

IDENTIFICATION

Product Code: MAINDEC-08-D1EB-D
Product Name: PDP-8, 8/I Extended Memory Checkerboard
Date Created: May 1, 1968
Maintainer: Diagnostics Group
Author: J. W. Richardson



1. ABSTRACT

The PDP-8, 8/I Extended Memory Checkerboard diagnostic is designed to provide worst case half-select noise conditions in order to determine the operational status of core memory. Four data patterns, and their complements, are written and checked for error. The patterns provided will generate the worst case noise conditions for a PDP-8 or 8/I equipped with standard or specially purchased core stacks, and will test systems equipped with from 8K to 32K words of core memory. Automatic program relocation is provided in order to test all memory stacks from each stack.

Teletype print-outs are provided for error identification. Also, the operator is given a degree of control over the program by various SR settings. These are explained in detail in Section 8.2.

2. REQUIREMENTS

2.1 Equipment

A standard PDP-8 or 8/I equipped with at least 8K words of core memory.

2.2 Storage

The program occupies locations 0010 to 3334.

2.3 Preliminary Programs

The Binary Loader must be in memory. Also, all diagnostics for a basic 4K PDP-8 must have been previously run successfully.

3. LOADING PROCEDURE

- a. Turn off the Teletype reader.
- b. Set the SR to 7777.
- c. Press LOAD ADDRESS; then START.
- d. Place the Binary tape in Teletype reader and turn on the reader.
- e. When the program has been loaded, stop the computer, turn off the reader, and remove the tape.

4. STARTING PROCEDURE

4.1 Starting Address

Start from address 200 to specify the amount of core memory to test; SR settings, and to receive a header print-out.

4.2 Restarting Address

Start from address 207 to change the test limits; SR settings, and to inhibit the header print-out.

4.3 Operator Action

Immediately after starting from address 200 or 207, the program will print TEST LIMITS. The operator must then specify, via the Teletype keyboard, the amount of core memory to test, followed by a carriage return.

The following rules govern the amount of memory to test:

- a. Type two octal numbers, separating the numbers with a comma. The first number signifies the lowest order 4K stack to test; the second signifies the highest order.
- b. The program expects the 4K stacks to be numbered sequentially starting with a stack 0.
- c. If the highest order stack to test is typed as the first stack, the program will interchange the two values so as to make the second value the first to test.
- d. After typing the second octal number, press the carriage return key to terminate the line.
- e. The program will test the lowest and highest order 4K stack specified, plus every stack between, starting with the lowest specified.
- f. Any single stack, or two or more sequential stacks may be specified.
- g. The stack containing the program may be included when specifying two or more stacks. The stack containing the program will be tested after automatic program relocation takes place (see Section 5.3.1).
- h. If a typing error is made, press the RUB-OUT key. TEST LIMITS will be printed again.

All previous input is disregarded.

For the following examples assume the program to be located in stack 0, and the program has been started from address 200 or 207. The amount of core memory available is 32K.

Example A:

TEST LIMITS

0,7↵ (↵ denotes carriage return)

Example A indicates stacks 0, 1, 2, 3, 4, 5, 6 and 7 will be tested.

Example B:

TEST LIMITS

7,0

The program will perform exactly as Example A.

Example C:

TEST LIMITS

4,5

Only stacks 4 and 5 will be tested.

Example D:

TEST LIMITS

3,3

Stack 3 alone will be tested.

Example E:

TEST LIMITS

0,0 PROGRAM IS LOCATED IN FIELD 0

TEST LIMITS

0,1

Example E shows the message printed by the program when a single stack is selected which currently contains the program. TEST LIMITS is printed again, and the operator must then correct the test limits.

Operation of the program is unpredictable if the amount of memory selected for testing exceeds the actual amount available, i.e., selecting 32K for testing on a PDP-8 or 8/I equipped with a maximum of 28K.

4.3.1 Setup SR

After the test limit is specified, the program will print SETUP SR. For normal program operation, the SR must be set to equal 0000_g. Press the carriage return key after setting the SR to 0000. The program will then run until stopped by the operator. Normal program operation is defined as performing all four checkerboard patterns on all of available memory from every memory stack.

5. OPERATING PROCEDURE

5.1 Program and Operator Action

- a. Load the program into stack 0 using the procedure described in Section 3.
- b. Set the SR to 200; press LOAD ADDRESS, and then start.
- c. The message TEST LIMITS will be printed. Specify the limits, via keyboard, as described in Section 4.3.
- d. The message SETUP SR will be printed. Set the SR to 0000_g, and press the carriage return key.
- e. The program will perform all four tests on all of core memory specified, after which, automatic program relocation takes place.

5.2 Operational Switch Settings

Normal operation of the program requires the SR set to 0000_g. Refer to Section 8.2, applications, for switch settings provided for trouble-shooting.

5.3 Subroutine Abstracts

5.3.1 Program Relocation

Program relocation is governed entirely by the amount of core memory selected for testing. Under certain conditions the program will not relocate at all, but will remain in the current 4K stack to perform the tests (see below). The program first relocates to the highest order 4K stack under test. From there it relocates to the next lower stack (after performing all four tests). The program keeps relocating to the next lower stack until it reaches the lowest order stack under test. The testing and relocation cycle is then repeated.

The contents of the entire 4K stack are relocated. This enables the RIM Loader, and any other information to be carried with the program.

The program provides a degree of protection for itself by recording the first error encountered in any stack. When a faulty stack is next in sequence to contain the program, the program will skip the faulty stack and relocate to the first lower order stack which is error-free. If all lower order stacks are faulty, program relocation will not take place. The tests will be run again from the current stack. Relocation will resume when an error-free stack is found.

Also, the program will not relocate if any of the conditions described below exist.

- a. Only one 4K stack is selected for testing.
- b. SR 9 is on a 1 to inhibit relocation (see Section 8.2.6).

The INSTRUCTION FIELD indicators will indicate the current stack containing the program.

5.3.2 The Checkerboard Patterns

Four test patterns, and their complements, are used to test memory. All memory stacks, except the one with the program, are tested with one pattern before the next test is executed.

Any one, or any combination, of the four tests may be run by placing one, or any combination, of SR 3, 4, 5, or 6 on a 1 after the message SETUP SR is printed. The test specified by the most significant switch on a 1 will be executed first. SR 3, 4, 5 and 6 all on a 0 will enable all tests to be run. SR 3= test 1; 4= test 2; 5= test 3; 6= test 4.

The following steps are performed by each of the four tests:

- a. Write the pattern once in all stacks selected for testing; starting with the lowest order stack.
- b. Select the lowest order stack and perform a read, complement data, write sequence once on each location, until all 4K has been complemented.

- c. Repeat step b 31 more times. The stack will end up with the pattern originally loaded.

No error checking has been performed as yet.

- d. Read 4-word segments and complement each segment 4 times; then read each of the 4 words and check for error.

- e. After checking the entire 4K stack for errors, repeat step d again. This time stall for a random period of time after reading and checking every 400₈ word block. The maximum stall is 18.4 ms; the minimum is 3 μ s.

- f. Setup for the next sequential 4K stack and repeat steps b through f.

When all selected stacks have been checked the next test in sequence is executed, and steps a through f repeated. Program relocation takes place after the fourth test is executed in this manner.

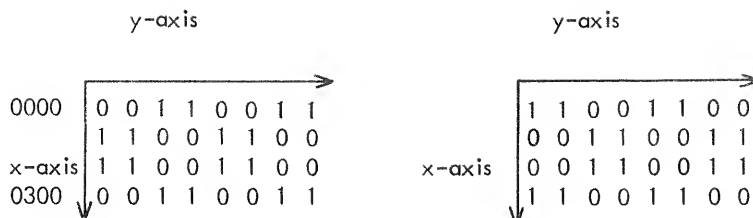
The patterns generated by each test are shown below. The matrices represent portions of one bit plane.

Test 1:

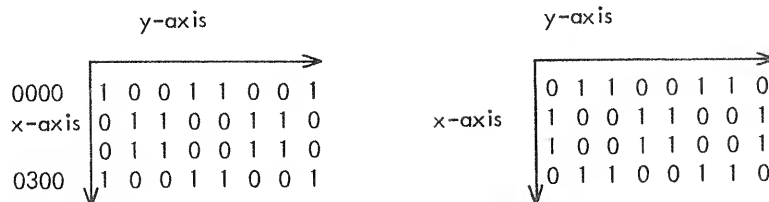
	y-axis	
0000	0 0 1 1 0 0 1 1	→
x-axis	0 0 1 1 0 0 1 1	
	1 1 0 0 1 1 0 0	
0300	1 1 0 0 1 1 0 0	↓

	y-axis	
1 1 0 0 1 1 0 0	1 1 0 0 1 1 0 0	→
x-axis	1 1 0 0 1 1 0 0	
	0 0 1 1 0 0 1 1	
	0 0 1 1 0 0 1 1	↓

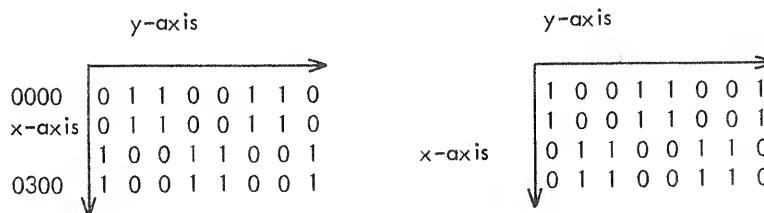
Test 2:



Test 3:



Test 4:



6. ERRORS

Starting the program from address 200 will give a header print-out after the SR has been setup. The header identifies the information printed when a data error is found. The header appears as:

	FIELD	OCTAL ADR.	GOOD	BAD	TEST
Where:	FIELD	= an octal number (0 to 7) indicating the 4K field containing the error.			
	OCTAL ADR.	= the memory address which contains the incorrect data.			
	GOOD	= what the data in octal, should have been. This will always equal 0000 or 7777.			
	BAD	= the data as read. This will equal the good data except for one or more bits complemented.			
	TEST	= the number (1 to 4) of the test which detected the error.			

After each error print-out the program continues on with the next sequential memory location.

6.1 Error Halts and Description

Placing SR 0 on a 1 during an error print-out will cause a halt at location 2641.
Press CONTINUE to resume testing.

7. RESTRICTIONS

7.1 Starting Restrictions

Start from address 200 to indicate the amount of core memory to test; to setup the SR and to receive a header print-out.

Starting from 207 requires the same operator action, but no header will be printed.

7.2 Operating Restrictions

None

8. MISCELLANEOUS

8.1 Execution Time

The time required to perform all four tests on one 4K memory stack is approximately 26 seconds.

8.2 Applications

For operating convenience, and as an aid to trouble-shooting, the SR may be used to control the program. The switch assignments and their effect on the program are described below. Please note that it is important that the program should be halted before changing the test selection switches. These switches are not sensed by the program during testing.

Halting the program with SR 0 is preferred, rather than with the STOP key. Using the STOP key may result in a halt while the program is in the process of relocating, which is disastrous.

8.2.1 Halt after Test or Error - SR 0

Placing SR 0 on a 1 at any time while the program is running will cause a halt after the current test is completed. The MB will equal 2461 in the current stack containing the program. Press CONTINUE to resume testing, or restart from 200 or 207 to enter new parameters.

Placing SR 0 on a 1 during an error type-out will also cause a halt at location 2461. Proceed exactly as described in the above paragraph.

8.2.2 Inhibit Error Print-out - SR 1

Placing SR 1 on a 1 causes all error print-outs to be inhibited. All other messages will not be inhibited. The program will continue to recognize errors, but will not print any information. SR 1 may be placed on a 1 or 0 while the program is running.

8.2.3 Bell on Error - SR 2

SR 2 on a 1 causes the program to ring the TTY BELL whenever an error is detected. This is convenient when testing with power supply margins. SR 2 has precedence over SR 1 if both should happen to be on a 1. SR 2 may be placed on a 1 or 0 while the program is running.

8.2.4 Test Selection SR 3 through 6

Any one, or any combination of tests may be executed by placing any one or any combination of SR 3 through 6 on a 1. Test selections may be made only when starting from 200 or 207. SR 3 specifies test 1; SR 4 test 2; SR 5 test 3; SR 6 test 4. The test specified by the most significant SR on a 1 will be executed first.

For most PDP-8s, SR 4 will provide the worst case pattern. For most PDP-8/Is, SR 5 will provide the worst case pattern.

If all four switches are on a 0, all four tests will be executed in order starting with test 1. Program relocation is not effected, regardless of the SR settings.

8.2.5 Inhibit Program Relocation - SR 7

The program normally relocates automatically as indicated by the INSTRUCTION FIELD indicators. To retain the program in its current 4K field, place SR 7 on a 1 at any time. Changing SR 7 to a 0 will permit relocation to resume.

8.2.6 SR 8, 9 and 10 - Not Used

8.2.7 Change TEST LIMITS and SR - SR 11

Placing SR 11 on a 1 will cause the program to automatically restart from address 207. The TEST LIMITS and SR may then be changed. SR 11 is sensed only after all specified tests have been completed on all of memory under test.

8.2.8 Loop on Address

A subroutine is provided which may be used to continuously loop on a single location, or a group of consecutive locations. No error checking is performed. The routine performs a read, and immediately follows with a write, on each location. The loop time between two reads, or two writes, is approximately 22.5 μ s.

Operating Procedure:

- a. Set the INSTRUCTION FIELD switches to the current field, and the SR to 1700.
- b. Set the DATA FIELD switches to equal the 4K field number to test.
- c. Press LOAD ADDRESS.
- d. Set the SR to equal the first address of the group.
- e. Press START. A halt will occur at 1703. Set the SR to equal the last address of the group.
- f. Press CONTINUE. The address(s) specified will be looped until stopped by the operator with STOP. SR 0 will not halt this routine.

