

PDP-1 COMPUTER
ELECTRICAL ENGINEERING DEPARTMENT
M.I.T.
CAMBRIDGE 39, MASSACHUSETTS

PDP-14

PDP-1 DRUM TEST

June 17, 1963

PDP-1 DRUM TEST

Introduction

The PDP-1 Drum Test is a program written in MACRO language, composed of a collection of test routines for checking the operations of the magnetic drum system on the M.I.T. PDP-1 computer. The routines test the drum-core information transfer (read, write, swap) functions and the sequence-break and reference-optimization functions, under a range of conditions. Extensive error print-outs are provided in which the amount of detail is selectable under sense-switch control. The choice of test can be specified at any time from the typewriter, including also the data to be used, drum fields to be tested and other parameters. The details are given in this memorandum.

Acknowledgement

This memo and the program which it describes have been directly taken from the work of W. Feurzeig of Bolt, Beranek and Newman Inc. (Cambridge, Massachusetts).

I am very grateful for the many hours Mr. Feurzeig devoted to helping me with the modifications on his program.

Modes

The program operates under its own steam if the user so desires. In this mode (called the preprogrammed-test-mode), it will execute a large number of tests in sequence without intervention from the user.

At any time the user may usurp control by:

1. putting up sense switch 6 and
2. striking a typewriter key (turning on program flag 1)

If he does this, the program goes into typewriter-control-mode and will wait for further typewriter information to specify the next test routine for its execution.

Following the execution of a test under typewriter-control-mode, the program will continue in preprogrammed-test-mode. In particular, it will sequence to the next test in its list following the one which was interrupted. If sense switch 6 is left up when program flag 1 gets turned on, control will revert to the typewriter again. Thus, if it is desired to do just a single test under typewriter control the user should remember to set sense switch 6 back down after he has been given control.

Tests

There are three kinds of tests defined to the program; the first checks the drum information-transfer operations; the second the optimization and break operations; the third combines transfer and reference-optimization functions.

There are two categories of transfer tests: (1) full block length tests and (2) variable block length tests. The full block length tests check reading, writing and swapping of 4000_g words of information from fixed drum and core origins for all combinations of drum fields. The only input they require is the name of the data sets being used. The variable block length tests check reading, writing and swapping of different size blocks of information originating at changing locations on drum and core, for different (specifiable) combinations of drum fields. There are standard versions of these tests but the user may dictate modifications by providing parameters under typewriter control mode.

There is a single test to check drum optimization and sequence break, i.e., to check the dra and dba instructions executed just prior to and just following a dba (which uses the initial return address) are sufficiently close in value, and that less than a full drum revolution has intervened temporally. This is done for all possible initial dra address returns.

Finally, there are two tests designed to check swapping with optimization at very high speed. These are the 2000_g-word block hot swap test and the one-word hot-swap test. These trade off the detailed checking of each swap operation done by the transfer tests for the much greater speeds possible by checking for errors

only after long time intervals. These tests are useful for spot-checking for troubles. If troubles do show up, it is possible to get more current error information (i.e., to note errors when they occur, rather than to make the detection that they have occurred sometime in the past minute) by then doing transfer tests.

Data

There are eight standard datasets used in the system. The associated single-character-names are indicated here and used for typewriter control. The lower half of core (registers 0-3777) is used for the program, while registers 4000-7777 contain data which is generated by the program and written on and read from all drum fields.

1. all "zeros"--denoted z
2. all "ones"--denoted o.

These two simple datasets may be useful in checking for pick-up and drop-out of bits.

3. "same name" data--denoted s

This is a characteristic value dataset defined as follows: In the drum location specified by (f,a) where f is the (f bit) drum field number and a is the (12 bit) core address of the cell where it was generated (i.e. between 4000 and 7777), the number "f" corresponds to bits 1-5 and "a" corresponds to bits 6-17. For example, same name data on field 13 has 134000 in locations 0 and 4000, 134001 in locations 1 and 4001, ..., 137777 in locations 3777 and 7777.

