before execution. The first instruction in the LINC sequence is ROL i 1, which puts $C(A_0)$ in the Link. Then a string of eight identical LINC instructions is executed; this is the instruction being tested. Finally, another ROL i 1 puts the C(L) into A_{11}, where it is accessible from the PDP-8. A list of the instructions tested, with their octal codes, is given in table T50-1.

1.3 SNS Test

This test is executed only when RS 3 is up. There is no data printout, and no internal checking by a PDP-8 program; the test is entirely visual. A self-contained LINC program in registers L20-L67 (see page T50-4) examines the states of the Sense Switches and sets the Relay Lights accordingly. The light above each switch in the up position is lit; all others are not. Between each pass, there is a delay of about 160 msec, so that if any switch is failing intermittently, the trouble will appear as a flashing light.

1.3.1 Operator - Actuate the Sense Switches and observe the behavior of the Relay Lights. To return to the main test program for the grand final halt, strike any signal-generating key on the ASR keyboard.

Description continued on page T50-4

2. SPECIAL SWITCH SETTINGS

RS 3	down	Skip SNS test and grand final halt. Return to Monitor for normal halt or continuation.
	up	Do SNS test; on return, stop at grand final halt.

3. PRINTOUT HEADINGS

3.1 KST Test

```
TEST 50A:  KST
     C(P)   SKIP?
```

C(P)	Contents of P
SKIP?	Indicates the occurrence (or none) of a skip.

3.2 Z-L Non-Disturbance Test

```
TEST 50B:  Z-L NON-DISTURBANCE
     INSTR   C(Z)   C(L)
```

INSTR	Octal code of LINC instruction being tested
C(Z),C(L)	Contents of Z and L, respectively

4. NON-ERROR MESSAGES

None

5. SPECIAL HALT

Grand Final Halt (upon return from SNS test)

C(PC)=1412 C(AC)=7777 C(A), C(B), C(S): "E-N-D"

Recovery: Press PDP-8 CONT to return to Monitor.

6. LINC PROGRAM SEQUENCES

6.1 KST Test

```
L1,            KST
L2,            HLT
L3,            HLT
```

6.2 Z-L Non-Disturbance Test

```
L1,            ROL I 1        /Put C(A0) in Link
L2-L11,        (*)            /*String of 8 identical LINC instructions being tested
L12,           ROL I 1        /Put C(L) in A11.
L13,           HLT
```

6.3 SNS Test

See page T50-4

7. EXECUTION TIME

38 sec (excluding SNS test).

8. ERROR STOPS AND INFORMATION

Error Message	C(AC)	Data Printed	Program Tag
INTS nnnn	No halt	INTS status	None
	A spurious LINC interrupt occurred. The octal number nnnn represents the interrupt status as read into the PDP-8 AC. See table at the front of this manual to determine the cause of the interrupt. The program does not stop.		
KST	5000	C(P), SKIP?	E5000
	Skip error. If KST skipped when KST FF was clear, the A and C entries under SKIP? are 1 and 0, respectively. If KST failed to skip when the flip-flop was set, the entries are 0 and 1. C(P) are also printed.		
Z-L	5010	All entries, 50B	E5010
	If either the Z-register or Link was disturbed during the execution of an instruction that should have left it alone, the pertinent data are printed. The octal code of the LINC instruction being tested is printed under INSTR for information. Table 50-1 is a list of the LINC instructions and their octal codes.		

TABLE T50-1 LINC INSTRUCTIONS AND OPERATION CODES, TEST 50B

Instr.	Code	Instr.	Code	Instr.	Code
ZTA	0005	ROL 14	0254	STH 15	1355
ATR	0014	ROR 14	0314	SHD 15	1415
RTA	0015	SKP	0467	SAE 15	1455
NOP	0016	LDA i 15	1035	SRO 15	1515
COM	0017	STA 15	1055	BCL 15	1555
SET 15	0055	ADA 15	1115	BSE 15	1615
DIS 15	0155	ADM 15	1155	BCO 15	1655
XSK 15	0215	LDH 15	1315	ADD 0	2000
				STC 0	4000
				JMP 12	6012