To resume normal operation, restart the program from 200 or 207 of the current field.

9. PROGRAM DESCRIPTION

The PDP-8, 8/I Extended Memory Checkerboard diagnostic is designed to create worst case memory noise conditions on systems equipped with 8K to 32K words of memory. The program executes four checkerboard patterns, plus their complements, on each 4K memory field. In addition, the program automatically relocates from field to field in order to test all 4K fields from every 4K field. Under normal operation, the amount of core memory tested at one time is that specified by the operator minus the 4K field containing the program. A TTY keyboard input routine is provided to enable the operator to specify the exact number of 4K fields to be tested. A print-out is provided for each error detected by the program.

Further control of the program is given to the operator by means of the SR. The operator may halt the program, inhibit error print-outs, substitute the TTY BELL for error indication, halt after error print-out, select any one or a group of tests, inhibit program relocation, and create an automatic restart to change the amount of memory to test.

A small subroutine is provided which will continuously read and write any single, or a group of locations within any 4K field. The operator must specify the locations by means of the SR.

Handwritten notes and signatures at the top of the page.

APR 24 1968

/PDP-8, 81, 8S EXTENDED MEMORY CHECKBOARD TEST,
/START AT 200, RESTART AT 211 TO SKIP HEADER,
/MIN. OF 8K OF CORE REQUIRED.

0281
5001
0002
0003

JMP,
0202
0203

COF=6221
CIF=6222
RUF=6214
RIF=6224

0204 0000
0205 0000
0206 0000
0207 0000
0210 0000
0211 0000
0212 0000
0213 0000
0214 0000
0215 0000
0216 0000
0217 0000
0220 0000
0221 0000
0222 0000
0223 0000
0224 0000
0225 0000
0226 0000
0227 0000
0230 0000
0231 0000
0232 0000
0233 0000
0234 0000
0235 0000
0236 0000
0237 0000
0240 0000
0241 0000
0242 0000
0243 0000
0244 0000
0245 0000
0246 0000
0247 0000
0250 0000
0251 0000
0252 0000
0253 0000

DATAFLD,
FLAGS,
INSEFLD,
FLOAD,
ERTBL,
ENTBL,
ERWRD,
ERWRD+10

7600
7600
7600
7600
7600
7600
7600
7600

NOVA,
K10,
K740,
K400,
K200,
K100,
K40,
K20,
XLMTS,
XTST1,
XTST2,
XTST3,
XTST4,
XMOVE,
XSETU,
K261,
K262,

0207
0208
0209
0210
0211
0212
0213
0214
0215
0216
0217
0220
0221
0222
0223
0224
0225
0226
0227
0230
0231
0232
0233
0234
0235
0236
0237
0240
0241
0242
0243
0244
0245
0246
0247
0250
0251
0252
0253

K263, 263
K264, 264
M20, 7760

M054 0263
M055 0264
M056 7760

0057	7740	M40,	7740
0060	7774	M4,	7774
0061	7773	M5,	7773
0062	0000	TNUM,	0
0063	1627	CBANK,	CBANK
0064	0652	X011,	X011
0065	0667	X1100,	X1100
0066	0734	X0110,	X0110
0067	0721	X1001,	X1001
0070	1600	CKBANK,	CKBANK
0071	1624	CKBANK,	CKBANK
0072	0000	COUNT,	0
0073	0000	FLECNT,	0
0074	0000	LOOP,	0
0075	0736	XRALL,	RDALL
0076	1037	XC-K1,	RCHK1
0077	1054	XC-K10,	RCHK10
0100	1071	XC-K2,	RCHK2
0101	1106	XC-K20,	RCHK20
0102	1123	XC-K3,	RCHK3
0103	1140	XC-K30,	RCHK30
0104	1200	XC-K4,	RCHK4
0105	1217	XC-K40,	RCHK40
0106	1056	TDM20,	TAD M20
0107	1057	TDM40,	TAD M40
0110	4515	JMS1,	JMS I XR01
0111	4516	JMS2,	JMS I XR02
0112	4517	JMS3,	JMS I XR03
0113	4520	JMS4,	JMS I XR04
0114	4552	JMS5,	JMS I XSALL
0115	1245	XR01,	RD1
0116	1322	XR02,	RD2
0117	1400	XR03,	RD3
0120	1455	XR04,	RD4
0121	2000	ERROR,	ERROR
0122	0000	MEMADR,	0
0123	0000	FIRST1,	0
0124	0000	LAST1,	0
0125	6201	KCIF,	6201
0126	6202	KCIF,	6202
0127	2641	XHLT,	HALT
0130	0213	RTN1,	RTN1
0131	1646	FIELD,	FIELD
0132	2146	PRER,	PRERR
0133	0007	K7,	7
0134	0000	CHAR,	0
0135	2474	PHDR,	PHDR
0136	2146	XPERR,	PRERR
0137	2115	XPING,	SPING

014 7764
 0141 7770
 0142 0260
 0143 0215
 0144 0377
 0145 0370
 0146 0277
 0147 2154
 0150 0201
 0151 0400
 0152 2702
 0153 0020
 0154 2166