4. "random" data--denoted r.

This is a special purpose pseudo-random number generator sequence designed for fast generation and easy regeneration on demand. When used in the tests, it produces different data for each field and for the cells within a field and also for each distinct call. At the beginning of a test, the program types out a starting random number which can be used later to get an exact regeneration of the particular sequence for a repetition of the test if this is desired.

- 5-8. checkerboard datasets--denoted c, d, e, f.

These have been useful in checking cores where certain patterns of 1 and 0 bit configurations have exposed strains not

otherwise detected. The datasets used here are the same ones employed in the DEC core diagnostics. They did not prove to be exceptionally interesting for drum error detection during the acceptance testing.

In each of the transfer tests, there are two datasets required. The "background" set with which the drum fields are initially loaded and the "foreground" set which is introduced into the various drum fields during the run of the test. Preprogrammed tests are included for a number of (background, foreground) dataset pairs. In typewriter control mode, the foreground dataset may be specified by the user if he desires; the sets are specified by the one-character-codes given above.

Preprogrammed Tests

The following list enumerates the tests built into the program for the preprogrammed test mode. If the user does not employ typewriter control mode, the program will execute these tests in the listed order. The notational conventions are:

1. nor--denotes the full block-length transfer test. Transfers are first done among all fields in the first half of these fields (locations 0-3777) and then swaps are done among all fields in the second half of these fields (locations 4000-777).
2. var--denotes the variable-block-length transfer test in which each half drum field is involved in swaps of all block lengths up to 4000₈ words. There are 4000₈ swaps with the first half of all fields and then 4000₈ swaps with the second half of all fields.
3. all--denotes the longer variable-block-length test in which each pair of drum fields is swapped for all 4000₈ block sizes (i.e., from one-word to full-block). The test is done in two halves as are the "nor" and "var" tests.
4. dra dba test--denotes the dra and dba instruction test described above.
5. hot swp tst--denotes the 2000₈-word block-length high-speed transfer test.
6. hot one tst--denotes the one-word-block-length high-speed transfer test.
7. o, z, s, r, c, d, e, f--denote the datasets described above.

Preprogrammed Test Sequence

<u>Test name</u>	<u>Background data</u>	<u>Foreground data</u>
nor	r	r
nor	s	s
nor	s	r
nor	r	c
nor	r	d
nor	r	e
nor	r	f
nor	r	s
nor	r	z
nor	r	o
nor	o	z
hot swp tst	r	r
nor	z	o
hot one tst	s	s
var	s	r
dra dba tst		
all	s	r

Typewriter Control Mode Tests

The tests mentioned thus far are callable under typewriter control mode. In addition, typewriter control allows the following:

1. the transfer tests may be called to operate with any of the datasets.
2. the variable-length-transfer tests may be called to operate on particular drum fields and/or between specified block-length values.
3. a particular pseudo-random number sequence may be specified for generation by the random dataset.
4. Any preprogrammed test may be skipped.

For convenience, two additional variable-length-transfer tests are defined for typewriter control usage:

1. One denotes the variable length transfer test in which a single specified drum field is exercised in swaps with all other fields and such that swaps of all block sizes are effected.

2. Two denotes the variable length transfer test in which a pair of specified drum fields are swapped for all block sizes.

When the program is in typewriter control mode, it waits for a string of characters from the typewriter to effect a test specification. The character set for use with typewriter control mode is as follows:

<u>Character</u>	<u>Meaning</u>
n	normal full-block-length transfer test
v	variable-block-length transfer test
1	one field variable-block-length transfer test
2	two field variable-block-length transfer test
b	dra dba test
h	hot swap test (2000 _g -word block-length)
w	hot swap test (one-word-block-length)
→	skip to next test in preprogrammed test sequence
↶	carriage return means "start over", i.e., re-initiate typewriter control mode to wait for new test information. This is the goof control character.
r	"random" dataset
s	"samename" dataset
z	"zeroes" dataset
o	"ones" dataset
c	"checkerboard a" dataset
d	"checkerboard b" dataset
e	"checkerboard c" dataset
f	"checkerboard d" dataset
/	special parameter (see below)
G	execute specified test (not necessary if character → is input).
Δ	spaces are ignored (except see below on calls for test 1 and 2).

