

dow
a7

IDENTIFICATION

Product Code: MAINDEC-08-D1HA-D
Product Name: PDP-8, 8/1, Extended Memory
Address Test
Date Created: March 13, 1968
Maintainer: Diagnostic Group
Author: J. W. Richardson

COPYRIGHT © 1968
DIGITAL EQUIPMENT CORPORATION



1. ABSTRACT

The PDP-8, 8/I Extended Memory Address Test tests all of memory not occupied by the program to make sure that each location can be uniquely addressed. This is performed by a series of four tests. The first two tests write the address and complement address of each memory location into itself, and then checks the contents of each location to make sure each is correct. The third test first sets all of memory not occupied by the program to all ones, and then writes a word of all zeroes, except for one bit, into each location and checks for error. The fourth test is similar except that a word of all ones, except for one bit, is written into each location and checks for error.

2. REQUIREMENTS

2.1 Equipment

A standard PDP-8 or 8/I with a minimum of 8K words of core memory.

2.2 Storage

The program requires locations 0010 to 2534 octal.

2.3 Preliminary Programs

The Binary loader must be in locations 7756-7776 octal. Also, all diagnostics for a basic PDP-8 or 8/I must have previously been run successfully.

3. LOADING PROCEDURE

3.1 Method

- a. Turn off the Teletype reader.
- b. Set the SR to 7777.
- c. Press LOAD ADDRESS, and then START.
- d. Place the Binary tape in the 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 211 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 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 8/1 equipped with a maximum of 28K.

4.3.1 Setup SR - After the test limits is specified, the program will print "SETUP SR". For normal program operation, the SR must be set to equal 0000₍₈₎. 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₍₈₎, 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₍₈₎. Refer to section 8.2, applications, for switch settings provided for trouble-shooting.

5.3 Subroutine Abstracts

5.3.1 The Patterns Used for Testing - The program executes a series of four tests on core memory. Each test writes a unique pattern, and checks each location for error.

Test 1 writes the value of each address into itself, from the lowest order to the highest order 4K field under test. The address pattern is then read and checked for error in the same direction, i.e., from the lowest to highest field under test. The pattern is then read and checked for error in the reverse directions, i.e., from the highest to the lowest field under test. When reading in the reverse direction, each location is repeatedly read a random number of times before reading the next location. The minimum number of reads per location is one, and the maximum is 20 octal. Error checking is done after each read.

Test 1 then writes the same address pattern again, only this time starting with the highest 4K field under test, and decrementing to the lowest under test. Reading and error checking is then performed as previously described.

Test 2 writes the ones complement value of each location into itself, and proceeds to read and check in the same manner as test 1.

Test 3 and 4 both write a "sliding" bit pattern throughout memory. The difference between the two tests being that test 3 rotates a single bit equal to 1, and test 4 rotates a single 0. Both tests use the following test sequence. The sequence is repeated 12 times, resulting in each bit of every memory location being complemented.

- a. Write all 1's into all of memory.
- b. Write a sliding 1 or 0 pattern into one 4K field.
- c. Read and test the 4K field in the forward direction only.
- d. Repeat steps b and c 11 more times before testing the next sequential field in the same manner.

After test 4 is completed on all memory fields, the program is relocated, and restarted with test 1.

6. ERRORS

Starting the program from address 200 will give a header print-out after the SR has been set up. 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.
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 2042. 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 set up 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 20 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 MA will equal 2042 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 timeout will also cause a halt at location 2042. Proceed exactly as described in the above paragraph.

8.2.2 Inhibit Error Printout - SR 1 - Placing SR 1 on a 1 causes all error printouts 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.

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 SR 7 and 8 - Not Used

8.2.6 Inhibit Program Relocation - SR 9 - The program normally relocates automatically as indicated by the INSTRUCTION FIELD indicators. To retain the program in its current 4K field, place SR 9 on a 1 at any time. Changing SR 9 to a 0 will permit relocation to resume.

8.2.7 SR 10 - Not used

8.2.8 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.

8.2.9 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 1137.
- b. Press LOAD ADDRESS
- c. Set the DATA FIELD switches to equal the 4K field number to test, and set the SR to equal the first address of the group.
- d. Press START. A halt will occur at 1142. Set the SR to equal the last address of the group.
- e. 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 Extended Memory address test is intended for use with a PDP-8, 8/I or 8/S equipped with the extended memory option. A total of four tests are executed by the program. Each test writes a unique pattern into core memory and then checks for error. The patterns were chosen so as to aid the operator in the event of addressing errors.

The program automatically relocates itself to each memory field under test to ensure that all fields may be correctly referenced from any field.

Control of the program is given to the operator by means of the SR. The operator may halt the program, inhibit error printouts, substitute the TTY BELL for error indication, halt after printout, select any one or a combination of the four tests, inhibit program relocation, and create an automatic program restart.

10. LISTING

XSETU,	SETU1
K261,	261
K262,	262
K263,	263
K264,	264
M20,	7760

0054	1400
0055	0261
0056	0262
0057	0263
0060	0264
0061	7760

0062	7740	M4,	7740
0063	7774	M4,	7774
0064	7773	M5,	7773
0065	0000	TNUM,	0
0066	1066	XBANK,	CBANK
0067	1103	XTBANK,	NXTBANK
0070	0000	COUNT,	0
0071	0000	FLCNT,	0
0072	0000	LOOP,	0
0073	1200	XRROR,	ERROR
0074	0000	MEMADR,	0
0075	0000	FIRST1,	0
0076	0000	LAST1,	0
0077	6201	KCDF,	6201
0100	6202	KCIF,	6202
0101	2041	XHLT,	HALT
0102	0213	XRTN,	RTN1
0103	1125	XFILD,	FIELD
0104	1346	XPRER,	PRERR
0105	0007	K7,	7
0106	0000	CHAR,	0
0107	1674	XHDR,	PHDR
0110	1346	XPERR,	PRERR
0111	1315	XPING,	SPING

0112	7765	M13,	7765
0113	7764	M14,	7764
0114	7770	M10,	7770
0115	0260	K260,	260
0116	0215	K215,	215
0117	0377	K377,	377
0120	0370	K370,	370
0121	0277	K277,	277
0122	1354	XCRLF,	CRLF
0123	0001	K1,	1
0124	0000	NXLOC,	0
0125	0000	EXIT,	0
0126	0000	PATR,	0
0127	0000	COMPR,	0
0130	0654	XWRF,	WFR1
0131	0720	XFWD,	FRK1
0132	1000	XBAK,	RBK1
0133	0671	XWRB,	WRB1
0134	0662	XWRF2,	WFR2
0135	0736	XFWD2,	FRK2
0136	1030	XBAK2,	RBK2
0137	0704	XWRB2,	WRB2
0140	1057	XONES,	ONES
0141	0000	SHIFT,	0
0142	7777	M1,	7777
0143	0000	DATA,	0

0200
0201
0202
0203
0204
0205
0206
0207
0210

*200
BEGIN,

IOF
CLA
RIF
DCA INSFLD
DCA FLAGS
JMS I XLMTS
JMS I XSTSR
JMS I XHDR
JMP RTN1

/PI OFF

/CLEAR PROGRAM FLAGS
/SETUP TEST LIMITS
/SETUP SR
/PRINT HEADER

/RESTART HERE
/

0211
0212
0213
0214
0215
0216
0217
0220
0221
0222
0223
0224
0225
0226
0227

RSTRT1,
RTN1,
ALAW,
JMS I XLMTS
JMS I XSTSR
RIF
DCA INSFLD
JMS I XFILD
TAD M10
DCA LOOP
7600
TAD ALAW
ISZ ERWRD
DCA I ERWRD
ISZ LOOP
JMP ALAW
TAD ERTBL
DCA ERWRD