M14, 7764
 M10, 7770
 K260, 260
 K215, 215
 K377, 377
 K370, 370
 K277, 277
 XCHLF, CRLF
 K1, 1
 NXLOC, 0
 XSALL, STALL
 EXIT, 0
 LASTX, LAST


```

0220  *220
0221  REGIN,
0222  CLA
0223  DCA FLAGS
0224  RIF
0225  DCA INSFLD
0226  JMS I XLMTS
0227  JMS I XTSTP
0228  JMS I XHDR
0229  JMP RTN1
0230  /
0231  /RESTART HERE
0232  /
0233  RSTRT1, JMS I XLMTS
0234  JMS I XTSTP
0235  RIF
0236  DCA INSFLD
0237  JMS I XFILD
0238  TAD M10
0239  DCA LOOP
0240  7600
0241  TAD ALAW
0242  IS# ERWRD
0243  DCA I ERWRD
0244  IS# LOOP
0245  JMP ALAW
0246  TAD ALAW
0247  DCA I LASTX
0248  TAD ERTBL
0249  DCA ERWRD
0250  /
0251  /EXAMINE SR
0252  TAD MCWA
0253  AND K740
0254  SZA
0255  JMP EXAM1
0256  TAD MCWA
0257  TAD K740
0258  DCA MCWA
0259  CLA
0260  TAD MCWA
0261  AND K400
0262  SZA
0263  JMP I XTST1
0264  CLA
0265  TAD MCWA
0266  AND K200
0267  SZA
0268  JMP I XTST2
0269  CLA
0270  TAD MCWA
0271  AND K100
0272  SZA
0273  JMP I XTST3
0274  CLA
0275  TAD MCWA
0276  AND K740
0277  SZA
0278  JMP I XTST3
0279  CLA
0280  TAD MCWA
0281  AND K740
0282  SZA
0283  JMP EXAM1
0284  TAD MCWA
0285  TAD K740
0286  DCA MCWA
0287  CLA
0288  TAD MCWA
0289  AND K400
0290  SZA
0291  JMP I XTST1
0292  CLA
0293  TAD MCWA
0294  AND K200
0295  SZA
0296  JMP I XTST2
0297  CLA
0298  TAD MCWA
0299  AND K100
0300  SZA
0301  JMP I XTST3
0302  CLA
0303  TAD MCWA
0304  AND K740
0305  SZA
0306  JMP EXAM1
0307  TAD MCWA
0308  TAD K740
0309  DCA MCWA
0310  CLA
0311  TAD MCWA
0312  AND K400
0313  SZA
0314  JMP I XTST1
0315  CLA
0316  TAD MCWA
0317  AND K200
0318  SZA
0319  JMP I XTST2
0320  CLA
0321  TAD MCWA
0322  AND K100
0323  SZA
0324  JMP I XTST3
0325  CLA
0326  TAD MCWA
0327  AND K740
0328  SZA
0329  JMP EXAM1
0330  TAD MCWA
0331  TAD K740
0332  DCA MCWA
0333  CLA
0334  TAD MCWA
0335  AND K400
0336  SZA
0337  JMP I XTST1
0338  CLA
0339  TAD MCWA
0340  AND K200
0341  SZA
0342  JMP I XTST2
0343  CLA
0344  TAD MCWA
0345  AND K100
0346  SZA
0347  JMP I XTST3
0348  CLA
0349  TAD MCWA
0350  AND K740
0351  SZA
0352  JMP EXAM1
0353  TAD MCWA
0354  TAD K740
0355  DCA MCWA
0356  CLA
0357  TAD MCWA
0358  AND K400
0359  SZA
0360  JMP I XTST1
0361  CLA
0362  TAD MCWA
0363  AND K200
0364  SZA
0365  JMP I XTST2
0366  CLA
0367  TAD MCWA
0368  AND K100
0369  SZA
0370  JMP I XTST3
0371  CLA
0372  TAD MCWA
0373  AND K740
0374  SZA
0375  JMP EXAM1
0376  TAD MCWA
0377  TAD K740
0378  DCA MCWA
0379  CLA
0380  TAD MCWA
0381  AND K400
0382  SZA
0383  JMP I XTST1
0384  CLA
0385  TAD MCWA
0386  AND K200
0387  SZA
0388  JMP I XTST2
0389  CLA
0390  TAD MCWA
0391  AND K100
0392  SZA
0393  JMP I XTST3
0394  CLA
0395  TAD MCWA
0396  AND K740
0397  SZA
0398  JMP EXAM1
0399  TAD MCWA
0400  TAD K740
0401  DCA MCWA
0402  CLA
0403  TAD MCWA
0404  AND K400
0405  SZA
0406  JMP I XTST1
0407  CLA
0408  TAD MCWA
0409  AND K200
0410  SZA
0411  JMP I XTST2
0412  CLA
0413  TAD MCWA
0414  AND K100
0415  SZA
0416  JMP I XTST3
0417  CLA
0418  TAD MCWA
0419  AND K740
0420  SZA
0421  JMP EXAM1
0422  TAD MCWA
0423  TAD K740
0424  DCA MCWA
0425  CLA
0426  TAD MCWA
0427  AND K400
0428  SZA
0429  JMP I XTST1
0430  CLA
0431  TAD MCWA
0432  AND K200
0433  SZA
0434  JMP I XTST2
0435  CLA
0436  TAD MCWA
0437  AND K100
0438  SZA
0439  JMP I XTST3
0440  CLA
0441  TAD MCWA
0442  AND K740
0443  SZA
0444  JMP EXAM1
0445  TAD MCWA
0446  TAD K740
0447  DCA MCWA
0448  CLA
0449  TAD MCWA
0450  AND K400
0451  SZA
0452  JMP I XTST1
0453  CLA
0454  TAD MCWA
0455  AND K200
0456  SZA
0457  JMP I XTST2
0458  CLA
0459  TAD MCWA
0460  AND K100
0461  SZA
0462  JMP I XTST3
0463  CLA
0464  TAD MCWA
0465  AND K740
0466  SZA
0467  JMP EXAM1
0468  TAD MCWA
0469  TAD K740
0470  DCA MCWA
0471  CLA
0472  TAD MCWA
0473  AND K400
0474  SZA
0475  JMP I XTST1
0476  CLA
0477  TAD MCWA
0478  AND K200
0479  SZA
0480  JMP I XTST2
0481  CLA
0482  TAD MCWA
0483  AND K100
0484  SZA
0485  JMP I XTST3
0486  CLA
0487  TAD MCWA
0488  AND K740
0489  SZA
0490  JMP EXAM1
0491  TAD MCWA
0492  TAD K740
0493  DCA MCWA
0494  CLA
0495  TAD MCWA
0496  AND K400
0497  SZA
0498  JMP I XTST1
0499  CLA
0500  TAD MCWA
0501  AND K200
0502  SZA
0503  JMP I XTST2
0504  CLA
0505  TAD MCWA
0506  AND K100
0507  SZA
0508  JMP I XTST3
0509  CLA
0510  TAD MCWA
0511  AND K740
0512  SZA
0513  JMP EXAM1
0514  TAD MCWA
0515  TAD K740
0516  DCA MCWA
0517  CLA
0518  TAD MCWA
0519  AND K400
0520  SZA
0521  JMP I XTST1
0522  CLA
0523  TAD MCWA
0524  AND K200
0525  SZA
0526  JMP I XTST2
0527  CLA
0528  TAD MCWA
0529  AND K100
0530  SZA
0531  JMP I XTST3
0532  CLA
0533  TAD MCWA
0534  AND K740
0535  SZA
0536  JMP EXAM1
0537  TAD MCWA
0538  TAD K740
0539  DCA MCWA
0540  CLA
0541  TAD MCWA
0542  AND K400
0543  SZA
0544  JMP I XTST1
0545  CLA
0546  TAD MCWA
0547  AND K200
0548  SZA
0549  JMP I XTST2
0550  CLA
0551  TAD MCWA
0552  AND K100
0553  SZA
0554  JMP I XTST3
0555  CLA
0556  TAD MCWA
0557  AND K740
0558  SZA
0559  JMP EXAM1
0560  TAD MCWA
0561  TAD K740
0562  DCA MCWA
0563  CLA
0564  TAD MCWA
0565  AND K400
0566  SZA
0567  JMP I XTST1
0568  CLA
0569  TAD MCWA
0570  AND K200
0571  SZA
0572  JMP I XTST2
0573  CLA
0574  TAD MCWA
0575  AND K100
0576  SZA
0577  JMP I XTST3
0578  CLA
0579  TAD MCWA
0580  AND K740
0581  SZA
0582  JMP EXAM1
0583  TAD MCWA
0584  TAD K740
0585  DCA MCWA
0586  CLA
0587  TAD MCWA
0588  AND K400
0589  SZA
0590  JMP I XTST1
0591  CLA
0592  TAD MCWA
0593  AND K200
0594  SZA
0595  JMP I XTST2
0596  CLA
0597  TAD MCWA
0598  AND K100
0599  SZA
0600  JMP I XTST3
0601  CLA
0602  TAD MCWA
0603  AND K740
0604  SZA
0605  JMP EXAM1
0606  TAD MCWA
0607  TAD K740
0608  DCA MCWA
0609  CLA
0610  TAD MCWA
0611  AND K400
0612  SZA
0613  JMP I XTST1
0614  CLA
0615  TAD MCWA
0616  AND K200
0617  SZA
0618  JMP I XTST2
0619  CLA
0620  TAD MCWA
0621  AND K100
0622  SZA
0623  JMP I XTST3
0624  CLA
0625  TAD MCWA
0626  AND K740
0627  SZA
0628  JMP EXAM1
0629  TAD MCWA
0630  TAD K740
0631  DCA MCWA
0632  CLA
0633  TAD MCWA
0634  AND K400
0635  SZA
0636  JMP I XTST1
0637  CLA
0638  TAD MCWA
0639  AND K200
0640  SZA
0641  JMP I XTST2
0642  CLA
0643  TAD MCWA
0644  AND K100
0645  SZA
0646  JMP I XTST3
0647  CLA
0648  TAD MCWA
0649  AND K740
0650  SZA
0651  JMP EXAM1
0652  TAD MCWA
0653  TAD K740
0654  DCA MCWA
0655  CLA
0656  TAD MCWA
0657  AND K400
0658  SZA
0659  JMP I XTST1
0660  CLA
0661  TAD MCWA
0662  AND K200
0663  SZA
0664  JMP I XTST2
0665  CLA
0666  TAD MCWA
0667  AND K100
0668  SZA
0669  JMP I XTST3
0670  CLA
0671  TAD MCWA
0672  AND K740
0673  SZA
0674  JMP EXAM1
0675  TAD MCWA
0676  TAD K740
0677  DCA MCWA
0678  CLA
0679  TAD MCWA
0680  AND K400
0681  SZA
0682  JMP I XTST1
0683  CLA
0684  TAD MCWA
0685  AND K200
0686  SZA
0687  JMP I XTST2
0688  CLA
0689  TAD MCWA
0690  AND K100
0691  SZA
0692  JMP I XTST3
0693  CLA
0694  TAD MCWA
0695  AND K740
0696  SZA
0697  JMP EXAM1
0698  TAD MCWA
0699  TAD K740
0700  DCA MCWA
0701  CLA
0702  TAD MCWA
0703  AND K400
0704  SZA
0705  JMP I XTST1
0706  CLA
0707  TAD MCWA
0708  AND K200
0709  SZA
0710  JMP I XTST2
0711  CLA
0712  TAD MCWA
0713  AND K100
0714  SZA
0715  JMP I XTST3
0716  CLA
0717  TAD MCWA
0718  AND K740
0719  SZA
0720  JMP EXAM1
0721  TAD MCWA
0722  TAD K740
0723  DCA MCWA
0724  CLA
0725  TAD MCWA
0726  AND K400
0727  SZA
0728  JMP I XTST1
0729  CLA
0730  TAD MCWA
0731  AND K200
0732  SZA
0733  JMP I XTST2
0734  CLA
0735  TAD MCWA
0736  AND K100
0737  SZA
0738  JMP I XTST3
0739  CLA
0740  TAD MCWA
0741  AND K740
0742  SZA
0743  JMP EXAM1
0744  TAD MCWA
0745  TAD K740
0746  DCA MCWA
0747  CLA
0748  TAD MCWA
0749  AND K400
0750  SZA
0751  JMP I XTST1
0752  CLA
0753  TAD MCWA
0754  AND K200
0755  SZA
0756  JMP I XTST2
0757  CLA
0758  TAD MCWA
0759  AND K100
0760  SZA
0761  JMP I XTST3
0762  CLA
0763  TAD MCWA
0764  AND K740
0765  SZA
0766  JMP EXAM1
0767  TAD MCWA
0768  TAD K740
0769  DCA MCWA
0770  CLA
0771  TAD MCWA
0772  AND K400
0773  SZA
0774  JMP I XTST1
0775  CLA
0776  TAD MCWA
0777  AND K200
0778  SZA
0779  JMP I XTST2
0780  CLA
0781  TAD MCWA
0782  AND K100
0783  SZA
0784  JMP I XTST3
0785  CLA
0786  TAD MCWA
0787  AND K740
0788  SZA
0789  JMP EXAM1
0790  TAD MCWA
0791  TAD K740
0792  DCA MCWA
0793  CLA
0794  TAD MCWA
0795  AND K400
0796  SZA
0797  JMP I XTST1
0798  CLA
0799  TAD MCWA
0800  AND K200
0801  SZA
0802  JMP I XTST2
0803  CLA
0804  TAD MCWA
0805  AND K100
0806  SZA
0807  JMP I XTST3
0808  CLA
0809  TAD MCWA
0810  AND K740
0811  SZA
0812  JMP EXAM1
0813  TAD MCWA
0814  TAD K740
0815  DCA MCWA
0816  CLA
0817  TAD MCWA
0818  AND K400
0819  SZA
0820  JMP I XTST1
0821  CLA
0822  TAD MCWA
0823  AND K200
0824  SZA
0825  JMP I XTST2
0826  CLA
0827  TAD MCWA
0828  AND K100
0829  SZA
0830  JMP I XTST3
0831  CLA
0832  TAD MCWA
0833  AND K740
0834  SZA
0835  JMP EXAM1
0836  TAD MCWA
0837  TAD K740
0838  DCA MCWA
0839  CLA
0840  TAD MCWA
0841  AND K400
0842  SZA
0843  JMP I XTST1
0844  CLA
0845  TAD MCWA
0846  AND K200
0847  SZA
0848  JMP I XTST2
0849  CLA
0850  TAD MCWA
0851  AND K100
0852  SZA
0853  JMP I XTST3
0854  CLA
0855  TAD MCWA
0856  AND K740
0857  SZA
0858  JMP EXAM1
0859  TAD MCWA
0860  TAD K740
0861  DCA MCWA
0862  CLA
0863  TAD MCWA
0864  AND K400
0865  SZA
0866  JMP I XTST1
0867  CLA
0868  TAD MCWA
0869  AND K200
0870  SZA
0871  JMP I XTST2
0872  CLA
0873  TAD MCWA
0874  AND K100
0875  SZA
0876  JMP I XTST3
0877  CLA
0878  TAD MCWA
0879  AND K740
0880  SZA
0881  JMP EXAM1
0882  TAD MCWA
0883  TAD K740
0884  DCA MCWA
0885  CLA
0886  TAD MCWA
0887  AND K400
0888  SZA
0889  JMP I XTST1
0890  CLA
0891  TAD MCWA
0892  AND K200
0893  SZA
0894  JMP I XTST2
0895  CLA
0896  TAD MCWA
0897  AND K100
0898  SZA
0899  JMP I XTST3
0900  CLA
0901  TAD MCWA
0902  AND K740
0903  SZA
0904  JMP EXAM1
0905  TAD MCWA
0906  TAD K740
0907  DCA MCWA
0908  CLA
0909  TAD MCWA
0910  AND K400
0911  SZA
0912  JMP I XTST1
0913  CLA
0914  TAD MCWA
0915  AND K200
0916  SZA
0917  JMP I XTST2
0918  CLA
0919  TAD MCWA
0920  AND K100
0921  SZA
0922  JMP I XTST3
0923  CLA
0924  TAD MCWA
0925  AND K740
0926  SZA
0927  JMP EXAM1
0928  TAD MCWA
0929  TAD K740
0930  DCA MCWA
0931  CLA
0932  TAD MCWA
0933  AND K400
0934  SZA
0935  JMP I XTST1
0936  CLA
0937  TAD MCWA
0938  AND K200
0939  SZA
0940  JMP I XTST2
0941  CLA
0942  TAD MCWA
0943  AND K100
0944  SZA
0945  JMP I XTST3
0946  CLA
0947  TAD MCWA
0948  AND K740
0949  SZA
0950  JMP EXAM1
0951  TAD MCWA
0952  TAD K740
0953  DCA MCWA
0954  CLA
0955  TAD MCWA
0956  AND K400
0957  SZA
0958  JMP I XTST1
0959  CLA
0960  TAD MCWA
0961  AND K200
0962  SZA
0963  JMP I XTST2
0964  CLA
0965  TAD MCWA
0966  AND K100
0967  SZA
0968  JMP I XTST3
0969  CLA
0970  TAD MCWA
0971  AND K740
0972  SZA
0973  JMP EXAM1
0974  TAD MCWA
0975  TAD K740
0976  DCA MCWA
0977  CLA
0978  TAD MCWA
0979  AND K400
0980  SZA
0981  JMP I XTST1
0982  CLA
0983  TAD MCWA
0984  AND K200
0985  SZA
0986  JMP I XTST2
0987  CLA
0988  TAD MCWA
0989  AND K100
0990  SZA
0991  JMP I XTST3
0992  CLA
0993  TAD MCWA
0994  AND K740
0995  SZA
0996  JMP EXAM1
0997  TAD MCWA
0998  TAD K740
0999  DCA MCWA
1000  CLA

```

0260 7200
0261 1033
0262 0241
0263 7440
0264 5447
0265 4531

EXAM4.

CLA
TAD MCWA
AND K40
SZA /TEST 4 IF NO SKIP
JMP I XTST4 /RESTORE DATA FIELD
JMS I XFILD

0266 7604
 0267 8150
 0270 7440
 0271 5211
 0272 7624
 0273 0742
 0274 7440
 0275 5213
 0276 5450
 0277 2645

LAS
 AND K1
 SZA RSTRT1
 JMP RSTRT1
 LAS
 AND K2A
 SZA
 JMP RTN1
 JMP I XMOVE
 / XTSR, SETSR

/CHECK SR 11

/INHIBIT MOVE IF A 1
 /GO RELOCATE

```

4451
0320
0321 7200
0322 1052
0323 3262
0324 7340
0325 3010
0326 4463
0327 7410
0328 5322
0329 1345
0330 3153
0331 4464
0332 4464
0333 4465
0334 4465
0335 5313
0336 4471
0337 5304

0322 4476
0323 5324

0324 4451
0325 7240
0326 3010
0327 4463
0328 7410
0329 5343
0330 1346
0331 3153
0332 4465
0333 4465
0334 4464
0335 5334
0336 4471
0337 5325

0343 4477
0344 5246

0345 0320
0346 0341

/TEST 1, WRITE CHECKER PATTERN #1.
/
TST1, JMS I XSETU /SET DF TO 1ST FIELD
CLA /TEST NUMBER
TAD K261
DCA TNUM
CMA
DCA 10
JMS I XBANK /SFT ADDRESS COUNT TO 7777
SKP /SEL IF FIELD HAS PROGRAM
JMP EXT1 /NO, BEGIN WRITING
TAD KXT1 /DONE ALL, NOW READ ALL
DCA EXIT
JMS I X0011 /WRITE 0011
JMS I X0011 /WRITE 0011 64 TIMES
JMS I X1100 /WRITE 1100 128 TIMES
JMS I X1100
JMP :-4 /KEEP WRITING
JMS I XTRNK /SETUP FOR NEXT FIELD
JMP TST1+4

XIT1, JMS I XCHK1 /READ EACH FIELD AND CHECK
/FOR ERRORS,
JMP TST1C /NOW WRITE COMPLEMENT

/
WRITE COMPLEMENT OF PATTERN 1
/
TST1C, JMS I XSETU /SEE DF TO 1ST FIELD,
CLA CMA /SET ADDRESS COUNT TO 7777
DCA 10 /SEE IF FIELD HAS PROGRAM
JMS I XBANK
SKP /ALL DONE, READ ALL
JMP EXT1C
TAD KXT1C
DCA EXIT
JMS I X1100 /WRITE 1100
JMS I X1100 /WRITE 1100 16 TIMES
JMS I X0011 /WRITE 0011 128 TIMES
JMS I X0011 /KEEP WRITING
JMP :-4 /SETUP FOR NEXT FIELD
JMS I XTRNK
JMP TST1C+1

EXT1C, JMS I XCHK1C /READ EACH BANK AND CHECK
/FOR ERRORS,
JMP EXAM2 /SEE IF TEST 2 IS SELECTED

KXT1, XIT1
KXT1C, XIT1C

```

/TEST 2, WRITE CHECKER PATTERN #2

/*400
/TST2,

0420
0402 4451
0401 7200
0402 1053
0403 3062
0404 7240
0405 3010
0406 4463
0407 7410
0410 5223
0411 1250
0412 3153
0413 4464
0414 4465
0415 4465
0416 4464
0417 4464
0420 5214
0421 4471
0422 5204

JMS I XSETU /SET DF FOR 1ST FIELD
CLA
TAD K262 /TEST #
DCA TNUM
CLA CMA
DCA 10 /SET ADDRESS COUNT TO 7777
JMS I XBANK /SEE IF FIELD HAS PROGRAM
SKP EXT2 /NO. BEGIN WRITING
JMP EXT2 /DONE ALL, NOW READ ALL
TAD KXT2
DCA EXIT /WRITE 0011
JMS I X0011 /WRITE 1100 128 TIMES
JMS I X1100
JMS I X1100
JMS I X0011 /WRITE 0011 128 TIMES
JMS I X0011
JMP I-4
JMS I XTBNK /SETUP FOR NEXT FIELD
JMP TST2+4