Description continued from page T50-1

1.3.2 SNS Test (This is entirely self-contained.)

```
L20,          CLR          /Clear A, Z, L
L21,          SNS i 0      /Is Switch 0 set?
L22,          ADD 50       /Yes.  Get pattern bit
L23,          SNS i 1      /Each switch is tested the same way
L24,          ADD 51
L25,          SNS i 2
L26,          ADD 52
L27,          SNS i 3
L30,          ADD 53
L31,          SNS i 4
L32,          ADD 54
L33,          SNS i 5
L34,          ADD 55
L35,          ATR          /Relay light pattern to lights.
L36,          CLR
L37,          LDA i 0      /Slow counter for time delay
L40,          7760         /Slow count of ten cycles.
L41,          STC 2        /Store count in L2.
L42,          STC 1        /Start fast count at 0000.
L43,          XSK i 1      /Index fast counter.  Cycle finished?
L44,          JMP .-1      /No.
L45,          XSK i 2      /Yes. Finished ten cycles?
L46,          JMP .-4      /No.
L47,          JMP 20       /Yes.  Go round again.

L50,          0040         /Pattern bit for Relay Light 0 (SW 0)
L51,          0020         /Pattern bit for SW1
L52,          0010         /SW2 pattern bit
L53,          0004         /SW3
L54,          0002         /SW4
L55,          0001         /SW5

L01,          fast count
L02,          slow count
```

TEST 6 - ROTATE CLASS

Tests ROL, ROR, SCR and LZE

1. DESCRIPTION

For each of the six possible rotate class instructions (ROL, ROL I, ROR, ROR I, SCR, and SCR I), both selected numbers and random numbers are used. Each number is placed in the A-register and rotated N places, where N varies from 0 to 17_8. The selected numbers are shown in section 1.1.1. The random numbers are generated by the same sequence used for the ADD test (q.v.), starting with a seed of 5470.

Before the LINC sequence is executed, the L- and Z-registers are cleared. Afterward, the A- and Z-registers are tested for correct contents. The L-register is tested indirectly, by executing an LZE; if the Link is set, 0001 is placed in the A-register. This provides a check of either the LZE instruction or the Link register.

For ROL and ROL I, the Z-register should be clear after execution. For ROL, ROR, and SCR, the Link should be clear after execution.

The LINC sequence for these tests is given in section 6.1.

For the "Vibrating A" test, a string of instructions, alternating ROL I 17 with ROR I 17, with an extra ROL I 17 at the end, is stored in memory. Each of the selected and random numbers is tested; the test is designed primarily to detect marginal failures in the shift gates.

After the preceding tests, ROL and ROR are tested once more, this time with the Link set before execution (all of the previous tests begin with a cleared Link). This test uses a cleared LINC accumulator. The instructions are ROL 17 and ROR 17; in neither case should the Link be cleared. Each instruction is tested 4096 times. The LINC sequence is shown in section 6.3.

1.1.1 Selected Operands Used For Rotate Test -

0000	7777	5252	7070	2634

Sliding 1s

0001	0010	0100	1000
0002	0020	0200	2000
0004	0040	0400	4000

Sliding 0s

7776	7767	7677	6777
7775	7757	7577	5777
7773	7737	7377	3777

2. SPECIAL SWITCH SETTINGS

None

3. PRINTOUT HEADING

```
TEST 6:  ROTATE CLASS INSTRUCTIONS.
    A1    A2    LINK    C(Z)    N
```

A1, A2	Contents of A before and after rotation, respectively.
LINK	Contents of LINC Link after rotation
C(Z)	Contents of Z after rotation
N	Number of places rotated

4. NON-ERROR MESSAGES

None

5. SPECIAL HALTS

None

6. LINC PROGRAM SEQUENCES

6.1 ROL, ROR, SCR Tests

```
L0001,     CLR            /Clear A, Z, L
L0002,     ADD 11         /Put operand in A
L0003,     (Rotate)       /Execute the rotate instruction
L0004,     STC 11         /Store the result
L0005,     LZE            /Is Link set?
L0006,     ADD 10         /Yes.  Put indicator in A
L0007,     HLT            /Return to PDP-8
L0010,     0001           /Link Set indicator
L0011,     (result)       /Contains operand before rotation, result afterward.
```

6.2 VIB Test

Same as for preceding tests, except

```
L0003,     JMP 20         /Jump to VIB string

L0020,     ROL I 17
L0021,     ROR I 17
.....      .....
L0076,     ROL I 17
L0077,     ROR I 17
L0100,     ROL I 17       /Odd-numbered ROL
L0101,     JMP 4          /Return to test sequence
```