Following any of the tests denoted by n, v, 1, and 2, the user may elect to specify any of the datasets r, s, z, o, c, d, e, f, for foreground data. If not, the program will choose the dataset r.

If the special parameter character / is used, it must be followed by a decimal digit. Prior to keying the digit, the test word switches must have been loaded with (right-justified) octal information, as follows:

<u>Special Parameter Character</u>	<u>Test Word Contents (Octal Values)</u>
0	initial drum field to be read
1	increment in drum field to be read
2	final drum field to be read
3	initial drum field to be written
4	increment in drum field to be written
5	final drum field to be written
6	initial block length transfer size
7	increment in block length transfer size
8	final block length transfer size
9	initial pseudo-random number for random dataset
a	advance parameter for hot swaps. The larger the number, the longer the delay between successive swaps.
n	number of cycles of hot swaps between each two-line printout.

Special parameters 0 through 8 are intended for use with the test v, to allow specified subsets of read and write fields and block lengths to be used. Any or all of these parameters may be invoked independently during a single test call. Examples are given below. To specify a test with variation in block lengths from size 1 word in steps of 10₈ words to 2000₈ words, one would set the values of parameters 6, 7, and 8 (successively) to 1, 10, 2000 respectively.

Special parameter 9 is usable with any of n, v, 1, or 2 when the dataset r is the foreground dataset and it is desired to reproduce a particular sequence of random numbers. (The value of the initial random number is output each time a test is executed and thus it is available for recall).

Special parameters a and n are used with either hot swap test.

If the special parameters are not involved, the program will use the following standard values for the associated variables.

<u>Parameters</u>	<u>Standard Value (Octal)</u>
0, 1, 2, 3, 4, 6, 7	1
2, 5	26
8	10000
9	arbitrary value obtained from executing a dra instruction.
a	36
n	320 in 2000g-word hot swaps 300 in one-word hot swap

The field specified for test 1 must be entered on the typewriter as follows: drum field in octal followed by space or comma. The fields specified for test 2 must be entered as follows: first drum field in octal followed by space or comma followed by second drum field followed by space or comma.

If a syntactic error is made in attempting to type in a test call, the program will type out a red question mark, carriage return, excise the (partially completed) call and wait for another test call.

Examples of Typewriter Control Calls

<u>Type-In</u>	<u>Action</u>
nG	Execute test n with all standard parameters
nzG	Execute test n with dataset z for foreground
n/9G	Execute test n with the contents of the test word switches for the starting generator of dataset r.
1 Δ 25, G	Execute test 1 on drum field 25.
2 Δ 7, 11 Δ G	Execute test 2 on fields 7 and 11.
vc/4/2/5 G	Execute test v with dataset c and special parameters 4, 2, and 5 (whose values have been set successively to the contents of the test word switches).
hG	Execute (2000g-word block) hot swap test.
→	Skip to next test in preprogrammed sequence.
vo/1	User signals goof--effects restart.
vn ↩	Syntactic error. Program will signal goof and restart.

Print-Outs

Prior to the execution of any test two lines of identifying information are output. The first line consists in one or two items--the code name of the tests and the code name of the background and foreground datasets being used when this item is relevant. In the case of the dra dba test (see below) this item is not relevant. (In the case of the hot swap tests, the datasets are fixed--random for the 2000_g-word block length test and a partial samename for the one word test.) The second line gives the value of the current initial pseudo-random number (even if the random dataset is not being used), followed by the constant 711711 which serves to identify the former item as the starting random number. This line is not relevant for the dra dba test.