XIT2,

EXT2, JMS I XCHK? /READ EACH FIELD AND CHECK
JMP TST2C /NOW WRITE COMPLEMENT

/WRITE COMPLEMENT OF PATTERN 2

TST2C, JMS I XSETU /SET DF FOR FIRST FIELD

CLA CMA
DCA I0 /SET ADR. COUNT TO 7777
JMS I XBANK /SEE IF FIELD HAS PROGRAM
SKP EXT2C /WRITE
/GO READ

JMS I X1100 /WRITE 1100
JMS I X0011 /WRITE 0011 128 TIMES

JMS I X1100 /WRITE 1100 128 TIMES
JMS I X1100

JMP :-4 /SETUP FOR NEXT FIELD
JMS I XTBANK
JMP TST2C+1

EXT2C, JMS I XCHK2C /READ EACH FIELD AND CHECK
JMP I ,*1 /SEE IF TEST 3 IS SELECTED
EXAM3

KXT2, XIT2
KXT2C, XIT2C

0423 4500
0424 5225

0425 4451
0426 7240
0427 3010
0430 4463
0431 7410
0432 5245
0433 1251
0434 3153
0435 4465
0436 4464
0437 4464
0440 4465
0441 4465
0442 5236
0443 4471
0444 5226

0445 4501
0446 5647
0447 0253

0450 0421
0451 0443

```

/TEST 3. WRITE CHECKER PATTERN #3
/
TST3, JMS I XSETU /SETUP FOR 1ST FIELD
CLA
TAD K26J
DCA TNUM /TEST NUMBER
CLA CMA
DCA 10 /SET ADR. COUNT TO 7777
JMS I XBANK /SEE IF FIELD HAS PROGRAM
SKP /GO WRITE
JMP EXT3 /GO READ
TAD KXT3
DCA EXIT
JMS I X1001 /WRITE 1001
JMS I X0110 /WRITE 0110 128 TIMES
JMS I X0110
JMS I X1001 /WRITE 1001 128 TIMES
JMS I X1001
JMP ,--4
JMS I XTBNK /SETUP FOR NEXT FIELD
JMP TST3+4

XIT3, /
EXT3, JMS I XCHK3 /READ EACH FIELD AND CHECK
JMP TST3C /WRITE COMPLEMENT

/ WRITE COMPLEMENT OF PATTERN 3
/
TST3C, JMS I XSETU /SETUP DF FOR 1ST FIELD
CLA CMA
DCA 10 /SET ADR. COUNT TO 7777
JMS I XBANK /SEE IF FIELD HAS PROGRAM
SKP /WRITE
JMP EXT3C /READ ALL
TAD KXT3C
DCA EXIT
JMS I X0110 /WRITE 0110
JMS I X1001 /WRITE 1001 128 TIMES
JMS I X1001
JMS I X0110 /WRITE 0110 128 TIMES
JMS I X0110
JMP ,--4
JMS I XTBNK /SETUP FOR NEXT FIELD
JMP TST3C+1

XIT3C, /
EXT3C, JMS I XCHK3C /READ EACH FIELD AND CHECK
JMP I ,+1 /SEE IF TEST 4 IS SELECTED
EXAM4

KXT3, XIT3
KXT3C, XIT3C

```

0452 4451
0453 7240
0454 1354
0455 3262
0456 7240
0457 3010
0460 4463
0461 7410
0462 5275
0463 1322
0464 3153
0465 4467
0466 4466
0467 4466
0470 4467
0471 4467
0472 5266
0473 4471
0474 5256

0475 4502
0476 5277

0477 4451
0500 7240
0501 3010
0502 4463
0503 7410
0504 5317
0505 1323
0506 3153
0507 4466
0510 4467
0511 4467
0512 4466
0513 4466
0514 5310
0515 4471
0516 5300

0517 4503
0520 5721
0521 0260
0522 0473
0523 0515

```

/TFST 4, WRITE PATTERN #4
/
*010
/
TST4, /SET DF FOR 1ST FIELD
CLA
TAD K244 /TEST NUMBER
DCA TVUM
CLA CMA
DCA 10 /SET ADR, COUNT TO 7777
JMS I XBANK /SEE IF FIELD HAS PROGRAM
SKP EXT4 /WRITE
JMP EXT4 /GO READ
TAD KXT4
DCA EXIT
JMS I X0110 /WRITE 0110
JMS I X0110 /WRITE 0110 64 TIMES
JMS I X1001 /WRITE 1001 128 TIMES
JMS I X1001
JMS I X0110
JMP *-4
XIT4, /SETUP FOR NEXT FIELD
JMS I XTBNK
JMP TST4+4

/
EXT4, /READ EACH FIELD AND CHECK
JMP TST4C /WRITE COMPLEMENT

/
WRITE COMPLEMENT OF PATTERN 4
/
TST4C, /SET DF FOR FIRST
CLA CMA /SET ADR, COUNT TO 7777
DCA 10 /SEE IF FIELD HAS PROGRAM
JMS I XBANK /WRITE
SKP /READ
JMP EXT4C
TAD KXT4C
DCA EXIT
JMS I X1001 /WRITE 1001 64 TIMES
JMS I X1001 /WRITE X0110 128 TIMES
JMS I X0110
JMS I X1001
JMP *-4
XIT4C, /SETUP FOR NEXT FIELD
JMS I XTBNK
JMP TST4C+1

/
KXT4, XIT4
KXT4C, XIT4C

```

0620

0621 4451
0622 7240
0623 1255
0624 3062
0625 7242
0626 3112
0627 4463
0628 7411
0629 5223
0630 1245
0631 3153
0632 4466
0633 4466
0634 4467
0635 4467
0636 4466
0637 5214
0638 4471
0639 5224

0623 4504
0624 5225

0625 4451
0626 7240
0627 3010
0628 4463
0629 7410
0630 5247
0631 1246
0632 3153
0633 4467
0634 4467
0635 4466
0636 4467
0637 4466
0638 4467
0639 5236
0640 4471
0641 5226
0642 0621
0643 0643

EXT4C, JMS I XCHK4C /READ FACH FIELD AND CHECK
 JMP I ,+1 /SFE IF READY TO MOVE
 EXAM4+5

/ROUTINE TO WRITE W011
 /

W011, 2
 TAD M20
 JCA COUNT /0
 DCA I 10 /0
 DCA I 10 /1
 CMA /1
 DCA I 10 /1
 CMA /1
 DCA I 10 /1
 ISZ COUNT /COUNT = -16 OR -32
 JMP W011+3 /LOOP
 JMS I XKRBNK /SEE IF END OF FIELD
 JMP I W011 /EXIT

/ROUTINE TO WRITE I100
 /

W1100, 0
 TAD M20
 DCA COUNT /1
 CMA /1
 DCA I 10 /1
 CMA /0
 DCA I 10 /0
 DCA I 10 /0
 ISZ COUNT /-16 OR -32
 JMP W1100+3 /LOOP
 JMS I XKRBNK /SEE IF END OF FIELD
 JMP I W1100 /EXIT

/ROUTINE TO WRITE 0110
 /

W0110, 0
 TAD M20
 DCA COUNT /0
 DCA I 10 /1
 CMA /1
 DCA I 10 /1
 CMA /1
 DCA I 10 /1
 DCA I 10 /0
 ISZ COUNT /-16 OR -32
 JMP W0110+3 /SEE IIF END OF FIELD
 JMS I XKRBNK /EXIT
 JMP I W0110

0647 4505
 0650 5651
 0651 2255

0652 0000
 0653 1056
 0654 3072
 0655 3410
 0656 3410
 0657 7040
 0658 3410
 0661 7040
 0662 3410
 0663 2072
 0664 5255
 0665 4470
 0666 5667

0667 0000
 0670 1056
 0671 3072
 0672 7040
 0673 3410
 0674 7040
 0675 3410
 0676 3410
 0677 3410
 0700 2072
 0701 5272
 0702 4470
 0703 5667

0704 0000
 0705 1056
 0706 3072
 0707 3410
 0710 7040
 0711 3410
 0712 7040
 0713 3410
 0714 3410
 0715 2072
 0716 5307
 0717 4470
 0720 5704

/ROUTINE TO WRITE 1001

0721	0722	0723	0724	0725	0726	0727	0730	0731	0732	0733	0734	0735
1001	1002	1003	1004	1005	1006	1007	1008	1009	1010	1011	1012	1013

```

/ROUTINE TO WRITE 1001
/
1001,
/
TAD W20
DCA COUNT
CMA
DCA I 10
DCA I 10
DCA I 10
CMA
DCA I 10
ISZ COUNT
JMP W1001+3
JMS I XKRNK
JMP I W1001
/1
/0
/0
/1
/-16 TO -32
/LOOP
/SEP IF END OF FIELD
/EXIT

```

```

0736 0000 /ROUTINE TO READ ALL OF MEMORY 8 TIMES, COMPLEMENTING
0737 7281 /THE PATTERN EACH PASS. NO ERROR CHECKING IS DONE.
0740 1057 /RDALL, 0
0741 3072 /-32 DECIMAL
0742 7240 /COUNTS PASSES THRU MEMORY
0743 3010 /SET ADR, REGS. TO 777
0744 7340 /SEE IF FIELD HAS PROGRAM
0745 3011 /READ
0746 3073 /DONE
0747 4463 /WRITE BACK
0750 7410 /DONE 1 FIELD WHEN SKIP
0751 5360 /DONE 32 PASSES WHEN SKIP
0752 7220 /DO ANOTHER PASS
0753 1410 /EXIT
0754 7040 /READ AND CHECK FOR ERROR ROUTINE
0755 3411 /
0756 2073 *100
0757 5353 /RCHKA, 0
0760 2072 JMS I XSETU
0761 5342 JMS I XRALL
0762 5736 DCA MEMADR
      DCA I XBANK
      SKP
      JMP I RCHKA
      TAD KRXT
      DCA EXIT
      0
      0
      0
      0
      0
      NOP
      JMP RLOPA+1
      TAD *-2
      TAD KIK
      SZA CLA
      JMP ,+4
      TAD JMS5
      DCA RXIT-2
      JMP RCHKA+3
      NOP
1000 0000 /SET DF TO 1ST FIELD
1001 4451 /READ ALL, DON'T CHECK
1002 4475 /SET ADR, COUN TO 0
1003 3122 /SEE IF FIELD HAS PROGRAM
1004 4463 /WILL = JMS I XRD1, 2, 3, OR 4
1005 7410 /WILL = NOP OR JMS STALL
1006 5600 /NOP IF 0
1007 1235 /=JMS I XSALL
1008 3153 /READ SLOW
1009 0000
1010 0000
1011 0000
1012 0000
1013 0000
1014 0000
1015 0000
1016 7000
1017 5212
1018 1216
1019 1236
1020 1236
1021 1236
1022 7640
1023 5227
1024 1114
1025 3216
1026 5203
1027 7000

```

1331 1227
1331 3216
1332 4471
1333 5202
1334 5600

1335 1020
1336 1000

TAU , -1
DCA RXIT-2
JMS I XTRNK /SETUP FOR NEXT FIELD
JMP RCHKA+2
JMP I RCHKA /EXIT

/
KRXI, RXIT
K1, 1000

```

/SETUP ROUTINES FOR RCHK1
/
RCHK1, Z
TAD JMS1 /JMS1 = JMS I XRD1
DCA RLOPA
TAD JMS1
DCA RLOPA+1 /JMS1 = JMS I XRD1
TAD JMS2
DCA RLOPA+2
TAD JMS2
DCA RLOPA+3
TAD JMS1
DCA RLOPA+4
JMS RCHKA /GO READ
JMP I RCHK1 /EXIT

RCHK1C, W
TAD JMS2 /JMS2 = JMS I XRD2
DCA RLOPA
TAD JMS2
DCA RLOPA+1
TAD JMS1
DCA RLOPA+2
TAD JMS1
DCA RLOPA+3
TAD JMS2
DCA RLOPA+4
JMS RCHKA /GO READ
JMP I RCHK1C /EXIT

RCHK2, Z
TAD JMS1 /JMS1 = JMS I XRD1
DCA RLOPA
TAD JMS2 /JMS I XRD2
DCA RLOPA+1
TAD JMS2
DCA RLOPA+2 /JMS I XRD1
TAD JMS1
DCA RLOPA+3
TAD JMS1
DCA RLOPA+4
JMS RCHKA /GO READ
JMP I RCHK2 /EXIT

RCHK2C, Z
TAD JMS2
DCA RLOPA /JMS I XRD1
TAD JMS1
DCA RLOPA+1
TAD JMS1
DCA RLOPA+2
TAD JMS2
DCA RLOPA+3

```

```

1137 3210
1140 1110
1141 3211
1142 1110
1143 3212
1144 1111
1145 3213
1146 1111
1147 3214
1151 1110
1151 3215
1152 4200
1153 5637

1154 0000
1155 1111
1156 3211
1157 1111
1160 3212
1161 1110
1162 3213
1163 1110
1164 3214
1165 1111
1166 3215
1167 4200
1170 5654

1171 0000
1172 1110
1173 3211
1174 1111
1175 3212
1176 1111
1177 3213
1180 1110
1181 3214
1182 1110
1183 3215
1184 4200
1185 5671

1186 0000
1187 1111
1188 3211
1189 1110
1192 3212
1193 1110
1194 3213
1195 1111
1196 3214

```

117 1111
112 3215
114 4220
112 5725

TAD JMS2
DCA RLOPA+4
JMS RCHKA /GO READ
JMP I RCHK2C /EXIT

/ RCHK3, Z

1123 0000 TAD JMS4 /JMS I XRD4
 1124 1113 DCA RLOPA /JMS I XRD3
 1125 3211 TAD JMS3
 1126 1112 DCA RLOPA+1
 1127 3212 TAD JMS3 /JMS I XRD3
 1128 1112 DCA RLOPA+2
 1129 3213 TAD JMS4 /JMS I XRD4
 1130 1113 DCA RLOPA+3
 1131 3214 TAD JMS4
 1132 1113 DCA RLOPA+4
 1133 3214 JMS RCHKA /GO READ
 1134 1113 JMP I RCHK3 /EXIT