/SET TEST LIMITS
/SETUP SR
/READ INSTRUCTION FIELD
/CURRENT FIELD

/-10

```

0230 1037 /EXAMINE SR
0231 0041 TAD MCWA
0232 7440 AND K740
0233 5237 SZA
0234 1037 JMP EXAM1
0235 1041 TAD MCWA
0236 3037 DCA MCWA
0237 7200 CLA
0240 1037 TAD MCWA
0241 0042 AND K400
0242 7440 SZA
0243 5447 JMP I XTST1
0244 7200 CLA
0245 1037 TAD MCWA
0246 0043 AND K200
0247 7440 JMP I XTST2
0250 5450 SZA
0251 7200 CLA
0252 1037 TAD MCWA
0253 0044 AND K100
0254 7440 SZA
0255 5451 JMP I XTST3
0256 7200 CLA
0257 1037 TAD MCWA
0260 0045 AND K40
0261 7440 JMP I XTST4
0262 5452 JMS I XFILD
0263 4503 LAS
0264 7604 AND K1
0265 0123 SZA
0266 7440 JMP RSTR1
0267 5211 LAS
0270 7604 AND K20
0271 0046 SZA
0272 7440 JMP RTN1
0273 5213 JMP I XMOVE
0274 5453
0275 2045 / XTSR, SETSR

```

/MADK 3,4,5 AND 6
/JO ALL IF 4

/SET ALL TEST BITS
/SAVE

/TEST 1 IF NO SKIP

/TEST 2 IF NO SKIP

/TEST 3 IF NO SKIP

/TEST 4 IF NO SKIP

/RESTORE DATA FIELD

/CHECK SR 11

/INHIBIT MOVE IF A 1

/GO RELOCATE


```

0276 4454 JMS I XSETU /SET DF TO 1ST FIELD
0277 1055 TAD K261 /TEST NUMBER
0300 3065 DCA TNUM
0301 3074 DCA MEMADR
0302 3126 DCA PATR
0303 4466 JMS I XBANK /SET ADR, COUNT TO 0
0304 7410 SKP /SEE IF FIELD HAS PROGRAM
0305 5313 JMP TST1A /NO, BEGIN WRITING
0306 4530 JMS I XWRF /WRITE FORWARDS
0307 4531 JMS I XFWD /READ FORWARDS
0310 4532 JMS I XBAK /READ BACKWARDS
0311 4467 JMS I XTBNK /SETUP FOR NEXT FIELD
0312 5301 JMP TST1+3

/
/ NOW WRITE BACKWARDS
/
TST1A, JMS I XSETU /SET DF TO 1ST FIELD
CLA CMA /SET ADR, COUNT TO 7777
DCA MEMADR /COMPARE WORD
DCA PATR /SEE IF FIELD HAS PROGRAM
JMS I XBANK /BEGIN WRITING
SKP EXAM2 /NEXT TEST
JMS I XWRB /WRITE BACKWARDS
JMS I XFWD /READ FORWARDS
JMS I XBAK /READ BACKWARDS
JMS I XTBNK /SETUP FOR NEXT FIELD
JMP TST1A+1 /NEXT TEST
JMP EXAM2

```

```

/TEST 2, WRITE COMPLEMENT ADDRESS
/PATTERN FORWARD AND BACKWARD.
/
TST2,   JMS I XSETU   /SET DF TO 1ST FIELD
        TAD K262      /TEST NUMBER
        DCA TNUM
        DCA MEMADR   /SET ADDRESS COUNT TO 0
        CMA
        DCA PATR     /COMPARE WORD
        JMS I XBANK  /SEE IF FIELD HAS PROGRAM
        SKP          /WRITE
        JMP TST2A    /NEXT TEST
        JMS I XWRF2
        JMS I XFWD2  /READ FORWARDS
        JMS I XBAK2  /READ BACKWARDS
        JMS I XTBNK  /SETUP FOR NEXT FIELD
        JMP TST2+3
    0330 4454
    0331 1056
    0332 3065
    0333 3074
    0334 7040
    0335 3126
    0336 4466
    0337 7410
    0340 5346
    0341 4534
    0342 4535
    0343 4536
    0344 4467
    0345 5333
    
```

```

/NOH WRITE BACKWARDS
/
TST2A, JMS I XSETU   /SET DF TO 1ST FIELD
        CLA CMA
        DCA MEMADR   /SET ADDRESS COUNT TO 7777
        CMA
        DCA PATR     /COMPARE WORD
        JMS I XBANK  /SEE IF FIELD HAS PROGRAM
        SKP          /WRITE
        JMP EXAM3    /NEXT TEST
        JMS I XWR02  /WRITE BACKWARDS
        JMS I XFWD2  /READ FORWARDS
        JMS I XBAK2  /READ BACKWARDS
        JMS I XTBNK  /SETUP FOR NEXT FIELD
        JMP TST2A+1 /NEXT TEST
        JMP EXAM3
    0346 4454
    0347 7240
    0350 3074
    0351 7040
    0352 3126
    0353 4466
    0354 7410
    0355 5251
    0356 4537
    0357 4535
    0360 4536
    0361 4467
    0362 5347
    0363 5251
    
```

/TEST3, WRITE ALL 1'S, THEN SLIDE A SINGLE 1 THRU EACH
 /WORD; READ AND WRITE IN THE FORWARD DIRECTION ONLY,
 /

0400	JMS I XSETU	/SET DF TO 1ST FIELD
0401	TAD K263	
0402	DCA TNUM	
0403	IAC	/SET BIT 11
0404	DCA PATR	/COMPARE WORD
0405	IAC	
0406	DCA COMPR	/SAVE FIRST BIT
0407	DCA MEMADR	/SET ADDRESS COUNT TO 0
0410	JMS I XBANK	/SEE IF FIELD HAS PROGRAM
0411	SKP	/WRITE
0412	JMP EXT3	
0413	JMS I XONES	/WRITE 1'S
0414	TAD M14	/-12 DECIMAL
0415	DCA SHIFT	
0416	TAD PATR	
0417	DCA I MEMADR	
0420	ISZ MEMADR	/DONE 4K WHEN SKIP
0421	SKP	
0422	JMP RSL1	/READ AND COMPARE
0423	ISZ SHIFT	/CHECK FOR 12 POSITIONS
0424	JMP ,+4	/POSITION NEXT BIT
0425	TAD COMPR	/START OVER WITH 11
0426	DCA PATR	
0427	JMP WROT	
0430	TAD PATR	
0431	CLL RAL	
0432	SZL	
0433	IAC	/START NEXT WITH BIT 11
0434	DCA PATR	
0435	JMP WROT*2	

WROT,

0400	0454
0401	1057
0402	3065
0403	7001
0404	3126
0405	7001
0406	3127
0407	3074
0410	4466
0411	7410
0412	5301
0413	4540
0414	1113
0415	3141
0416	1126
0417	3474
0420	2074
0421	7410
0422	5236
0423	2141
0424	5230
0425	1127
0426	3126
0427	5214
0430	1126
0431	7104
0432	7430
0433	7001
0434	3126
0435	5216

```

/READ AND TEST ROUTINE FOR SLIDING ONE
/
RSL1:  TAD COMPR      /FIRST POSITION
      DCA PATR
      DCA MEMADR    /SET ADDRESS COUNT TO 0
      TAD M14      /-12 DECIMAL
      DCA SHIFT
      TAD I MEMADR
      DCA DATA    /SAVE
      TAD DATA
      CIA          /2'S COMPLEMENT
      TAD PATR
      SZA CLA     /OK IF 0
      JMS I XRROR /PRINT ERROR
      ISZ MEMADR  /DONE IF SKIP
      SKP CKDN1  /DONE
      JMP CKDN1  /CHECK FOR 12 POSITIONS
      ISZ SHIFT  /POSITION NEXT BIT
      JMP ,+4    /START OVER WITH 11
      TAD COMPR
      DCA PATR
      JMP RROT
      TAD PATR
      CLL RAL
      SZL
      IAC
      DCA PATR
      JMP RROT+2
0436 1127
0437 3126
0440 3074
0441 1113
0442 3141
0443 1474
0444 3143
0445 1143
0446 7041
0447 1126
0450 7640
0451 4473
0452 2074
0453 7410
0454 5270
0455 2141
0456 5262
0457 1127
0460 3126
0461 5241
0462 1126
0463 7104
0464 7430
0465 7001
0466 3126
0467 5243