The test code name is one of the following: nor, var, one, two, all, dra dba tst, hot swp tst, hot swp tst one wrd. These correspond in obvious fashion to the tests described above. The dataset code name consists of the code for the background dataset followed by the code for the foreground dataset followed by the character d. Thus, srd specifies samename background data, random foreground data and ssd specifies samename data for both background and foreground.

Dra dba Test Output

For the dra dba test, the second line of print consists of the header information: ert, beg, end, del, int, out.

The header item ert signifies error type code. It has the value 777777 if the sequence break from the dba instruction has not occurred within a drum revolution time. It has the value 6666 in its address part when the contents of the io register after the dba execution is not the same as the contents of the io register before the dba execution, i.e., as the break address. It has the value 02 in the instruction part if the initial dra address and the final dra address are not sufficiently close (within 7 addresses). If no error, ert is zero.

The header items beg and end denote the beginning and final dra values and del is the magnitude of their difference.

Tmo indicates the count obtained in the writing loop after the dba is executed and before the break occurs.

There is a line of printout for each error. Normally, there is no printout under the heading line unless there is an error. However, all output may be forced by setting sense switch 3 down. Also, it should be noted that if sense switch 1 is up, all output, including error information, will be suppressed!

Transfer Test Error Printouts

The error printout information for tests other than dra dba is described now.

If there is at least one word which was transferred with some error, a line of header information will print as follows: ert, exp, obs, del, loc. Ert denotes error type, exp and obs denote the expected and observed value, respectively; del denotes the exclusive-or of exp and obs, i.e., the bits which do not match and loc denotes the location in core of the word in error.

Following the header line, there will normally be one line of printout for each word transferred with error, giving the values of these header variables. Finally, there is a summary header line with the following information: tst, iow, acr, ior, aca, ioa, odd, eve, org, blk. The last two items are printed only for the variable block length tests. The significance of these variables is:

- tst: Test transfer operation and drum fields involved when error occurred.
- iow: Contents of io register prior to dia execution.
- acr: Contents of ac register prior to dcc execution.
- ior: Contents of io register prior to dcc execution.
- aca: Contents of ac register after dcc execution.
- ioa: Contents of io register after dcc execution.
- odd: Total number of odd parity errors during transfer.
- eve: Total number of even parity errors during transfer.
- org: Origin of block in core (address of first word transferred).
- blk: Size of block (number of words transferred).

Following the summary header line, there is a line with the values of the summary variables.

The coding of error types in ert is as follows:

<u>Instruction Part</u>	<u>Meaning</u>
Bits 0-2:	Correspond to bits 0-2 of the io register after execution of a dra instruction following the drum transfer.
Bits 3-5:	
0	The instruction after the executed dcc was skipped.
1	The instruction after the executed dcc was not skipped - an error.

<u>Address Part</u>	<u>Meaning</u>
0001	Word has odd parity error.
0002	Word has even parity error.

The hot swap tests type out a line of starting random number information for each (approximately two minutes long) iteration of fast swaps executed. Following any iteration in which errors have occurred, the error printout which follows is identical with that of the other transfer tests.

The hot swap tests may (like other tests) be terminated by another test call under typewriter control. They will also be terminated at the end of an iteration if sense switch 2 is down.

It is intended that an additional printout will be introduced for execution at the beginning of diagnostic run, giving the various error and control conventions used by this program. Also, there may be provided a short error printout facility in which in the event of errors during a transfer, only the names of the fields and the error type code will be recorded.

Flags and Switches

Program flag 1 has the standard signification for typewriter input control.

Program flags 2, 3, and 4 are on when a drum read, write or swap, respectively, is in progress.

Program flag 5 is on in the event (see below) that a transfer of a full block is called for on a one-word-at-a-time basis. It is also used internally by the hot swap tests.