1123 0000
 1124 1113
 1125 3211
 1126 1112
 1127 3212
 1128 1112
 1129 3213
 1130 1113
 1131 3214
 1132 1113
 1133 3214
 1134 1113
 1135 3215
 1136 4222
 1137 5723

/ RCHK3C, Z

1140 0000 TAD JMS3 /JMS I XRD3
 1141 1112 DCA RLOPA /JMS I XRD4
 1142 3211 TAD JMS4
 1143 1113 DCA RLOPA+1
 1144 3212 TAD JMS4
 1145 1113 DCA RLOPA+2
 1146 3213 TAD JMS3
 1147 1112 DCA RLOPA+3
 1148 3214 TAD JMS3
 1149 1112 DCA RLOPA+4
 1150 3215 JMS RCHKA /GO READ
 1151 1112 JMP I RCHK3C /EXIT

1140 0000
 1141 1112
 1142 3211
 1143 1113
 1144 3212
 1145 1113
 1146 3213
 1147 1112
 1148 3214
 1149 1112
 1150 3215
 1151 1112
 1152 3215
 1153 4222
 1154 5747

/ *1200

/ RCHK4, Z

1200 0200 JMS I XFILD /JMS I XRD3
 1201 4531 TAD JMS3 /JMS I XRD3
 1202 1112 DCA I XLOPA
 1203 3637 TAD JMS3
 1204 1112 DCA I XLOPB
 1205 3640 TAD JMS4
 1206 1113 DCA I XLOPC
 1207 3641 TAD JMS4 /JMS I XRD4
 1208 1113 DCA I XLOPD
 1209 3642 TAD JMS3
 1210 1112 DCA I XLOPE
 1211 3643 JMS I XCFL
 1212 1112 JMS I XCHKA /GO READ
 1213 3643 JMP I RCHK4 /EXIT

1200 0200
 1201 4531
 1202 1112
 1203 3637
 1204 1112
 1205 3640
 1206 1113
 1207 3641
 1208 1113
 1209 3642
 1210 1112
 1211 3643
 1212 1112
 1213 3643
 1214 4644
 1215 4636
 1216 5600

/ RCHK4C, Z

1217 0000 JMS I XFILD /JMS I XRD4
 1218 4531 TAD JMS4
 1219 1113 DCA I XLOPA
 1220 3637 TAD JMS4
 1221 1113 DCA I XLOPB
 1222 3640 TAD JMS3

1217 0000
 1218 4531
 1219 1113
 1220 3637
 1221 1113
 1222 3640
 1223 1112

1226 3641
1227 1112
1230 3642
1231 1113
1232 3643
1233 4644
1234 4035
1235 5617

1236 1720
1237 1711
1240 1712
1241 1713
1242 1714
1243 1715
1244 1722

DCA I XL0PC
TAD JMS3
DCA I XL0PD
TAD JMS4
DCA I XL0PE
JMS I XCFL
JMS I XCHKA
JMP I RCHK4C /EXIT

/GO READ

XCHKA,
XLOPA,
XLOPR,
XL0PC,
XLOPD,
XLOPE,
XCFL,
RCHKA
RLOPA
RLOPA+1
RLOPA+2
RLOPA+3
RLOPA+4
CFLD

PAUSE

/BI-85 EXTENDED CHECKERBOARD - TAPE 2

/READ ROUTINES FOR 0111; 1100; 0110 AND 1001

```

1245 0170
1246 1256
1247 3972
1250 1260
1251 3973
1252 1141
1253 3974
1254 1522
1255 7340
1256 3522
1257 2074
1260 5254
1261 2073
1262 7410
1263 5266
1264 2122
1265 5252

1266 1122
1267 1260
1270 3710
1271 7100
1272 1410
1273 7440
1274 4521
1275 7100
1276 1410
1277 7440
1300 4521
1301 7120
1302 1410
1323 7040
1324 7440
1325 4521
1306 7120
1327 1410
1317 7340
1311 7440
1312 4521
1313 2072
1314 5320
1315 4470
1316 2122
1317 5645

1320 2122
1321 5250

RD1,
TAD M20 /-16
JCA COUNT /-4
TAD M4 /-4
JCA FLCNT /-8
TAD M12
JCA LOOP
TAD I MEMADR
CMA
JCA I MEMADR
ISZ LOOP
JMP +4
ISZ FLCNT
SKP +3
JMP +3
ISZ MEMADR
JMP CLOP1

CLOP1,
TAD I MEMADR
CMA
JCA I MEMADR
ISZ LOOP
JMP +4
ISZ FLCNT
SKP +3
JMP +3
ISZ MEMADR
JMP CLOP1

/COMPLEMENT 8 TIMES
/DOONE 4 ADRS. WHEN SKIP

TAD MEMADR
TAD M4 /SUBTRACT 4
DCA 10 /NOW USE AUTO-INDEX
CLL
TAD I 10
SZA /0
JMS I XRROR /PRINT ERROR
CLL
TAD I 10
SZA /0
JMS I XRROR /PRINT ERROR
STL
TAD I 10
CMA /1
SZA /PRINT ERROR
STL
TAD I 10
CMA /1
SZA /PRINT ERROR
JMS I XRROR /PRINT ERROR
JMP +4
JMS I XKBK /SEE IF END OF FIELD
ISZ MEMADR
JMP I RD1

ISZ MEMADR /KEEP READING
JMP RD1+3
    
```

```

/
RD2,
1322 2000
1323 1056
1324 3072
1325 1008
1326 3073

CLOP2,
1327 1141
1328 3074
1329 1522
1330 7040
1331 3522
1332 2074
1333 5331
1334 2073
1335 7410
1336 5343
1337 2122
1338 5327

/
1339 1122
1340 1060
1341 3010
1342 7120
1343 1410
1344 7040
1345 4521
1346 7120
1347 1410
1348 7040
1349 4521
1350 7120
1351 1410
1352 7040
1353 4521
1354 7120
1355 1410
1356 7040
1357 4521
1358 7120
1359 1410
1360 7040
1361 4521
1362 7120
1363 1410
1364 7040
1365 4521
1366 7120
1367 1410
1368 7040
1369 4521
1370 7120
1371 1410
1372 7040
1373 4521
1374 7120

/
1375 2122
1376 5325

```

/ RD2,

TAD M2
DCA COUNT
TAD M4
DCA FLCNT

/-16

/-4

/-R

/READ

CLOP2,

TAD M1
DCA LOOP
TAD I MEMADR
CMA
DCA I MEMADR
ISZ LOOP
JMP +4
ISZ FLCNT
SKP
JMP +3
ISZ MEMADR
JMP CLOP2

/COMPLEMENT 4 TIMES

/DONE 4 ADRS. WHEN SKIP

/INCREMENT ADDRESS

/

TAD MEMADR

TAD M4

DCA I0

STL

TAD I 10

CMA

SZX I XRROR

STL

TAD I 10

CMA

SZX I XRROR

CLL

TAD I 10

SZX I XRROR

JMS I XRROR

SZX I XRROR

CLL

TAD I 10

SZX I XRROR

JMS I XRROR

SZX I XRROR

ISZ COUNT

JMP +4

JMS I XKBK

ISZ MEMADR

JMP I RD2

ISZ MEMADR

JMP RD2+3

ISZ MEMADR

JMP RD2+3

ISZ MEMADR

JMP RD2+3

ISZ MEMADR

JMP RD2+3

ISZ MEMADR

JMP RD2+3

ISZ MEMADR

JMP RD2+3

ISZ MEMADR

JMP RD2+3

/KEEP READING

/SEE IF END OF FIELD

```

1430
1431 5204
1432 1156
1433 3072
1434 1760
1435 3073
1436 1141
1437 3074
1438 1522
1439 7040
1440 3522
1441 2074
1442 5207
1443 2073
1444 7410
1445 5221
1446 2122
1447 5205

1421 1122
1422 1760
1423 3710
1424 7100
1425 1410
1426 7440
1427 4521
1428 7120
1429 1410
1430 7040
1431 7440
1432 4521
1433 7120
1434 1410
1435 7040
1436 7440
1437 4521
1438 7100
1439 1410
1440 7440
1441 4521
1442 7100
1443 1410
1444 7440
1445 4521
1446 2072
1447 5253
1448 4470
1449 2122
1450 5600

1453 2122
1454 5203
/
*1400
/
R03,
/
TAD M20
DCA COUNT /-16
TAD M4 /-4
DCA FLCNT /-8
TAD M10 /READ
DCA LOOP
TAD I MEMADR
CMA
DCA I MEMADR
ISZ LOOP /COMPLEMENT 4 TIMES
JMP ,+4 /DONE 4 IF 0
ISZ FLCNT
SKP
JMP ,+3
ISZ MEMADR /DO NEXT
JMP CLOP3

TAD MEMADR
TAD M4
DCA I0 /USE AUTO-INDEX
CLL
TAD I 10
SZA /0
JMS I XRROR /PRINT ERROR
TAD I 10 /1
CMA
SZA /PRINT ERROR
JMS I XRROR
STL /1
TAD I 10 /PRINT ERROR
CMA
SZA /1
JMS I XRROR /PRINT ERROR
CLL /0
TAD I 10 /PRINT ERROR
SZA /PRINT ERROR
JMS I XRROR /PRINT ERROR
ISZ COUNT /0
JMP ,+4 /PRINT ERROR
JMS I XKBNK /SEE IF END OF FIELD
ISZ MEMADR
JMP I R03
ISZ MEMADR
JMP R03+3

```

1455	0100			
1456	1356	TAD M21	/-16	
1457	3072	DCA COUNT		
1460	1768	TAD M4	/-4	
1461	3376	DCA FLCNT		
1462	1141	TAD M1A	/-8	
1463	3374	DCA LOOP		
1464	1522	TAD I MEMADR		
1465	7348	CMA		
1466	3522	DCA I MEMADR		
1467	2374	ISZ LOOP	/COMPLEMENT 8 TIMES	
1470	5264	JMP *-4		
1471	2373	ISZ FLCNT	/DONE 4 ADRS, WHEN SKIP	
1472	7410	SKP		
1473	5276	JMP *+3		
1474	2122	ISZ MEMADR	/INCREMENT ADDRESS	
1475	5262	JMP CLOP4		
1476	1122	TAD MEMADR		
1477	1360	TAD M4		
1500	3310	DCA 10	/USE AUTO-INDEX	
1521	7120	STL		
1522	1410	TAD I 10	/1	
1523	7040	CMA		
1524	7440	SZA	/PRINT ERROR	
1525	4521	JMS I XRROR		
1526	7100	CLL		
1527	1410	TAD I 10	/0	
1510	7440	SZA	/PRINT ERROR	
1511	4521	JMS I XRROR		
1512	7100	CLL		
1513	1410	TAD I 10	/0	
1514	7440	SZA	/PRINT ERROR	
1515	4521	JMS I XRROR		
1516	7120	STL		
1517	1410	TAD I 10	/1	
1520	7040	CMA		
1521	7440	SZA	/PRINT ERROR	
1522	4521	JMS I XRROR		
1523	2072	ISZ COUNT		
1524	5330	JMP *+4		
1525	4470	JMS I XKBK	/SEE IF END OF FIELD	
1526	2122	ISZ MEMADR		
1527	5655	JMP I RD4		
1537	2122	ISZ MEMADR		
1531	5260	JMP RD4*3		

```

1600
1601
1602
1603
1604
1605
1606
1607
1610
1611
1612
1613
1614
1615
1616
1617
1620
1621
1622
1623

/ROUTINE TO CHECK FOR END OF FIELD
/
*1600
CBANK, 4
CLA
TAD 12
CMA
SZA CLA
JMP I CBANK /NOT DONE
JMP I EXIT /DONE

/ROUTINE TO SEE IF TESTED FIELD HAS PROGRAM
/
CBANK, 4
RIF
DCA SAVIF
RDF
CIA
TAD SAVIF
SZA CLA
JMP I CBANK /EQUAL IF AC=0
JMS I XTRNK /DOESN'T HAVE PROGRAM
JMP I CBANK /INCREMENT DATA FIELD
ISZ CBANK /TEST NEW FIELD
JMP I CBANK /DONE ALL CAUSE PROGRAM NOW
SAVIF, 4
JMP I CBANK /IN HIGHEST FIELD
/EXIT

```

```

1624 0000 /ROUTINE TO SET UP FOR NEXT FIELD
1625 7200 /
1626 6214 /NEXTBNK, 0
1627 7041 /READ DATA FIELD
1630 1124 /C(LAST1) = LAST IN TEST
1631 7640 /ALL DONE IF 0
1632 5230 /C(LAST1) = LAST IN TEST
1633 2224 /
1634 5242 /
1635 6214 /EXIT
1636 1034 /INCREMENT DATA FIELD
1637 1125 /ADD ,6201
1640 3241 /CHANGE TO NEW DATA FIELD
1641 5221 /
1642 7004 /CHECK SWITCH REGISTER
1643 7710 /LAS
1644 4527 /SPA CLA /CHEC HALT
1645 5624 /JMS I XHLT /GO HALT, SR0=1
1646 0000 /
1647 7220 /RESTORE DATA FIELD AND CHECK SR
1650 6214 /FIELD, 0
1651 3014 /CLA
1652 6224 /RDF
1653 1125 /DCA DATFLD /SAVE TESTED FIELD#
1654 3255 /TAD KODF
1655 6201 /DCA ,+1 /MAKE DATA AND INST FIELD EQUAL
1656 7204 /CDF AX
1657 5646 /CLA
1658 5646 /JMP I FIELD