6.3 Link-Set Test

Same as for VIB, except

```
L0020,     ROL I 1              /Set Link from A0
L0021,     (ROL 17 or ROR 17)
L0022,     JMP 4                /Return to test sequence
```

7. EXECUTION TIME

45 sec

8. ERROR STOPS AND INFORMATION

Error Message	C(AC)	Data Printed	Program Tag
INTS nnnn	No halt	INTS status	None
	A spurious LINC interrupt occurred. The octal number nnnn represents the interrupt status as read into the PDP-8 AC. See table at the front of this manual to determine the cause of the interrupt. The program does not stop.		
ROL	0600	All entries	RERR
ROL I	0601	All entries	RERR
ROR	0610	All entries	RERR
ROR I	0611	All entries	RERR
SCR	0620	All entries	RERR
SCR I	0621	All entries	RERR
	In each case, the result of the rotation was incorrect. All registers are checked, and erroneous data is printed where applicable; the correct values of data not in error, of A1, and of N, are printed for information. NOTE: If C(L) is incorrect, the error may also lie in the execution of LZE.		
VIB	0630	All entries	RERR
	A sequence of ROL-ROR pairs was executed, and the result was incorrect. The same criteria for the preceding tests apply.		
L ROL	0640	A2, LINK	RERR
L ROR	0641	A2, LINK	RERR
	a. The set Link was cleared during the execution of ROL 17 or ROR 17 with C(A)=0000. b. The A-register was not clear after execution of either instruction.		

APPENDIX 1

SUMMARY OF SUDSY II SWITCH OPTIONS AND OPERATING PROCEDURES

LOADING THE MONITOR

From LINCtape

Lift LOAD

Put 0700 in LS

Put 0013 in RS

Lift DO

Put 4030 in RS

Press PDP-8 LOAD ADD, then START

From Punched Tape

Put 7777 in RS

Press PDP-8 LOAD ADD

Put RS 0 up for ASR,
down for high-speed reader

Press PDP-8 START

LOADING AN INDIVIDUAL TEST

From LINCtape

Put 0100 in RS

Press PDP-8 LOAD ADD

Put RS 0 down

Put RS [illegible] down

Put RS 2 up

Put test number in RS 6-11

Press PDP-8 START

From Punched Tape

Put 0100 in RS

Press PDP-8 LOAD ADD

Put RS 0 up for ASR, down
for high-speed reader

Put RS 1 up

Press PDP-8 START

STARTING AN INDIVIDUAL TEST

Loading from LINCtape

Set Left Switches

Press PDP-8 CONT

Loaded from Punched Tape

Set Left Switches

Put 1000 in RS

Press PDP-8 LOAD ADD, then START

Loading and Running a Continuous Test (From LINCtape Only)

Load the Monitor (see above)

Put 0100 in RS

Press PDP-8 LOAD ADD

Set Left Switches

Put RS 3 up to do SNS test and stop
down to skip SNS test and recycle through test sequence

Press PDP-8 START

SUMMARY OF SUDSY II SWITCH OPTIONS

MONITOR CONTROL (LOADING AND OPERATING OPTIONS)

Switch	Position	Function
RS 0*	up	Read test from ASR
	down	Read test from High-Speed Reader (750 or PC01)
RS 1	up	Read test from punched tape.
	down	Read test from LINCtape.
RS 2**	up	Read test whose number appears in RS 6-11
	down	Read tests in sequence
RS 3	up	Do SNS test (in Epilog); stop at grand final halt.
	down	Skip SNS test; do not stop. (Returns to Monitor.)
RS 6-11	(Test No.)	Octal number of test to be read in (sensed only when RS 2 is up).

*RS 0 is sensed only when RS 1 is up.
**RS 2 is sensed only when RS 1 is down.

PROGRAM CONTROL

LS 11	up	Repeat the current cycle of the test
	down	Do not repeat; proceed to next cycle
LS 10	up	Repeat the entire test
	down	Return to Monitor
LS 9	up	Repeat a section of a test (certain tests only)
	down	Do not repeat the section

ERROR PROCESSING OPTIONS

LS 0	up	Do not ring the bell.
LS 1	up	Do not print error messages and data
LS 2	up	Do not stop after an error
LS 3	up	Ignore LINC interrupt errors
	down	Print LINC interrupt diagnostic and data.

digital

DIGITAL EQUIPMENT CORPORATION • MAYNARD, MASSACHUSETTS