```

```

/CKDN1:  TAD COMPR
      CLL RAL      /DONE ALL 12 IF NO SKIP
      SZL
      JMP ,+4
      DCA PATR
      TAD PATR
      JMP WROT=6   /WRITE IN SAME FIELD AGAIN
/
      JMS I XTBNK  /SETUP FOR NEXT FIELD
      JMP TST3+3
      JMP I ,+1   /NEXT TEST
      EXAM4
0470 1127
0471 7104
0472 7430
0473 5277
0474 3126
0475 1126
0476 5206
0477 4467
0500 5203
0501 5702
0502 0256

```

/TEST 4. WRITE ALL 1'S, THEN SLIDE A SINGLE 0 THRU EACH
/WORD. WRITE AND READ IN THE FORWARD DIRECTION ONLY.
/

0503	4454	JMS I XSETU	/SET DF TO 1ST FIELD
0504	1060	TAD K264	/TEST NUMBER
0505	3065	DCA TNUM	
0506	7001	IAC	
0507	3127	DCA COMPR	
0510	1127	TAD COMPR	/COMPARE WORD
0511	7040	CMA	
0512	3126	DCA PATR	
0513	3074	DCA MEMADR	/SET ADDRESS COUNT TO 0
0514	4466	JMS I XBANK	/SEE IF FIELD HAS PROGRAM
0515	7410	SKP	/WRITE
0516	5751	JMP I XT4	
0517	4540	JMS I XONES	/WRITE 1'S
0520	7120	STL	
0521	1113	TAD M14	/-12 DECIMAL
0522	3141	DCA SHIFT	
0523	1126	TAD PATR	
0524	3474	DCA I MEMADR	
0525	2074	ISZ MEMADR	/DONE 4K WHEN SKIP
0526	7410	SKP	
0527	5752	JMP I XSL0	/READ AND COMPARE
0530	2141	ISZ SHIFT	/CHECK FOR 12 POSITIONS
0531	5336	JMP .+5	/POSITION NEXT BIT
0532	1127	TAD COMPR	/START OVER WITH BIT 11
0533	7040	CMA	
0534	3126	DCA PATR	
0535	5320	JMP WR1T	
0536	1126	TAD PATR	
0537	7004	RAL	
0540	7420	SNL	
0541	5344	JMP .+3	
0542	3126	DCA PATR	
0543	5323	JMP WR1T+3	
0544	7200	CLA	
0545	7001	IAC	
0546	7040	CMA	
0547	7120	STL	
0550	5342	JMP .+6	
0551	0263	EXAM4+5	
0552	0600	XSL0,	

WRIT,

XT4,
XSL0,

```

0600 1127
0601 7040
0602 3126
0603 3074
0604 1113
0605 3141
0606 1474
0607 3143
0610 1143
0611 7040
0612 1126
0613 7040
0614 7640
0615 4473
0616 2074
0617 7410
0620 5242
0621 2141
0622 5227
0623 1127
0624 7040
0625 3126
0626 5204
0627 1126
0630 7120
0631 7004
0632 7420
0633 5236
0634 3126
0635 5206
0636 7200
0637 7001
0640 7040
0641 5234
0642 1127
0643 7104
0644 7420
0645 5652
0646 4467
0647 5653
0650 5651
0651 0263
0652 0507
0653 0506

*600
/READ AND TEST ROUTINE FOR SLIDING ZERO
/
RSL0, TAD COMPR /1ST POSITION
CMA
DCA PATR
DCA MEMADR /SET ADR, COUNT TO 0
TAD M14 /12 DECIMAL
DCA SHIFT
TAD I MEMADR /READ
DCA DATA /SAVE
TAD DATA
CMA
TAD PATR
CMA
SZA CLA /MUST=0
JMS I XRROR /PRINT ERROR
ISZ MEMADR /DONE WHEN SKIP
SKP
JMP CKDN0
ISZ SHIFT /CHECK FOR 12 POSITIONS
JMP I+5 /START OVER WITH 11
TAD COMPR
CMA
DCA PATR
JMP RR1T
TAD PATR
STL
RAL
SNL
JMP I+3
DCA PATR
JMP RR1T+2
CLA
IAC
CMA
JMP I+5
/
CKDN0, TAD COMPR
CLL RAL
SNL
JMP I XRT4
/
JMS I XTBNK /SETUP FOR NEXT FIELD
JMP I XST4
JMP I I+1 /DONE ALL TESTS
EXAM4+5
/
XRT4, TST4+4
XST4, TST4+3

```

```

0654 0000
0655 1074
0656 3474
0657 2074
0660 5255
0661 5654

/ WRITE FORWARD ROUTINE FOR TST1
/
WFR1,
  0 TAD MEMADR /WRITE C(MEMADR) INTO
  DCA I MEMADR /SAME ADDRESS
  ISZ MEMADR /DONE WHEN SKIP
  JMP WFR1+1
  JMP I WFR1 /EXIT

/ WRITE FORWARD ROUTINE FOR TST2
/
WFR2,
  0 TAD MEMADR /WRITE COMPLEMENT OF
  CMA /C(MEMADR) INTO SAME
  DCA I MEMADR
  ISZ MEMADR /DONE 4K WHEN SKIP
  JMP WFR2+1
  JMP I WFR2 /EXIT

/ WRITE BACKWARD ROUTINE FOR TST1A
/
WRB1,
  0 DCA LOOP /4K COUNTER
  TAD MEMADR /INITIALLY=7777
  DCA I MEMADR
  ISZ LOOP /DONE 4K WHEN SKIP
  SKP /EXIT
  JMP I WRB1
  TAD MEMADR
  TAD M1
  DCA MEMADR /SUBTRACT 1 FROM ADDRESS
  JMP WRB1+2

/ WRITE BACKWARDS ROUTINE FOR TST2A
/
WRB2,
  0 DCA LOOP /4K COUNTER
  TAD MEMADR /INITIALLY=7777
  CMA
  DCA I MEMADR /DONE 4K WHEN SKIP
  ISZ LOOP
  SKP /EXIT
  JMP I WRB2
  TAD MEMADR
  TAD M1
  DCA MEMADR /SUBTRACT 1 FROM ADDRESS
  JMP WRB2+2

```

/READ FORWARD ROUTINE FOR TST1 AND TST1A

0720	0000	Ø	TAD I MEMADR	
0721	1474		DCA DATA	/SAVE
0722	3143		TAD DATA	
0723	1143		CIA	/COMPARE WORD
0724	7041		TAD PATR	
0725	1126		Ø	/MUST EQUAL Ø
0726	7640		SZA CLA	/PRINT ERROR
0727	4473		JMS I XRROR	/DONE 4K WHEN SKIP
0730	2074		ISZ MEMADR	
0731	7410		SKP	/EXIT
0732	5720		JMP I RFR1	
0733	2126		ISZ PATR	
0734	5321		JMP RFR1*1	/SHOULD NEVER GET HERE
0735	5720		JMP I RFR1	