```

```

1700
1701 7200
1704 7604
1700 3123
1703 7402
1704 7604
1705 3124
1706 1123
1707 3122
1708 1522
1711 3522
1712 1122
1713 7041
1714 1124
1715 7650
1716 5306
1717 2122
1720 5312
1721 7402

1722 0500
1723 7200
1724 1014
1725 1120
1726 3327
1727 6201
1730 7200
1731 5722

/ *1700
/ /START HERE TO LOOP ON ADDRESS
/
CLA
LAS
DCA FIRST1 /READ LOWER LIMIT
HLT /NOW SETUP UPPER LIMIT
LAS
DCA LAST1
TAD FIRST1
DCA MEMADR
TAD I MEMADR /READ
DCA I MEMADR /WRITE
TAD MEMADR
CIA
TAD LAST1
SNA CLA
JMP OVER
ISZ MEMADR
JMP WRLOP
HLT

/ CFLO,
0
CLA DATFLD /TEST FIELD
TAD KCDF
DCA ,+1
CDF 00 /RESTORE TEST FIELD
CLA
JMP I CFLO /EXIT

```

```

2000 /PRINT ERROR ROUTINE
2001 /
2002 *2'00'
2003 ERROR,
2004
2005 /READING 1'S IF LINK = 1
2006 /SAVE RAD DATA
2007 /SAVE GOOD DATA
2008 /OCTAL ADDRESS
2009 /RESTORE DATA FIELD
2010 /DATA FIELD
2011 /LAST = FIELD WITH LAST ERROR
2012 /SAME IF 0
2013 /DON'T STORE
2014 /TABLE POINTER
2015 /END OF TABLE IF = 0
2016 /RESTORE POINTER
2017 /INCREMENT POINTER
2018 /STORE IN TABLE
2019
2020 /SR2 ON A 1 = RING BELL
2021 /RING HELL
2022 /SR1 A 1 = NO PRINT
2023 /SET TO TESTED FIELD
2024
2025
2026
2027
2028
2029
2030
2031
2032
2033
2034
2035
2036
2037
2038
2039
2040
2041
2042
2043
2044
2045
2046
2047
2048
2049
2050
2051
2052
2053

```



```

2154 4354 JMS CRLF /CR,LF
2155 1214 TAU DATFLO /TEST NUMBER
2156 7012 RTR
2157 7110 RAR
2158 1142 TAU K260 /PRINT
2159 4346 JMS PRERR /-12 DECIMAL
2160 1140 TAD M14 /SPACE 12
2161 3274 DCA LOOP /OCTAL ADR,
2162 4315 JMS SPING /SAVE
2163 1365 TAD OCADR /PRINT
2164 3134 DCA CHAR /-8 DECIMAL
2165 4326 JMS PROCTL /SPACE 8
2166 1141 TAU M12
2167 3274 DCA LOOP
2168 4315 JMS SPING
2169 1364 TAD GOOD
2170 3134 DCA CHAR
2171 4326 JMS PROCTL /PRINT
2172 1261 TAD M5 /-5
2173 3074 DCA LOOP /SPACE 5
2174 4315 JMS SPING
2175 1363 TAD BAD
2176 3134 DCA CHAR /PRINT
2177 4326 JMS PROCTL
2178 1061 TAD M5
2179 3074 DCA LOOP
2180 4315 JMS SPING /SPACE 5
2181 1363 TAD BAD
2182 3134 DCA CHAR /PRINT
2183 4326 JMS PROCTL
2184 1061 TAD M5
2185 3074 DCA LOOP
2186 4315 JMS SPING /SPACE 5
2187 1062 TAD TNUM /TEST NUMBER
2188 4346 JMS PRERR /PRINT

/ SW0, LAS /CH CK SR0
SPA CLA /GO HALT
JMS I XHLT /EXIT
JMP EREXT

/ /PRINT SPACES
/ SPING, 0 TAD K240 /SPACE
TLS /PRINT
TSF
JMP M-1
ISZ LOOP
JMP SPING+2
CLA
JMP I SPING /EXIT

2111 7604
2112 7110
2113 4527
2114 5246

2115 0000
2116 1370
2117 6046
2120 6041
2121 5320
2122 2074
2123 5317
2124 7200
2125 5715

```

```

2126 0000 /PRINT OCTAL
2127 1060 /PROCTL, 0
2128 3074 /-4
2129 1134 /DIGIT COUNTER
2130 7104 POSITN, TAD M4
2131 7006 DCA LOOP
2132 3134 TAD CHAR
2133 1134 CLL RAL
2134 7004 RTL
2135 3134 DCA CHAR
2136 7004 TAD CHAR
2137 0133 RAL
2138 1142 AND K7
2139 4346 TAD K260
2140 2074 JMS PRERR
2141 5331 ISZ LOOP
2142 7200 JMP POSITN
2143 5726 CLA
2144 0000 JMP I PROCTL
2145 6046 /EXIT

2146 0000 /PRINT A NUMBER
2147 6046 /
2148 6041 PRERR, 0
2149 5350 TFS
2150 7200 JMP , -1
2151 5746 CLA
2152 0000 JMP I PRERR
2153 0000 /EXIT

2154 0000 /CARRIAGE RETURN, LINE FEED
2155 7200 /
2156 1143 CRLF, 0
2157 4346 CLA
2158 1371 TAD K215
2159 4346 JMS PRERR
2160 5754 TAD K212
2161 0000 JMS PRERR
2162 0000 JMP I CRLF

2163 0000 /
2164 0000 BAD, 0
2165 0000 GOOD, 0
2166 0000 OCADR, 0
2167 0207 LAST, 0
2168 0240 K207, 0
2169 0212 K240, 0
2170 0212 K212, 0

```

```

2200 /
2201 / *2200
2202 / /ROUTINE TO SET UP TO FIRST TEST FIELD
2203 /
2204 / SETU1, 0
2205 /   CLA
2206 /   TAD FIRST1 /FIRST TO TEST
2207 /   TAD KODF
2208 /   DCA ,+1
2209 /
2210 /   CDF 00 /CHANGE TO TEST FIELD
2211 /   JMP I SETU1 /EXIT
2212 /
2213 / /ROUTINE TO ACCEPT TEST LIMITS FROM
2214 / /KEYBOARD INPUT
2215 /
2216 / SLMTS, 0
2217 /   JMS I XFILD
2218 /   JMS I XCRLF /CR, LF
2219 /   JMS I XTLIM /PRINT TEST LIMITS
2220 /   JMS I XCRLF /CR, LF
2221 /   JMS KEYIN /GO ACCEPT INPUT
2222 /   JMS LEGAL /SEE IF IT'S LEGAL
2223 /   TAD CHAR
2224 /   AND K7 /MASK AC 9=11
2225 /   CLL RAL
2226 /   RTL /POSITION TO AC 6=8
2227 /   DCA FIRST1 /FIRST TO TEST
2228 /   JMS KEYIN /WAIT FOR COMMA
2229 /   TAD CHAR /GET INPUT
2230 /   CIA
2231 /   TAD K254 /OK IF 0
2232 /   SNA
2233 /   JMP ,+3 /PRINT QUESTION MARK
2234 /   JMP SLMTS+2
2235 /   JMS KEYIN /WAIT FOR 2ND
2236 /   JMS LEGAL /SEE IF IT'S LEGAL
2237 /   TAD CHAR
2238 /   AND K7 /MASK AC 9=11
2239 /   CLL RAL
2240 /   RTL /POSITION TO AC 6=8
2241 /   DCA LAST1 /LAST TO TEST
2242 /   TAD FIRST1
2243 /   CIA
2244 /   TAD LAST1
2245 /   SMA
2246 /   JMP OKAS /1ST IS > LAST IF NEG
2247 /   CLA

```

2252 1123
 2251 3134
 2252 1124
 2253 3123
 2254 1134
 2255 3124
 2256 7200
 2257 1124
 2260 7041
 2261 1123
 2262 7440
 2263 5273
 2264 1123
 2265 7041
 2266 1216
 2267 7640
 2270 5273
 2271 4752
 2272 5211
 2273 4323
 2274 1134
 2275 7041
 2276 1143
 2277 7450
 2300 5607
 2301 4344
 2302 5211
 2303 0000
 2304 6032
 2305 6031
 2306 5305
 2307 6036
 2310 3134
 2311 1134
 2312 4536
 2313 5703

TAD FIRST1
 DCA CHAR
 TAD LAST1
 DCA FIRST1 /LAST NOW IS FIRST
 TAD CHAR
 DCA LAST1 /FIRST IS NOW LAST
 CLA
 TAD LAST1
 CIA
 TAD FIRST1 /SEL IF EQUAL
 SZA /YFS IF 0
 JMP ALOK
 TAD FIRST1 /NOW SEE IF IT HAS PROGRAM
 CIA /CURRENT FIELD
 TAD INSELD /NO IF A 1
 SZA CLA
 JMP ALOK
 JMS I XLCAT /PRINT PROGRAM LOCATION
 JMP SLMTS+2 /AND START OVER
 JMS KEYIN /WAIT FOR C,R,
 TAD CHAR
 CIA
 TAD K215 /NOT A C.R, IF A SKIP
 SNA
 JMP I SLMTS /PRINT QUESTION MARK
 JMS QUERY /START OVER
 JMP SLMTS+2
 /
 KEYIN:
 KCC
 KSF
 JMP *-1
 KRB
 DCA CHAR
 TAD CHAR
 JMS I XPERR
 JMP I KEYIN

OKAS,

ALOK,

/
 KEYIN:

2314	0820		
2315	1134	LEGAL,	TAD CHAR
2316	7041		CIA
2317	1144		TAD K377
2320	7650		SNA CLA
2321	5211		JMP SLMTS+2
2322	1134		TAD CHAR
2323	0145		AND K370
2324	7041		CIA
2325	1142		TAD K260
2326	7650		SNA CLA
2327	5714		JMP I LEGAL
2330	1134		TAD CHAR
2331	7041		CIA
2332	1353		TAD K254
2333	7650		SNA CLA
2334	5714		JMP I LEGAL
2335	1134		TAD CHAR
2336	7041		CIA
2337	1143		TAD K215
2340	7650		SNA CLA
2341	5714		JMP I LEGAL
2342	4344		JMS QUERY
2343	5211		JMP SLMTS+2
2344	0000	QUERY,	
2345	4547		JMS I XCRLF
2346	1146		TAD K277
2347	4536		JMS I XPERR
2350	5744		JMP I QUERY
2351	2446		XTLIM,
2352	2400		XLOCAT,
2353	0254		K254,

/RUB-OUT IF 0

/A COMMA IF 0

/A C.R. IF 0

/START OVER /QUERY

/PRINT QUERY MARK

```

2400 2400
2401 2401
2402 2402
2403 2403
2404 2404
2405 2405
2406 2406
2407 2407
2410 2410
2411 2411
2412 2412
2413 2413
2414 2414
2415 2415
2416 2416
2417 2417
2420 2420
2421 2421
2422 2422
2423 2423
2424 2424
2425 2425
2426 2426
2427 2427
2430 2430
2431 2431
2432 2432
2433 2433
2434 2434
2435 2435
2436 2436
2437 2437
2440 2440
2441 2441
2442 2442
2443 2443
2444 2444
2445 2445

/PRINT FIELD PROGRAM IS IN
/
*2400
/LOCAT,
  CLA
  TAD INSFLD
  RTR
  RAR
  AND K7
  TAD K260
  DCA FLDN
  TAD PRGAM
  DCA 12
  TAD I 12
  SNA
  JMP I LOCAT
  JMS I XPERR
  JMP PLOCT

PLOCT,
  SNA
  JMP I LOCAT
  JMS I XPERR
  JMP PLOCT

/PRGAM,
  P
  R
  O
  G
  R
  A
  M
  I
  S
  I
  N
  F
  I
  E
  L
  D
  X
  TERMINATOR

/FLDN = PRGAM+25
  FLDN,
  0

```

```

2446 0000 /TLIMIT, 0 /PRINT TEST LIMITS
2447 7200 CLA TAD TSTL
2451 3012 DCA 12
2452 1412 TAD I 12
2453 7450 SNA /DONE IF 0
2454 5646 JMP I TLIMIT /DONE IF 0
2455 4536 JMS I XPERR
2456 5252 JMP PLIMIT

2457 2457 /TSTL,
2460 0324 /T
2461 0305 /E
2462 0323 /S
2463 0324 /T
2464 0240 /
2465 0314 /L
2466 0311 /I
2467 0315 /M
2470 0311 /I
2471 0324 /T
2472 0323 /S
2473 0000 /TERMINATOR

```

```

/HEADER ROUTINE
/PHDR, 0
2474 0000 JMS I XCRLF /CR, LF
2475 4547 TAD FILD
2476 1332 DCA 12
2477 3012 TAD I 12 /PRINT FIELD
2500 1412 SNA /DONE IF 0
2501 7450 JMP ,+3
2502 5305 JMS I XPERR
2503 4536 JMP PFILD
2504 5300 TAD M5
2505 1061 DCA LOOP
2506 3074 JMS I XPING /SPACE 5
2507 4537 TAD OILDR
2510 1341 DCA 12
2511 3012