/READ FORWARD ROUTINE FOR TST2 AND TST2A

0736	0000	Ø	TAD I MEMADR	
0737	1474		DCA DATA	/SAVE
0740	3143		TAD DATA	
0741	1143		CIA	/COMPARE WORD
0742	7041		TAD PATR	
0743	1126		Ø	/MUST=Ø
0744	7640		SZA CLA	/PRINT ERROR
0745	4473		JMS I XRROR	/DONE 4K WHEN SKIP
0746	2074		ISZ MEMADR	
0747	7410		SKP	/EXIT
0750	5736		JMP I RFR2	
0751	1126		TAD PATR	/SUBTRACT 1
0752	7041		CIA	
0753	7040		CHA PATR	
0754	3126		JMP RFR2*1	
0755	5337			


```

1000
1001 0000
1002 3072
1003 7040
1004 3074
1005 7040
1006 3126
1007 4665
1010 1474
1011 3143
1012 1143
1013 7041
1014 1126
1015 7640
1016 4473
1017 2071
1020 5207
1021 2072
1022 7410
1023 5600
1023 1074
1024 1142
1025 3074
1026 1074
1027 5205

```

```

*1000
/READ BACKWARD ROUTINE FOR ISI1 AND ISI1A
/
RBK1,
0 DCA LOOP /4K COUNTER
CMA
DCA MEMADR /SET ADR, COUNT TO 7777
CMA
DCA PATR /COMPARE WORD
JMS I XSALL
TAD I MEMADR /READ /SAVE
DCA DATA
TAD DATA
CIA
TAD PATR
SZA CLA /MUST=0
JMS I XRROR /PRINT ERROR
ISZ FLCNT /RANDOM LOOP
JMP LBK1
ISZ LOOP /DONE 4K WHEN SKIP
SKP
JMP I RBK1 /EXIT

TAD MEMADR
TAD M1
DCA MEMADR /SUBTRACT 1 FROM ADDRESS
TAD MEMADR
JMP LBK1=2

```

```

/READ BACKWARD ROUTINE FOR TST2 AND TST2A
/
RbK2,
0 DCA LOOP /4K COUNTER
CMA MEMADR /SET ADR, COUNT TO 7777
DCA PATR /COMPARE WORD
JMS I XSALL /READ /SAVE
TAD I MEMADR /SAVE
DCA DATA
TAD DATA
CIA
TAD PATR
SEA CLA /MUST=0
JMS I XRROR /PRINT ERROR
ISZ FLCNT /RANDOM LOOP
JMP LbK2 /DONE 4K WHEN SKIP
ISZ LOOP
SKP
JMP I RbK2 /EXIT
ISZ PATR /COMPARE WORD
TAD MEMADR
TAD M1
DCA MEMADR
JMP LbK2=1 /LOOP

```

1030 0000
1031 3072
1032 7040
1033 3074
1034 3126
1035 4665
1036 1474
1037 3143
1040 1143
1041 7041
1042 1126
1043 7640
1044 4473
1045 2071
1046 5236
1047 2072
1050 7410
1051 5630
1052 2126
1053 1074
1054 1142
1055 3074
1056 5235

```

/ROUTINE TO WRITE ONES IN ONE FIELD
/
ONES,
0 CLA CMA
DCA I MEMADR /DONE 4K WHEN SKIP
ISZ MEMADR
JMP ONES+1 /EXIT
JMP I ONES

```

1057 0000
1060 7240
1061 3474
1062 2074
1063 5260
1064 5657
1065 2102

```

/ XSALL, STALL

```

PAUSE

/EXTENDED MEMORY ADDRESS TEST - TAPE 2
/ROUTINE TO SEE IF TESTED FIELD HAS PROGRAM

1066	0000
1067	6224
1070	3302
1071	6214
1072	7041
1073	1302
1074	7640
1075	5666
1076	4467
1077	5666
1100	2266
1101	5666
1102	0000

/	CBANK,	0
	RIF	
	DCA	SAVIF
	RDF	
	CIA	
	TAD	SAVIF
	SZA	CLA
	JMP	I CBANK
	JMS	I XTBANK
	JMP	I CBANK
	ISZ	CBANK
	JMP	I CBANK
	SAVIF,	0

	/READ INST, FIELD
	/SAVE
	/READ DATA FIELD
	/EQUAL IF AC=0
	/DOESN'T HAVE PROGRAM
	/INCREMENT DATA FIELD
	/TEST NEW FIELD
	/DONE ALL CAUSE PROGRAM NOW
	/IN HIGHEST FIELD
	/EXIT

1103	0000	/ROUTINE TO SET DF FOR NEXT FIELD
1104	7200	/NEXTBNK, 0
1105	6214	CLA
1106	7041	RDF
1107	1076	CIA
1110	7640	TAD LAST1
1111	5314	SEA CLA
1112	2303	JMP ,+3
1113	5321	ISZ NXTBNK
1114	6214	JMP ,+6
1115	1040	RDF
1116	1077	TAD K10
1117	3320	TAD KCDF
1120	6201	DCA ,+1
		CDF 00
		/CHANGE TO NEW DATA FIELD
		/CHECK SWITCH REGISTER
1121	7604	LAS
1122	7710	SPA CLA
1123	4501	JMS I XHLT
1124	5703	JMP I NXTBNK
		/CHECK HALT
		/GO HALT, SR0=1
		/EXIT
		/INCREMENT DATA FIELD
		/ADD ,6201
		/HEAD DATA FIELD
		/C(LAST1) = LAST TO TEST
		/ALL DONE IF 0
		/EXIT
		/INCREMENT DATA FIELD
		/SAVE TESTED FIELD#
		/RESTORE DATA FIELD AND CHECK SR
		FIELD, 0
1125	0000	CLA
1126	7200	RDF
1127	6214	DCA DATELD
1130	3020	RIF
1131	6224	TAD KCDF
1132	1077	DCA ,+1
1133	3334	CDF 00
1134	6201	CLA
1135	7200	JMP I FIELD
1136	5725	/MAKE DATA AND INST FIELD EQUAL


```

1200 /PRINT ERROR ROUTINE
1201 /
1202 *1200
1203 ERROR,
1204
1205 TAD DATA
1206 DCA BAD
1207 TAD PATR
1208 DCA GOOD
1209 TAD MEMADR
1210 DCA OCADR
1211 CMA
1212 DCA FLCNT
1213 DCA JMS I XFILD
1214 TAD DATFLD
1215 CIA
1216 TAD LAST
1217 SNA CLA
1218 JMP SW2
1219 TAD DATFLD
1220 DCA LAST
1221 TAD ERWRD
1222 CIA
1223 TAD ENTL
1224 SZA CLA
1225 JMP +3
1226 TAD ERTBL
1227 DCA ERWRD
1228 TAD DATFLD
1229 ISZ ERWRD
1230 DCA I ERWRD
1231
1232
1233 LAS
1234 RTL
1235 SMA CLA
1236 JMP SW1
1237 TAD K207
1238 JMS PRERR
1239 JMP SW0
1240 LAS
1241 RAL
1242 SMA CLA
1243 JMP EPRNT
1244 TAD DATFLD
1245 TAD KCDF
1246 DCA +1
1247 CDF 00
1248 CLA
1249 JMP I ERROR
1250
1251
1252
1253
1200 /SAVE BAD DATA
1201
1202 /SAVE GOOD DATA
1203
1204 /OCTAL ADDRESS
1205
1206 /RESTORE DATA FIELD
1207 /DATA FIELD
1208
1209 /LAST = FIELD WITH LAST ERROR
1210 /SAME IF 0
1211 /DON'T STORE
1212
1213 /TABLE POINTER
1214
1215 /END OF TABLE IF = 0
1216
1217 /RESTORE POINTER
1218
1219 /INCREMENT POINTER
1220 /STORE IN TABLE
1221
1222 /SR2 ON A 1 = RING BELL
1223
1224 /RING BELL
1225
1226 /SR1 A 1 = NO PRINT
1227
1228 /SET TO TESTED FIELD
1229
1230
1231
1232
1233
1234
1235
1236
1237
1238
1239
1240
1241
1242
1243
1244
1245
1246
1247
1248
1249
1250
1251
1252
1253