```

2512	1412	P00DR,	TAD I 12			
2513	7452		SNA	/DONE IF 0	/PRINT OCTAL ADR	
2514	5317		JMP ,+3			
2515	4536		JMS I XPERR			
2516	5312	/	JMP P00DR			
2517	1061		TAD M5			
2521	3274		DCA LOOP	/SPACE 5		
2521	4537		JMS I XPING			
2522	1355		TAD GOOD			
2523	3812		DCA 12			
2524	1412	P00DR,	TAD I 12	/DONE IF 0	/PRINT GOOD	
2525	7452		SNA			
2526	5731		JMP I ,+3			
2527	4536		JMS I XPERR			
2530	5324		JMP P00DR			
2531	2600	/	RSPACE	/NEXT PAGE		
2532	2532					
2533	0316	/	FILD,			
2534	0311		306	/F		
2535	0305		311	/I		
2536	0314		505	/E		
2537	0304		314	/L		
2540	0000		304	/D		
2541	2541	/				
2542	0317		317	/O		
2543	0303		305	/C		
2544	0324		324	/T		
2545	0301		301	/A		
2546	0314		314	/L		
2547	0240		240			
2550	0301		301	/A		
2551	0304		304	/D		
2552	0322		322	/R		
2553	0256		256		/,	
2554	0200	/	0			
2555	2555					
2556	0327	/	GOOD,			
2557	0317		307	/G		
2560	0317		317	/O		
2561	0304		317	/O		
2562	0100		304	/D		
2562	0100		E			
2563	5674	/	EXHDR,			
			JMP I PHDR			


```

2620 2620 2626 2626
2621 1661 2627 0324
2622 3074 2628 0305
2623 4537 2629 0323
2624 1234 2630 0324
2625 3012 2631 0000
2626 1412 2632 2634
2627 7452 2633 0302
2628 5212 2634 0301
2629 4536 2635 0304
2630 5205 2636 0000
2631 1061 2637 2641
2632 3074 2638 2642
2633 4537 2639 2643
2634 1226 2640 2644
2635 3012 2641 2563
2636 1412 2642 2563
2637 7450 2643 2563
2638 5224 2644 2563
2639 4536 2645 2563
2640 5217 2646 2563
2641 4547 2647 2563
2642 5644 2648 2563

/ *2600
/ RSPACE,
TAD M5
DCA LOOP
JMS I XPING
TAD RADU
DCA I2
TAD I 12
SNA
JMP ,+3
JMS I XPERR
JMP PRAD
TAD M5
DCA LOOP
JMS I XPING
/SPACE 5
/PRINT BAD
/DONE IF 0

/ PTSTN,
TAD TSTN
DCA I2
TAD I 12
SNA
JMP ,+3
JMS I XPERR
JMP PTSTN
JMS I XGRLF
JMP I XPHDR
/PRINT TEST
/DONE IF 0
/CR, LF
/EXIT

/ TSTN,
324
305
323
324
0
/T
/E
/S
/T

/ BAOD,
302
301
304
0
/B
/A
/D

/ HALT,
0
HLT
JMP I HALT
/RESTART HERE OR RTRN1

/ XPHDR,
EXHDR

```

```

2645 0000
2646 4531
2647 4547
2650 1270
2651 3012
2652 1412
2653 7452
2654 5257
2655 4536
2656 5252
2657 6336
2660 6231
2661 5265
2662 6036
2663 4536
2664 5645
2665 7604
2666 3033
2667 5260

2672 2670
2671 0323
2672 0305
2673 0324
2674 0325
2675 0320
2676 0240
2677 0323
2700 0322
2701 0000

2702 0000
2703 4531
2704 4316
2705 3074
2706 2074
2707 5306
2710 1014
2711 1125
2712 3313
2713 6201
2714 7200
2715 5702

/
/
/ *WAIT HERE TO SETUP SR, TYPE CARRIAGE RETURN
/ AFTER SETTING SR,
/
SETSP, 2
JMS I XFILD /RESTORE DATA FIELD
JMS I XCRLF /CR, LF
TAD STSR
DCA 12
TAD I 12 /PRINT SETUP SR
SNA /DONE IF 0
JMP ,+3
JMS I XPERR
JMP PSTSR
KRB
KSF ,+4
JMP ,+4
KRB
JMS I XFERR
JMP I SETSR
LAS
DCA MCWA
JMP WTCR

/ STSR,
323 /S
305 /E
324 /T
325 /U
320 /P
240 /S
323 /R
522
0

/ STALL,
0
JMS I XFILD
JMS GENRAN /GET ANOTHER
DCA LOOP
ISE LOOP /18.5 MS MAX,
JMP ,+1
TAD DATFLD
TAD KCDF
DCA ,+1
CDF 02
CLA
JMP I STALL /EXIT

```

2716 8700
 2717 1354
 2720 7241
 2721 1342
 2722 7640
 2723 5333
 2724 1355
 2725 3342
 2726 1341
 2727 7124
 2730 7432
 2731 1150
 2732 3341
 2733 1742
 2734 1341
 2735 3742
 2736 1742
 2737 2342
 2740 5716

GENRAN, 2
 TAD RANTAB
 CIA
 TAD RANDEX
 SZA CLA
 JMP RANTAD-1
 TAD TBLRAN
 JCA RANDEX
 TAD RANCON
 CLL RAL
 SZL
 TAD K1
 DCA RANCON
 TAD I RANDEX
 RANTAD,
 DCA I RANCON
 TAD I RANDEX
 ISZ RANDEX
 JMP I GENRAN

2741 1234
 2742 2753
 2743 4321
 2744 1416
 2745 5363
 2746 6060
 2747 3035
 2750 2572
 2751 3237
 2752 0214
 2753 0000
 2754 2753
 2755 2743
 2756 0177

/
 RANCON, 1234
 RANDEX, RANTBL+10
 RANTBL, 4321
 1416
 5363
 6060
 3035
 2572
 3237
 0214
 0
 RANTAB, 2753
 TBLRAN, RANTBL
 K177, 177

```

3000 /ROUTINE TO DETERMINE FIELD FOR RELOCATION
3001 /
3002 / *3000
3003 /
3004 CMOVE, JMS I XFILD /SET DF TO CURRENT FIELD
3005 7600
3006 TAD ERTBL
3007 DCA ERWRD /SETUP ERROR TABLE POINTER
3008 TAD FIRST1 /FIRST TESTED FIELD
3009 CIA
3010 TAD LAST1 /LAST TESTED FIELD
3011 SNA CLA /DON'T MOVE IF EQUAL
3012 JMP I XRTN /START OVER
3013 RAR
3014 SZL /FIRST MOVE IF A SKIP
3015 JMP I XTMV /SETUP FOR NEXT MOVE
3016 IAC /SET BIT 11
3017 DCA FLAGS
3018
3019 TAD LAST1 /LAST TO TEST = 1ST MOVE
3020 DCA INSFLD /NEW CURRENT FIELD
3021 TAD INSFLD
3022 TAD *10 /SUBTRACT 1 FROM NEW CURRENT
3023 DCA NXLOC /NXLOC=DEEST,N FOR NEXT TIME
3024 RIF
3025 CIA
3026 TAD INSFLD
3027 SNA CLA /IS NEXT SAME AS CURRENT
3028 JMP SUB1 /YES, TRY NEXT LOWER FIELD
3029
3030

```

/CHECK FOR ERROR IN NEW FIELD
/

3031 2122 /CERR, ISZ ERWRD /POINTER*1
 3032 1201 TAD CMOVE+1
 3033 7041 CIA
 3034 1422 TAD I ERWRD
 3035 7650 SNA CLA /NO ERRORS RECORDED IF 0
 3036 5310 JMP STMV /INITIALIZE MOVE

3037 1422 CVXT, TAD I ERWRD
 3040 7041 CIA
 3041 1016 TAD INSFLD
 3042 7650 SNA CLA /ERROR IN NEW FIELD IF 0
 3043 5253 JMP EQUAL
 3044 1022 TAD ERWRD
 3045 7041 CIA
 3046 1021 TAD ENTBL
 3047 7650 SNA CLA /ENTBL=ERWRD+10
 3050 5310 JMP STMV /TABLE DONE IF 0
 3051 2022 ISZ ERWRD /INITIALIZE MOVE
 3052 5237 JMP CNXT /POINTER*1

/EQUAL, TAD I ERWRD /GET ERROR FIELD
 CIA
 TAD FIRST1
 SNA CLA /DON'T MOVE IF = TO FIRST
 JMP I XRTN /START OVER
 TAD I ERWRD
 SNA CLA /IS IT FIELD 0?
 JMP SUB1 /YES
 TAD INSFLD /CURRENT NEXT
 TAD M10 /SUBTRACT 1 FROM DF
 DCA VXLOC

/SUB1, TAD ERTBL /RESTORE TABLE POINTER
 DCA ERWRD
 TAD NXLOC
 CIA
 TAD INSFLD
 SNA CLA /NEXT = CURRENT NEXT IF 0
 JMP EQUAL /NEW CURRENT FIELD
 TAD NXLOC
 DCA INSFLD
 TAD INSFLD
 CIA
 TAD FIRST1 /IS IT = LOWEST FIELD
 SNA CLA /YES
 JMP CKERR /CURRENT NEW FIELD
 TAD INSFLD /SUBTRACT 1 FROM DF
 TAD M10 /NEXT FIELD LOWER
 DCA VXLOC

4/24/68 13:41,0

JMP CKERR

3107 5231

```

3110 7200 CLA
3111 1020 TAD ERTBL
3112 3022 DCA ERWRD
3113 6224 RIF
3114 3723 DCA I YSRCE
3115 1723 TAD I XSRCE
3116 7041 CIA
3117 1016 TAD INSFLD
3120 7650 SNA CLA
3121 5530 JMP I XRTN
3122 5724 JMP I XMVE

```

/RESTORE TABLE POINTER

```

/DCN'T MOVE IF EQUAL
/START OVER
/GO MOVE

```

```

/ STMV,
/ XSRCE,
  XMVE,
  XT MV,
/

```

```

3123 3327 SOURCE
3124 3307 MOVE
3125 3200 NXTMV

```

```

3200 7600
/ NXTMV, 7600
          RIF
          DCA SOURCE
          ISZ ERWRD
          TAD NXTMV
          CIA
          TAD I ERWRD
          SNA CLA
          JMP STNXT
          TAD I ERWRD
          CIA
          TAD NXLOC
          SNA CLA
          JMP SUR2
          TAD ERWRD
          CIA
          TAD ENTB
          SNA CLA
          JMP STNXT
          ISZ ERWRD
          JMP CKNXT

          TAD ENTB
          DCA ERWRD
          TAD NXLOC
          CIA
          TAD INSFLD
          SNA CLA
          JMP CKNT
          TAD NXLOC
          CIA
          TAD FIRST1
          SZA CLA
          JMP STNX
          JMP MVRK
          TAD NXLOC
          CIA
          TAD FIRST1
          SNA CLA
          JMP NXTHI
          TAD NXLOC
          DCA INSFLD
          TAD INSFLD
          TAD M12
          DCA NXLOC
          JMP MOVE

          TAD ENTB
          DCA ERWRD
          TAD NXLOC
          CIA
          TAD INSFLD
          SNA CLA
          JMP CKNT
          TAD NXLOC
          CIA
          TAD FIRST1
          SZA CLA
          JMP STNX
          JMP MVRK
          TAD NXLOC
          CIA
          TAD FIRST1
          SNA CLA
          JMP NXTHI
          TAD NXLOC
          DCA INSFLD
          TAD INSFLD
          TAD M12
          DCA NXLOC
          JMP MOVE

```

```

3201 7601
3204 6224
3209 3327
3213 2722
3224 1220
3225 7041
3226 1422
3227 7650
321 5225
3211 1422
3212 7041
3213 1151
3214 7650
3215 5255
3216 1222
3217 7041
322 1221
3221 7650
3222 5225
3223 2022
3224 5211

3225 1220
3226 3022
3227 1151
3232 7041
3231 1016
3232 7650
3233 5242
3234 1151
3235 7041
3236 1123
3237 7640
3240 5247
3241 5302
3242 1151
3243 7041
3244 1123
3245 7650
3246 5275
3247 1151
3250 3016
3251 1016
3252 1141
3253 3151
3254 5307

```

```

          /CURRENT FIELD
          /POINTER +1

          /NO ERRORS RECORDED IF 0
          /INITIALIZE MOVE

          /ERROR IN NEW FIELD IF 0
          /TRY NEXT LOWER FIELD

          /DONE WITH TABLE IF 0
          /INITIALIZE MOVE
          /POINTER +1

          /NEXT=CURRENT IF 0

          /NEXT = LOWEST IF 0
          /MOVE TO LOWEST TEST FIELD

          /NEXT = LOWEST IF 0
          /SETUP TO MOVE TO HIGHEST
          /NEXT LOWER FIELD
          /IS NOW CURRENT FIELD

          /SUBTRACT 1 FROM NEW
          /NEW NEXT LOWER FIELD
          /GO MOVE

```



```

3255 1220 TAD ERTBL
3256 3222 DCA ERWRD
3257 1151 TAD NXLOC
3260 7450 SNA
3261 5530 JMP I XRTV
3262 1141 TAD M1P
3263 3151 DCA NXLOC
3264 1151 TAD NXLOC
3265 7141 CIA
3266 1016 TAD INSFLD
3267 7640 SZA CLA
3270 5203 JMP CHXT
3271 1151 TAD NXLOC
3272 7450 SNA
3273 5203 JMP CHXT
3274 5262 JMP SUB2*5

3275 1124 TAD LAST1
3276 3151 DCA NXLOC
3277 1124 TAD LAST1
3300 3016 DCA INSFLD
3301 5203 JMP CHXT

3302 1151 TAD NXLOC
3303 3016 DCA INSFLD
3304 6224 RIF
3305 3327 DCA SOURCE
3306 3015 DCA FLAGS

```

```

/
SUR2,
/RESTORE TABLE POINTER
/NEXT LOWER FIELD
/=FIELD 0 IF 0
/START OVER CAN'T MOVE
/SUBTRACT 1
/NOW = 2 FIELDS LOWER

/CURRENT FIELD
/ARE THEY EQUAL
/NO
/YES
/DOES IT = FIELD 0
/NO

/VERY LAST TO TEST
/MAKE IT NEXT FIELD

/CLEAR BIT 11