```

```

1254 4354 JMS CRLF /CR,LF
1255 1020 TAD DATFLD /TEST NUMBER
1256 7012 RTH
1257 7010 RAR
1260 1115 TAD K260 /PRINT
1261 4346 JMS PRERR /-12 DECIMAL
1262 1113 TAD M14 /SPACE 12
1263 3072 DCA LOOP /OCTAL ADR,
1264 4315 JMS SPING /SAVE
1265 1365 TAD OCADR /PRINT
1266 3106 DCA CHAR /-8 DECIMAL
1267 4326 JMS PROCTL /SPACE 8
1270 1114 TAD M10
1271 3072 DCA LOOP
1272 4315 JMS SPING
1273 1364 TAD GOOD
1274 3106 DCA CHAR
1275 4326 JMS PROCTL /PRINT
1276 1064 TAD M5 /=5
1277 3072 DCA LOOP /SPACE 5
1300 4315 JMS SPING
1301 1363 TAD BAD
1302 3106 DCA CHAR
1303 4326 JMS PROCTL /PRINT
1304 1064 TAD M5
1305 3072 DCA LOOP
1306 4315 JMS SPING
1307 1065 TAD TNUM
1310 4346 JMS PRERR /SPACE 5
        /TEST NUMBER
        /PRINT

```

```

1311 7604 /SW0,
1312 7710 SPA CLA /CH CK SR0
1313 4501 JMS I XHLT /GO HALT
1314 5246 JMP EREXT /EIT

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

```



```

1326 0000 /PRINT OCTAL
1327 1063 /
1330 3072 PROCTL, 0 TAD M4 /-4
1331 1106 DCA LOOP /DIGIT COUNTER
1332 7104 TAD CHAR
1333 7006 CLL RAL
1334 3106 RTL CHAR
1335 1106 TAD CHAR
1336 7004 RAL
1337 0105 AND K7
1340 1115 TAD K260 /MAKE ASCII
1341 4346 JMS PRERR /PRINT ONE
1342 2072 ISE LOOP /DO NEXT
1343 5331 JMP POSITN
1344 7200 CLA
1345 5726 JMP I PROCTL /EXIT

```

```

/PRINT A NUMBER

```

```

1346 0000
1347 6046
1350 6041
1351 5350
1352 7200
1353 5746

```

```

PRERR, 0
      TLS
      TSF
      JMP ,=-1
      CLA
      JMP I PRERR /EXIT

```

```

/CARRIAGE RETURN, LINE FEED

```

```

1354 0000
1355 7200
1356 1116
1357 4346
1360 1371
1361 4346
1362 5754

```

```

CRLF, 0
      CLA
      TAD K215 /CR
      JMS PRERR
      TAD K212 /LF
      JMS PRERR
      JMP I CRLF

```

```

1363 0000
1364 0000
1365 0000
1366 0000
1367 0207
1370 0240
1371 0212

```

```

BAD, 0
GOOD, 0
OCADR, 0
LAST, 0
K207, 207
K240, 240
K212, 212

```

```

1400
/ *1400
/ /ROUTINE TO SET DF TO FIRST TEST FIELD
/
SETU1, 0
  CLA
  TAD FIRST1 /FIRST TO TEST
  TAD KCDF
  DCA ,+1
CDF 00 /CHANGE TO TEST FIELD
JMP I SETU1 /EXIT
/ROUTINE TO ACCEPT TEST LIMITS FROM
/KEYBOARD INPUT
/
SLM1, 0
  JMS I XFILD
  JMS I XCRLF /CR, LF
  JMS I XTLLIM /PRINT TEST LIMITS
  JMS I XCRLF /CR, LF
  JMS KEYIN /GO ACCEPT INPUT
  JMS LEGAL /SEE IF IT'S LEGAL
  TAD CHAR
  AND K7 /MASK AC 9-11
  CLL RAL /POSITION TO AC 6-8
  RTL /FIRST TO TEST
  DCA FIRST1 /WAIT FOR COMMA
  JMS KEYIN /GET INPUT
  TAD CHAR
  CIA /OK IF 0
  TAD K254
  SNA
  JMP ,+3 /PRINT QUESTION MARK
  JMS QUERY
  JMP SLM1+2 /WAIT FOR 2ND
  JMS KEYIN /SEE IF IT'S LEGAL
  JMS LEGAL
  TAD CHAR
  AND K7 /MASK AC 9-11
  CLL RAL /POSITION TO AC 6-8
  RTL /LAST TO TEST
  DCA LAST1
  TAD FIRST1
  CIA
  TAD LAST1
  SMA OKAS
  JMP OKAS
  CLA

```

```

1400 0000
1401 7200
1402 1075
1403 1077
1404 3205
1405 6201
1406 5600
1407 0000
1410 4503
1411 4522
1412 4751
1413 4522
1414 4303
1415 4314
1416 1106
1417 0105
1420 7104
1421 7006
1422 3075
1423 4303
1424 1106
1425 7041
1426 1353
1427 7450
1430 5233
1431 4344
1432 5211
1433 4303
1434 4314
1435 1106
1436 0105
1437 7104
1440 7006
1441 3076
1442 1075
1443 7041
1444 1076
1445 7500
1446 5256
1447 7200

```

1450 1075
 1451 3106
 1452 1076
 1453 3075
 1454 1106
 1455 3076
 1456 7200
 1457 1076
 1460 7041
 1461 1075
 1462 7440
 1463 5273
 1464 1075
 1465 7041
 1466 1023
 1467 7640
 1470 5273
 1471 4752
 1472 5211
 1473 4303
 1474 1106
 1475 7041
 1476 1116
 1477 7450
 1500 5607
 1501 4344
 1502 5211
 1503 0000
 1504 6032
 1505 6031
 1506 5305
 1507 6036
 1510 3106
 1511 1106
 1512 4510
 1513 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 /SEE IF EQUAL
 SZA /YES IF 0
 JMP ALOK /NOW SEE IF IT HAS PROGRAM
 TAD FIRST1 /CURRENT FIELD
 CIA /NO IF A 1
 TAD INSFLD /PRINT PROGRAM LOCATION
 SZA CLA /AND START OVER
 JMP ALOK /WAIT FOR C.R.
 JMS I XLCAT
 JMP SLMTS+2
 JMS KEYIN
 TAD CHAR
 CIA
 TAD K215 /NOT A C.R. IF A SKIP
 SNA /PRINT QUESTION MARK
 JMP I SLMTS /START OVER
 JMS QUERY
 JMP SLMTS+2
 Ø
 KCC
 KSF
 JMP ,--1
 KRB
 DCA CHAR
 TAD CHAR
 JMS I XPERR
 JMP I KEYIN

OKAS,

ALOK,

KEYIN,

1514 0000
 1515 1106
 1516 7041
 1517 1117
 1520 7650
 1521 5211
 1522 1106
 1523 0120
 1524 7041
 1525 1115
 1526 7650
 1527 5714
 1530 1106
 1531 7041
 1532 1353
 1533 7650
 1534 5714
 1535 1106
 1536 7041
 1537 1116
 1540 7650
 1541 5714
 1542 4344
 1543 5211

1544 0000
 1545 4522
 1546 1121
 1547 4510
 1550 5744

1551 1646
 1552 1600
 1553 0254

/ LEGAL,
 0 TAD CHAR
 CIA
 TAD K377 /RUB=OUT IF 0
 SNA CLA
 JMP SLMTS+2
 TAD CHAR
 AND K370
 CIA
 TAD K260
 SNA CLA
 JMP I LEGAL
 TAD CHAR
 CIA
 TAD K254 /A COMMA IF 0
 SNA CLA
 JMP I LEGAL
 TAD CHAR
 CIA
 TAD K215
 SNA CLA
 JMP I LEGAL /A C.R, IF 0
 JMS QUERY /QUERY
 JMP SLMTS+2 /START OVER

/ QUERY,
 0 JMS I XCRLF
 TAD K277
 JMS I XPERR /PRINT QUERY MARK
 JMP I QUERY

/ XTLIM:
 XLCAT: TLIMT
 K254, LOCAT
 254

/ TLIMIT, 0 /PRINT TEST LIMITS

CLA
TAD ISTL
DCA 12
TAD I 12
SNA
JMP I TLIMIT
JMS I XPERR
JMP PLIMIT

/DONE IF 0 /DONE IF 0

/ TSTL,

324
305
323
324
240
314
311
315
311
324
323
0

/T
/E
/S
/T
/
/L
/I
/M
/I
/T
/S
/TERMINATOR

/HEADER ROUTINE

/ PHDR,

JMS I XCRLF
TAD FILD
DCA 12
TAD I 12
SNA
JMP ,+3
JMS I XPERR
JMP PFILD
TAD M5
DCA LOOP
JMS I XPING
TAD OTLDR
DCA 12

/CR, LF

PFILD,

/PRINT FIELD /DONE IF 0

1646 0000
1647 7200
1650 1257
1651 3012
1652 1412
1653 7450
1654 5646
1655 4510
1656 5252
1657 1657
1660 0324
1661 0305
1662 0323
1663 0324
1664 0240
1665 0314
1666 0311
1667 0315
1670 0311
1671 0324
1672 0323
1673 0000

1674 0000
1675 4522
1676 1332
1677 3012
1700 1412
1701 7450
1702 5305
1703 4510
1704 5300
1705 1064
1706 3072
1707 4511
1710 1341
1711 3012

POCDR, TAD I 12 /DONE IF 0 /PRINT OCTAL ADR

SNA
JMP .+3
JMS I XPERR
JMP POCDR

1712 1412
1713 7450
1714 5317
1715 4510
1716 5312

TAD M5
DCA LOOP /SPACE 5
JMS I XPING
TAD GODD
DCA 12
TAD I 12 /PRINT GOOD
SNA /DONE IF 0

JMP I .+3
JMS I XPERR
JMP PGOOD
BSPCE /NEXT PAGE

1717 1064
1720 3072
1721 4511
1722 1355
1723 3012
1724 1412
1725 7450
1726 5731
1727 4510
1730 5324
1731 2000

FILED, 306 /F
311 /I
305 /E
314 /L
304 /D
0

OTLDR, 317 /O
303 /C
324 /T
301 /A
314 /L
240 /A
301 /D
304 /R
322 /,
256 /,
0

1732 1732
1733 0306
1734 0311
1735 0305
1736 0314
1737 0304
1740 0000

GODD, 1755 /G
1756 0307 /O
1757 0317 /O
1760 0317 /O
1761 0304 /D
1762 0000

EXHDR, JMP I PHDR

1741 1741
1742 0317
1743 0303
1744 0324
1745 0301
1746 0314
1747 0240
1750 0301
1751 0304
1752 0322
1753 0256
1754 0000


```

/
/
/ WAIT HERE TO SETUP SR. TYPE CARRIAGE RETURN
/ AFTER SETTING SR,
/
SETS,
0 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
JMP ,+4
KRB
JMS I XPERR
JMP I SETSR
LAS
DCA MCWA
JMP WTCR

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

/ STALL,
0 JMS I XFILD /GET ANOTHER
JMS GENRAN
AND K17
CMA
DCA FLCNT
TAD DATFLD
TAD KCDF
DCA ,+1
CDF 00 /RESTORE DATA FIELD
CLA /EXIT
JMP I STALL
0017

/ K17,
0017
2045 0000
2046 4503
2047 4522
2050 1270
2051 3012
2052 1412
2053 7450
2054 5257
2055 4510
2056 5252
2057 6036
2060 6031
2061 5265
2062 6036
2063 4510
2064 5645
2065 7604
2066 3037
2067 5260
2070 2070
2071 0323
2072 0305
2073 0324
2074 0325
2075 0320
2076 0240
2077 0323
2100 0322
2101 0000
2102 0000
2103 4503
2104 4317
2105 0316
2106 7040
2107 3071
2110 1020
2111 1077
2112 3313
2113 6201
2114 7200
2115 5702
2116 0017
0017

```

2117	0000	GENRAN, 0	TAD RANTAB
2120	1355		CIA
2121	7041		TAD RANDEX
2122	1343		SZA CLA
2123	7640		JMP RANTAD-1
2124	5334		TAD TBLRAN
2125	1356		DCA RANCON
2126	3343		TAD RANCON
2127	1342		CLL RAL
2130	7104		SZL
2131	7430		TAD K1
2132	1123		DCA RANCON
2133	3342		TAD I RANDEX
2134	1743		TAD RANCON
2135	1342	RANTAD,	DCA I RANDEX
2136	3743		TAD I RANDEX
2137	1743		ISZ RANDEX
2140	2343		JMP I GENRAN
2141	5717		
2142	1234	/	RANCON, 1234
2143	2154		RANDEX, RANTBL+10
2144	4321		RANTBL, 4321
2145	1416		1416
2146	5363		5363
2147	6060		6060
2150	3035		3035
2151	2572		2572
2152	3237		3237
2153	0214		0214
2154	0000		0
2155	2154		RANTAB, 1-1
2156	2144		TBLRAN, RANTBL

```

/ROUTINE TO DETERMINE FIELD FOR RELOCATION
/
*2200
/
CMOVE,   JMS I XFILD   /SET DF TO CURRENT FIELD
          7600
          TAD ERTBL
          DCA ERWRD   /SETUP ERROR TABLE POINTER
          TAD FIRST1  /FIRST TESTED FIELD
          CIA
          TAD LAST1   /LAST TESTED FIELD
          SNA CLA     /DON'T MOVE IF EQUAL
          JMP I XRTN   /START OVER
          TAD FLAGS
          RAR
          SZL         /FIRST MOVE IF A SKIP
          JMP I XTMV  /SETUP FOR NEXT MOVE
          IAC
          DCA FLAGS
          TAD LAST1
          DCA INSFLD  /NEW CURRENT FIELD
          TAD INSFLD
          TAD M10
          DCA NXLOC   /SUBTRACT 1 FROM NEW CURRENT
          RIF         /NXLOC=DOEST'N FOR NEXT TIME
          CIA
          TAD INSFLD
          SNA CLA
          JMP SUB1    /IS NEXT SAME AS CURRENT
                   /YES, TRY NEXT LOWER FIELD
2200      4503
2201      7600
2202      1024
2203      3026
2204      1075
2205      7041
2206      1076
2207      7650
2210      5502
2211      1021
2212      7010
2213      7430
2214      5725
2215      7001
2216      3021
2217      1076
2220      3023
2221      1023
2222      1114
2223      3124
2224      6224
2225      7041
2226      1023
2227      7650
2230      5266