```

```

/ROUTINE TO RELOCATE 4K FIELDS
/MOVE,
TAD KCDF /6201
TAD SOURCE /CURRENT FIELD
DCA SOURCE /SOURCE NOW = CDF N
TAD KCDF /6201
TAD INSFLO /NEW FIELD
DCA DESTN /DESTN NOW = CDF N
TAD SOURCE
CIA
TAD DESTN
SNA CLA
JMP I XRTN
CMA
DCA 10 /SOURCE COUNT
CMA
DCA 11 /DESTINATION COUNT
DCA LOOP /4K COUNTER
SOURCE, 2 /WILL = CDF N
DESTN, 2 /TAKE FROM HERE
DCA I 11 /PUT IN HERE
ISZ LOOP /DOVE 4K WHEN SKIP
JMP SOURCE /KEEP MOVING
TAD KCIF /6202
TAD INSFLO /NEW FIELD
DCA ,+1
CIF 00 /CHANGE TO NEW FIELD
JMP I XRTN /EXIT TO RTN1 IN
/NEW FIELD
$

```

3327 1125
3314 1327
3311 3327
3312 1125
3313 1116
3314 3331
3315 1327
3316 7041
3317 1331
3320 7650
3321 5530
3322 7040
3323 3010
3324 7040
3325 3011
3326 3074
3327 0000
3330 1410
3331 0000
3332 3411
3333 2074
3334 5327
3335 1126
3336 1016
3337 3340
3340 6202
3341 5530

THERE ARE NO ERRORS

SYMBOL TABLE

ALAW	0220
ALOK	2273
BAD	2163
BAD0	2634
REGIN	0200
RSPACE	2600
CHANK	1627
CDF	6201
CDOM1	7760
CFLD	1722
CHAR	0184
CHXT	3223
CIF	6272
CKRnk	1600
CKERR	3231
CKNT	3242
CKNXT	3211
CLOP1	1252
CLOP2	1327
CLOP3	1405
CLOP4	1462
CMOVF	3000
CNXT	3237
COUNT	0272
CRLF	2154
DATAFLD	0214
DESTN	3331
ENTBL	0021
EPRNT	2054
EQUAL	3253
EREXT	2246
ERROR	2020
ERTBL	0020
ERWRD	0022
EXAM1	0241
EXAM2	0246
EXAM3	0253
EXAM4	0260
EXHDR	2563
EXIT	0153
EXT1	0322
EXT1C	0343
EXT2	0423
EXT2C	0445
EXT3	0475
EXT3C	0517
EXT4	0623
EXT4C	0647
FILD	1646
FILD	2532
FIRST1	0123
FLAGS	0015
FLCNT	0073

SYMBOL TABLE

FLCN	2444
FLOAT	2217
GENRAN	2716
GOOD	2555
GOOD	2164
HALT	2641
INSFLD	2216
JMS1	2112
JMS2	2111
JMS3	2112
JMS4	2113
JMS5	2114
KCOF	2125
KCIF	2126
KEYIN	2323
KRXT	2335
KXT1	2345
KXT1C	2346
KXT2	2452
KXT2C	2451
KXT3	2522
KXT3C	2523
KXT4	2645
KXT4C	2646
K1	2152
K1K	2236
K1A	2234
K100	2242
K177	2756
K20	2042
K200	2237
K207	2167
K212	2171
K215	2143
K240	2172
K254	2353
K260	2142
K261	2252
K262	2053
K263	2054
K264	2055
K277	2146
K370	2145
K377	2144
K40	2241
K400	2036
K7	2133
K740	2235
LAST	2166
LASTY	2154
LAST1	2124
LEGAL	2314
LOCAT	2422

SYMBOL TABLE

LOOP	2074
MCWA	2233
MEMAIR	2122
MOVE	3397
MVRK	3392
M1W	1141
M14	1140
M2W	2456
M4	2660
M4P	2257
M5	2261
NXLOC	2151
NXTBANK	1624
NXTHI	3275
NXTMV	3280
OCADR	2165
OKAS	2256
OTLDR	2541
OVFR	1706
PBAD	2605
PFILED	2500
PGOOD	2524
PHDR	2474
PLIMIT	2452
PLUCT	2412
POGDR	2512
POSITN	2131
PRERR	2146
PRGAM	2417
PROCTL	2126
PSTR	2652
PTSTN	2617
QUERY	2344
RANCON	2741
RANDEX	2742
RANTAB	2754
RANTAD	2734
RANTBL	2743
RCHKA	1000
RCHK1	1037
RCHK1C	1054
RCHK2	1071
RCHK2C	1126
RCHK3	1123
RCHK3C	1140
RCHKA	1272
RCHK4C	1217
RDALL	2736
RDF	6214
RDL0P	2753
RD1	1245
RD2	1322
RD3	1422

SYMBOL TABLE

RD4	1455
RIF	6224
RLOPA	1211
RSTRY1	2211
RTA1	2213
RXIT	1020
SAVIF	1623
SETSF	2645
SETU1	2220
SUMTS	2227
SOURCE	3327
SPIVG	2115
STALL	2722
STMV	3110
STNX	3247
STXYT	3225
STSR	2672
SUR1	3266
SUR2	3255
SW1	2111
SW2	2242
SW3	2033
TBLRAN	2755
TD*2A	0106
TD*4A	0107
TL*MT	2446
TNUM	2262
TSTL	2457
TSTN	2626
TST1	2300
TST1C	2324
TST2	2400
TST2C	2425
TST3	2452
TST3C	2477
TST4	2600
TST4C	2625
WRLOP	1710
WTCR	2660
W0011	2652
W0112	2724
W1021	2721
W1122	2667
XBANK	2263
XCFL	1244
XCHKA	1236
XCHK1	2276
XCHK1C	2277
XCHK2	2122
XCHK2C	2121
XCHK3	2102
XCHK3C	2103
XCHK4	2174

SYMBOL TABLE

XCHK4C	0105
XCRLF	0147
XFILD	0131
XHDR	0135
XHLT	0127
XIT1	0320
XIT1C	0341
XIT2	0421
XIT2C	0443
XIT3	0473
XIT3C	0515
XIT4	0621
XIT4C	0643
XIBNK	0070
XLCAT	2352
XLMTS	0043
XLOPA	1237
XLOPB	1240
XLOPC	1241
XLOPD	1242
XLOPE	1243
XMOVE	0050
XMVE	3124
XPERR	0136
XPHDR	2644
XPING	0137
XPRER	0132
XRALL	0075
XR01	0115
XR02	0116
XR03	0117
XR04	0120
XRROR	0121
XRTN	0130
XSALL	0152
XSETU	0051
XSRCE	3123
XSTSR	0277
XTBANK	0071
XTLIM	2351
XTMV	3125
XTST1	0044
XTST2	0045
XTST3	0046
XTST4	0047
X0011	0064
X0110	0066
X1001	0067
X1100	0065

SYMBOL TABLE

DATAFLD	0014
FLAGS	0015
INSFLD	0016
FLOAD	0017
ERTBL	0020
ENTBL	0021
ERRR	0022
MCVA	0033
K1A	0034
K74B	0035
K402	0036
K200	0037
K102	0040
K44	0041
K20	0042
XLMTS	0043
XTST1	0044
XTST2	0045
XTST3	0046
XTST4	0047
XMOVE	0050
XSETJ	0051
K261	0052
K262	0053
K263	0054
K264	0055
M20	0056
M40	0057
M4	0060
M5	0061
INUM	0062
XBANK	0063
X0011	0064
X1100	0065
X0110	0066
X1001	0067
XKRK	0070
XTRK	0071
COUNT	0072
FLCNT	0073
LOOP	0074
XRALL	0075
XCHK1	0076
XCHK1C	0077
XCHK2	0100
XCHK2C	0101
XCHK3	0102
XCHK3C	0103
XCHK4	0104
XCHK4C	0105
TDM20	0106
TDM40	0107
JMS1	0110

SYMBOL TABLE

JMS2	0111
JMS3	0112
JMS4	0113
JMS5	0114
XRD1	0115
XRD2	0116
XRD3	0117
XRD4	0120
XRROR	0121
MEMADR	0122
FIRSTI	0123
LASTI	0124
KCDF	0125
KCIF	0126
XHLT	0127
XRTN	0130
XFILE	0131
XPRER	0132
K7	0133
CHAR	0134
XHDR	0135
XPERR	0136
XPING	0137
M14	0140
M10	0141
K260	0142
K215	0143
K377	0144
K370	0145
K277	0146
XCRLF	0147
K1	0150
NXLOC	0151
XCALL	0152
EXIT	0153
LASTX	0154
BEGIN	0200
RSTRT1	0211
RTN1	0213
ALAW	0220
EXAM1	0241
EXAM2	0246
EXAM3	0253
EXAM4	0260
XSTSP	0277
TST1	0300
XIT1	0320
EXT1	0322
TST1C	0324
XIT1C	0341
EXT1C	0343
KXT1	0345
KXT1C	0346

SYMBOL TABLE

TST2	0420
XIT2	0421
EXT2	0423
TST2C	0425
XIT2C	0443
EXT2C	0445
KXT2	0450
KXT2C	0451
TST3	0452
XIT3	0473
EXT3	0475
TST3C	0477
XIT3C	0515
EXT3C	0517
KXT3	0522
KXT3C	0523
TST4	0620
XIT4	0621
EXT4	0623
TST4C	0625
XIT4C	0643
KXT4	0645
KXT4C	0646
EXT4C	0647
W0011	0652
W1100	0667
W0110	0724
W1001	0721
RDALL	0736
RULOP	0753
CDON1	0760
RCHKA	1020
RLOPA	1011
RXIT	1022
KRXT	1035
K1K	1036
RCHK1	1237
RCHK1C	1054
RCHK2	1271
RCHK2C	1106
RCHK3	1123
RCHK3C	1140
RCHK4	1220
RCHK4C	1217
XCHKA	1236
XLOPA	1237
XLOPR	1240
XLOPC	1241
XLOPD	1242
XLOPE	1243
XCFL	1244
RD1	1245
CLOP1	1252

SYMBOL TABLE

RD2	1322
CLOP2	1327
RD3	1400
CLOP3	1405
RD4	1455
CLOP4	1462
CKRANK	1600
CHANK	1607
SAVIF	1623
NXTBANK	1624
FIELD	1646
OVER	1706
WRLOP	1710
CFLO	1722
ERROR	2000
SW2	2033
SW1	2042
EREXT	2046
EPRNT	2054
SW0	2111
SPING	2115
PROCTL	2126
POSITN	2131
PRFRR	2146
CRUF	2154
RAO	2163
GOOD	2164
OCADR	2165
LAST	2166
K207	2167
K240	2170
K212	2171
SETU1	2200
SLMTS	2207
OKAS	2256
ALOK	2273
KEYIN	2303
LEGAL	2314
QUERY	2344
XTLIM	2351
XLCAT	2352
K254	2353
LOCAT	2400
PLOCT	2412
PRCAM	2417
FLDN	2444
TLIMT	2446
PLIMT	2452
TSTL	2457
PHDR	2474
PFILD	2500
PQCDR	2512
PG000	2524

SYMBOL TABLE

FILD	2532
OTLDR	2541
GOOD	2555
EXHDR	2563
BSPCE	2600
PHAD	2625
PTSTN	2617
TSTN	2626
RA00	2634
HALT	2641
XPHDR	2644
SETSR	2645
PSTSR	2652
WTCR	2660
STSR	2670
STALL	2702
GENRAN	2716
RANTAD	2734
RANCON	2741
RANDEX	2742
RANTBL	2743
RANTAB	2754
TBLRAN	2755
K177	2756
CMOVE	3000
CKERR	3031
CVXT	3037
EQUAL	3053
SUR1	3066
STMV	3110
XSRCE	3123
XMVE	3124
XTMV	3125
NXTMV	3200
CHNXT	3203
CKNXT	3211
STNXT	3225
CKNT	3242
STNX	3247
SUR2	3255
NXTHI	3275
MVRK	3302
MOVE	3307
SOURCE	3327
DESTN	3331
CDF	6201
CIF	6202
RDF	6214
RIF	6224

MAINDEC EVALUATION REQUEST

After sufficient familiarization with the operation and documentation of this MAINDEC, please indicate your assessment of the following areas and return this form to Digital Equipment Corporation.

IDENTIFICATION: MAINDEC NO. _____ Program Title _____

USAGE: Used by: Field Service Production Other _____

Frequency of Usage: Daily Weekly Monthly

SUGGESTIONS FOR IMPROVEMENT

1. Are the program loading and operating instructions: clear? , incomplete? , difficult to follow?

2. Do the error reports and program documentation provide sufficient diagnostic information. in all cases? , in most cases? , in very few cases? . Suggestions for improvement:

3. Is the program effective in isolating malfunctions: in all cases? , in most cases? , in very few cases? . Would additional Scope loops or Switch Register control be helpful? _____
Suggestions for improvement:

4. Does the program ever fail to detect malfunctions exposed by other software? _____
Were Margins used? _____ Please describe malfunction in detail:

5. Does the program ever report non-existent malfunctions? _____
Please indicate erroneous report and any pertinent operating conditions:

6. Does this MAINDEC ever expose malfunctions in the Central Processor or other peripheral units not detected by the appropriate MAINDEC? _____
Please describe malfunction and MAINDEC(S) used:

7. Does the document provide a general understanding of the functional programming requirements of the system? Good , Fair , None . Would a general description of programming requirements increase the effectiveness of this MAINDEC? _____

Remarks:

Fold Here

Do Not Tear - Fold Here and Staple

FIRST CLASS
PERMIT NO. 33
MAYNARD, MASS.

BUSINESS REPLY MAIL
NO POSTAGE STAMP NECESSARY IF MAILED IN THE UNITED STATES

Postage will be paid by:

digital

Digital Equipment Corporation
Diagnostic Programming Group
146 Main Street, Building 12
Maynard, Massachusetts 01754