```

```

2231 2026 /CHECK FOR ERROR IN NEW FIELD
2232 1201 /CKERR, ISZ ERWRD /POINTER+1
2233 7041 TAD CMOVE+1
2234 1426 CIA
2235 7650 TAD I ERWRD /NO ERRORS RECORDED IF 0
2236 5310 SNA CLA /INITIALIZE MOVE
      JMP STMV

2237 1426 CNXT, TAD I ERWRD
2240 7041 CIA
2241 1023 TAD INSFLD /ERROR IN NEW FIELD IF 0
2242 7650 SNA CLA
2243 5253 JMP EQUAL
2244 1026 TAD ERWRD
2245 7041 CIA
2246 1025 TAD ENTBL /ENTBL=ERWRD+10
2247 7650 SNA CLA /TABLE DONE IF 0
2250 5310 JMP STMV /INITIALIZE MOVE
2251 2026 ISZ ERWRD /POINTER+1
2252 5237 JMP CNXT

2253 1426 /EQUAL, TAD I ERWRD /GET ERROR FIELD
2254 7041 CIA
2255 1075 TAD FIRST1
2256 7650 SNA CLA /DON'T MOVE IF = TO FIRST
2257 5502 JMP I XRTN /START OVER
2260 1426 TAD I ERWRD
2261 7650 SNA CLA /IS IT FIELD 0?
2262 5266 JMP SUB1 /YES
2263 1023 TAD INSFLD /CURRENT NEXT
2264 1114 TAD M10 /SUBTRACT 1 FROM DF
2265 3124 DCA NXLOC

2266 1024 /SUB1, TAD ERTBL /RESTORE TABLE POINTER
2267 3026 DCA ERWRD
2270 1124 TAD NXLOC
2271 7041 CIA
2272 1023 TAD INSFLD
2273 7650 SNA CLA
2274 5253 JMP EQUAL
2275 1124 TAD NXLOC
2276 3023 DCA INSFLD
2277 1023 TAD INSFLD
2300 7041 CIA
2301 1075 TAD FIRST1
2302 7650 SNA CLA
2303 5231 JMP CKERR /YES
2304 1023 TAD INSFLD /CURRENT NEW FIELD
2305 1114 TAD M10 /SUBTRACT 1 FROM DF
2306 3124 DCA NXLOC /NEXT FIELD LOWER
2307 5231 JMP CKERR

```

2310	7200	CLA		
2311	1024	TAD	ERTBL	
2312	3026	DCA	ERWRD	/RESTORE TABLE POINTER
2313	6224	RIF		
2314	3723	DCA	I XSRCE	
2315	1723	TAD	I XSRCE	
2316	7041	CJA		
2317	1023	TAD	INSFLD	
2320	7650	SNA	CLA	/DON'T MOVE IF EQUAL
2321	5502	JMP	I XRTRN	/START OVER
2322	5724	JMP	I XMVE	/GO MOVE
2323	2522	SOURCE		
2324	2507	MOVE		
2325	2400	NXTMV		

2400	*2400		
2401	/		
2402	NXTMV,	/600	
2403	CHNXT,	RIF	/CURRENT FIELD
2404		DCA SOURCE	/POINTER +1
2405		ISZ ERWRD	
2406		TAD NXTMV	
2407		CIA	
2410		TAD I ERWRD	/NO ERRORS RECORDED IF 0
2411		SNA CLA	/INITIALIZE MOVE
2412		JMP STNXT	
2413		TAD I ERWRD	
2414		CIA	
2415		TAD NXLOC	/ERROR IN NEW FIELD IF 0
2416		SNA CLA	/TRY NEXT LOWER FIELD
2417		JMP SUB2	
2420		TAD ERWRD	
2421		CIA	
2422		TAD ENTBL	/DONE WITH TABLE IF 0
2423		SNA CLA	/INITIALIZE MOVE
2424		JMP STNXT	/POINTER +1
2425		ISZ ERWRD	
2426		JMP CKNXT	
2427		TAD ERTBL	
2428		DCA ERWRD	/RESTORE TABLE POINTER
2429		TAD NXLOC	/NEXT LOWER FIELD
2430		CIA	
2431		TAD INSFLD	
2432		SNA CLA	/NEXT=CURRENT IF 0
2433		JMP CKNT	
2434		TAD NXLOC	
2435		CIA	
2436		TAD FIRST1	/NEXT = LOWEST IF 0
2437		SEA CLA	/MOVE TO LOWEST TEST FIELD
2440		JMP STNX	
2441		JMP MVBK	
2442		TAD NXLOC	
2443		CIA	
2444		TAD FIRST1	/NEXT = LOWEST IF 0
2445		SNA CLA	/SETUP TO MOVE TO HIGHEST
2446		JMP NXTHI	/NEXT LOWER FIELD
2447		TAD NXLOC	/IS NOW CURRENT FIELD
2450		DCA INSFLD	
2451		TAD INSFLD	/SUBTRACT 1 FROM NEW
2452		TAD M10	/NEW NEXT LOWER FIELD
2453		DCA NXLOC	/GO MOVE
2454		JMP MOVE	
2455			
2456			
2457			
2458			
2459			
2460			
2461			
2462			
2463			
2464			
2465			
2466			
2467			
2468			
2469			
2470			
2471			
2472			
2473			
2474			
2475			
2476			
2477			
2478			
2479			
2480			
2481			
2482			
2483			
2484			
2485			
2486			
2487			
2488			
2489			
2490			
2491			
2492			
2493			
2494			
2495			
2496			
2497			
2498			
2499			

2455 1024
 2456 3026
 2457 1124
 2460 7450
 2461 5502
 2462 1114
 2463 3124
 2464 1124
 2465 7041
 2466 1023
 2467 7640
 2470 5203
 2471 1124
 2472 7450
 2473 5203
 2474 5262

/ SUB2,

TAD ERTBL
 DCA ERWRD
 TAD NXLOC
 SNA I XRTN
 TAD M10
 DCA NXLOC
 TAD NXLOC
 CIA
 TAD INSFLD
 SZA CLA
 JMP CHNXT
 TAD NXLOC
 SNA
 JMP CHNXT
 JMP SUB2*5

/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
 /YES
 /NO

/ NXTHI,

TAD LAST1
 DCA NXLOC
 TAD LAST1
 DCA INSFLD
 JMP CHNXT

/VERY LAST TO TEST
 /MAKE IT NEXT FIELD

/ MVBK,

TAD NXLOC
 DCA INSFLD
 RIF
 DCA SOURCE
 DCA FLAGS

/CLEAR BIT 11

2475 1076
 2476 3124
 2477 1076
 2500 3023
 2501 5203
 2502 1124
 2503 3023
 2504 6224
 2505 3322
 2506 3021

```

/ROUTINE TO RELOCATE 4K FIELDS
/
MOVE,
TAD KCDF
TAD SOURCE
DCA SOURCE
TAD KCDF
TAD INSFLD
DCA DESTN
CMA
DCA 10
CMA
DCA 11
DCA LOOP
SOURCE, 0
DESTN, 0
DCA I 11
ISZ LOOP
JMP SOURCE
TAD KCIF
TAD INSFLD
DCA I+1
CIF 00
JMP I XRTN
/6201
/CURRENT FIELD
/SOURCE NOW = CDF N
/NEW FIELD
/DESTN NOW = CDF N
/SOURCE COUNT
/DESTINATION COUNT
/4K COUNTER
/WILL = CDF N
/TAKE FROM HERE
/PUT IN HERE
/DONE 4K WHEN SKIP
/KEEP MOVING
/6202
/NEW FIELD
/CHANGE TO NEW FIELD
/EXIT TO RTN1 IN
/NEW FIELD

```

2507 1077
2510 1322
2511 3322
2512 1077
2513 1023
2514 3324
2515 7040
2516 3010
2517 7040
2520 3011
2521 3072
2522 0000
2523 1410
2524 0000
2525 3411
2526 2072
2527 5322
2530 1100
2531 1023
2532 3333
2533 6202
2534 5502

S

THERE ARE NO ERRORS

SYMBOL TABLE

ALAW	0220
ALOK	1473
BAD	1363
BADD	2034
BEGIN	0200
BSPCE	2000
CBANK	1066
CDF	6201
CFLD	1161
CHAR	0106
CHNXT	2403
CIF	6202
CKDN0	0642
CKDN1	0470
CKERR	2231
CKNT	2442
CKNXT	2411
CMOVE	2200
CNXT	2237
COMPR	0127
COUNT	0070
CRLF	1354
DATA	0143
DATFLD	0020
DESTN	2524
ENTBL	0025
EPRNT	1254
EQUAL	2253
EREXT	1246
ERROR	1200
ERTBL	0024
ERWRD	0026
EXAM1	0237
EXAM2	0244
EXAM3	0251
EXAM4	0256
EXHDR	1763
EXIT	0125
EXT3	0501
EXT4	0650
FIELD	1125
FILD	1732
FIRST1	0075
FLAGS	0021
FLCNT	0071
FLDN	1644
GENRAN	2117
GODD	1755
GOOD	1364
HALT	2041
INSFLD	0023
KCDF	0077
KCIF	0100

SYMBOL TABLE

KEYIN	1503
K1	0123
K10	0040
K100	0044
K17	2116
K20	0046
K200	0043
K207	1367
K212	1371
K215	0116
K240	1370
K254	1553
K260	0115
K261	0055
K262	0056
K263	0057
K264	0060
K277	0121
K370	0120
K377	0117
K40	0045
K400	0042
K7	0105
K740	0041
LAST	1366
LAST1	0076
LBK1	1007
LBK2	1036
LEGAL	1514
LOCAT	1600
LOOP	0072
MCMA	0037
MEMADR	0074
MOVE	2507
MVBK	2502
M1	0142
M10	0114
M13	0112
M14	0113
M20	0061
M4	0063
M40	0062
M5	0064
NXLOC	0124
NXTBnk	1103
NXTHI	2475
NXTMV	2400
OCADR	1365
OKAS	1456
ONES	1057
OTLDR	1741
OVER	1145
PATR	0126

SYMBOL TABLE

PBAD	2005
PFILD	1700
PGOOD	1724
PHDR	1674
PLIMIT	1652
POCT	1612
POCDR	1712
POSITN	1331
PRERR	1346
PRGAM	1617
PROCTL	1326
PSTSR	2052
PTSTN	2017
QUERY	1544
RANCON	2142
RANDEX	2143
RANTAB	2155
RANTAD	2135
RANTBL	2144
RBK1	1000
RBK2	1030
ROF	0214
RFR1	0720
RFR2	0736
RIF	0224
RROT	0441
RR1T	0604
RSL0	0600
RSL1	0436
RSTRT1	0211
RTN1	0213
SAVIF	1102
SETSR	2045
SETU1	1400
SHIFT	0141
SLMTS	1407
SOURCE	2522
SPING	1315
STALL	2102
STMV	2310
STNX	2447
STNXT	2425
STSK	2070
SUB1	2266
SUB2	2455
SW0	1311
SW1	1242
SW2	1233
TBLRAN	2156
TLIMIT	1646
TNUM	0065
TSTL	1657
TSTN	2026

SYMBOL TABLE

TST1	0276
TST1A	0313
TST2	0330
TST2A	0346
TST3	0400
TST4	0503
WFR1	0654
WFR2	0662
WRB1	0671
WRB2	0704
WRLOP	1147
WROT	0414
WR1T	0520
WTCR	2060
XBAK	0132
XBAK2	0136
XBANK	0066
XCRLF	0122
XFILD	0103
XFWD	0131
XFWD2	0135
XHDR	0107
XHLT	0101
XLGAT	1552
XLMTS	0022
XMOVE	0053
XMVE	2324
XONES	0140
XPERR	0110
XPHDR	2044
XPING	0111
XPRER	0104
XRROR	0073
XRTN	0102
XRT4	0652
XSALL	1065
XSETU	0054
XSL0	0552
XSRCE	2323
XSTSR	0275
XST4	0653
XIBNK	0067
XILIM	1551
XTMV	2325
XTST1	0047
XTST2	0050
XTST3	0051
XTST4	0052
XT4	0551
XWRB	0133
XWRB2	0137
XWRF	0130
XWRF2	0134

SYMBOL TABLE

SYMBOL TABLE

DATFLD	0020
FLAGS	0021
XLMTS	0022
INSFLO	0023
ERTBL	0024
ENTBL	0025
ERWRD	0026
MCWA	0037
K10	0040
K740	0041
K400	0042
K200	0043
K100	0044
K40	0045
K20	0046
XTST1	0047
XTST2	0050
XTST3	0051
XTST4	0052
XMOVE	0053
XSETU	0054
K261	0055
K262	0056
K263	0057
K264	0060
M20	0061
M40	0062
M4	0063
M5	0064
TNUM	0065
XBANK	0066
XTBANK	0067
COUNT	0070
FLCNT	0071
LOOP	0072
XRROR	0073
MEMADR	0074
FIRST1	0075
LAST1	0076
KCDF	0077
KCIF	0100
XHLT	0101
XRYN	0102
XFILD	0103
XPRER	0104
K7	0105
CHAR	0106
XHDR	0107
XPERR	0110
XPING	0111
M13	0112
M14	0113
M10	0114

SYMBOL TABLE

K260	0115
K215	0116
K377	0117
K370	0120
K277	0121
XCRLF	0122
K1	0123
NXLOC	0124
EXIT	0125
PATR	0126
COMPR	0127
XWRF	0130
XFWD	0131
XBAK	0132
XWRB	0133
XWRF2	0134
XFWD2	0135
XBAK2	0136
XWRB2	0137
XONES	0140
SHIFT	0141
M1	0142
DATA	0143
BEGIN	0200
RSTRT1	0211
RTN1	0213
ALAW	0220
EXAM1	0237
EXAM2	0244
EXAM3	0251
EXAM4	0256
XSTSR	0275
TST1	0276
TST1A	0313
TST2	0330
TST2A	0346
TST3	0400
WROT	0414
RSL1	0436
RROT	0441
CKDN1	0470
EXT3	0501
TST4	0503
WRIT	0520
XT4	0551
XSL0	0552
RSL0	0600
RRIT	0604
CKDN0	0642
EXT4	0650
XRT4	0652
XST4	0653
WFR1	0654

SYMBOL TABLE

WFR2	0662
WRB1	0671
WRB2	0704
RFR1	0720
RFR2	0736
RBK1	1000
LBK1	1007
RBK2	1030
LBK2	1036
ONES	1057
XSALL	1065
CBANK	1066
SAVIF	1102
NXTBANK	1103
FIELD	1125
OVER	1145
WRLOP	1147
CFLD	1161
ERROR	1200
SW2	1233
SW1	1242
EREXT	1246
EPRNT	1254
SW0	1311
SPING	1315
PROCTL	1326
POSITN	1331
PRERR	1346
CRLF	1354
BAD	1363
GOOD	1364
OCADR	1365
LAST	1366
K207	1367
K240	1370
K212	1371
SETU1	1400
SLMTS	1407
OKAS	1456
ALOK	1473
KEYIN	1503
LEGAL	1514
QUERY	1544
XTLIM	1551
XLCAT	1552
K254	1553
LOCAT	1600
PLDCT	1612
PRGAM	1617
FLON	1644
TLIMT	1646
PLIMT	1652
TSTL	1657

SYMBOL TABLE

PHDR	1674
PFILD	1700
POCDR	1712
PGOOD	1724
FILD	1732
OFILDR	1741
GDD	1755
EXHDR	1763
BSPCE	2000
PBAD	2005
PTSTN	2017
TSTN	2026
BADD	2034
HALT	2041
XPHDR	2044
SETSR	2045
PSTSR	2052
WTCR	2060
STSR	2070
STALL	2102
17	2116
GENRAN	2117
RANTAD	2135
RANCON	2142
RANDEX	2143
RANTBL	2144
RANTAB	2155
TBLRAN	2156
CMOVE	2200
CKERR	2231
CNXT	2237
EQUAL	2253
SUB1	2266
STMV	2310
XSRCE	2323
XMVE	2324
XTMV	2325
NXTMV	2400
CHNXT	2403
CKNXT	2411
STNXT	2425
CKNT	2442
STNX	2447
SUB2	2455
NXTHI	2475
MVBK	2502
MOVE	2507
SOURCE	2522
DESTN	2524
COF	6201
CIF	6202
RDF	6214
RIF	6224

SYMBOL TABLE