

done
at

IDENTIFICATION

Product Code: MAINDEC-08-D1HA-D

Product Name: PDP-8, 8/I, Extended Memory
Address Test

Date Created: March 13, 1968

Maintainer: Diagnostic Group

Author: J. W. Richardson

COPYRIGHT © 1968
DIGITAL EQUIPMENT CORPORATION

600

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/I 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 typeout 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

4/1/68 16:50,4

PAGE 1

/PNP-B, 81, 8S EXTENDED MEMORY ADDRESS TEST.
/START AT 200, RESTART AT 207 TO SKIP HEADER.
/MIN. OF 8K OF CORE REQUIRED.
/

*1
/
0001 5001 JMP,
0002 0002 0002
0003 0003 0003 /
0010 0010 /*010
/
6201 CDF=6201
6202 CIF=6202
6214 RDF=6214
6224 RIF=6224 /
0010 0000 0 0
0011 0000 0 0
0012 0000 0 0
0013 0000 0 0
0014 0000 0 0
0015 0000 0 0
0016 0000 0 0
0017 0000 0 0
0020 0000 DATAFLD, 0
0021 0000 FLAGS, 0
0022 1407 XLMTS, SLMTS
0023 0000 INFSLD, 0
0024 0026 ERtbl, ERWRD
0025 0036 ENTBL, ERWRD+10
0026 0026 ERWRD, 1
0027 7600 7600
0030 7600 7600
0031 7600 7600
0032 7600 7600
0033 7600 7600
0034 7600 7600
0035 7600 7600
0036 7600 7600
0037 0000 MCNA, 0
0040 0010 K10, 10
0041 0740 K740, 740
0042 0400 K400, 400
0043 0200 K200, 200
0044 0100 K100, 100
0045 0040 K40, 40
0046 0020 K20, 20
0047 0276 XTST1, TST1
0050 0330 XTST2, TST2
0051 0400 XTST3, TST3
0052 0503 XTST4, TST4
0053 2200 XMOVE,

PAGE 1-1

4/1/68 16:50,4

	XSETU,	SETU1
0054	1400	
0055	0261	261
0056	0262	262
0057	0263	263
0060	0264	264
0061	7760	M20, 7760

0062	7740	M4A,	/740
0063	7774	M4,	/774
0064	7773	M5,	773
0065	0000	TNUM,	0
0066	1066	XBANK,	C BANK
0067	1103	XTBNK,	NX TBNK
0070	0000	COUNT,	0
0071	0000	FLCNT,	0
0072	0000	LOOP,	0
0073	1200	XERROR,	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,	SRING

M113,	7765	M14,	/7764		
01113	7764	M16,	7770		
01114	7770	K260,	260		
01115	0260	K215,	215		
01116	0215	K377,	377		
01117	0377	K370,	370		
01120	0370	K277,	277		
01121	0277	XCRLF,	CRLF		
01122	1354	K1,	1		
01123	0001	NXLOC,			
01124	0000	EXIT,			
01125	0000	PATR,			
01126	0000	COMPR,			
01127	0000	WFR1,			
01130	0654	XWRF,			
01131	0720	XFWD,			
01132	1000	XBAK,			
01133	0671	XWRB,			
01134	0662	XWRF2,			
01135	0736	XFWD2,			
01136	1030	XBAK2,			
01137	0704	XWRB2,			
01140	1005	XONES,			
01141	0000	SHIFT,			
01142	7777	M1,	7777		
	0000	DATA,			

*200
0200 6002 *BEGIN, IOF /PI OFF
0201 7200 CLA
0202 6224 RIF
0203 3023 DCA INSFLD
0204 3021 DCA FLAGS /CLEAR PROGRAM FLAGS
0205 4422 JMS I XLMTS /SETUP TEST LIMITS
0206 4675 JMS I XSTS /SETUP SR
0207 4507 JMS I XHDR /PRINT HEADER
0210 5213 JMP RTN1

/RESTART HERE
/RSTRT1, JMS I XLMTS /SET TEST LIMITS
RTN1, JMS I XSTS /SETUP SR
RTN1, RIF /READ INSTRUCTION FIELD
RTN1, DCA INSFLD /CURRENT FIELD
RTN1, JMS I XFILD
RTN1, TAD M10 /-10
RTN1, DCA LOOP
RTN1, ALAW, 7600
RTN1, TAD ALAW
RTN1, ISZ ERWRD
RTN1, DCA I ERWRD
RTN1, ISZ LOOP
RTN1, JMP ALAW
RTN1, TAD ERTBL
RTN1, DCA ERWRD

/

```

1037          TAD MCWA      /EXAMINE SR
0231 0041          AND K740      TAD MCWA      /MAJIK 3,4,5 AND 6
0232 7440          SZA         AND K740      /JO ALL IF 4
0233 5237          JMP EXAM1   SZA
0234 1037          TAD MCWA   JMP EXAM1
0235 1041          TAD K740   TAD MCWA
0236 3037          DCA MCWA   /SET ALL TEST BITS
0237 7200          CLA         DCA MCWA
0240 1037          TAD MCWA   /SAVE
0241 0042          AND K400   SZA
0242 7440          SZA         AND K400      /TEST 1 IF NO SKIP
0243 5447          EXAM1,    JMP I XTST1
0244 7200          CLA         TAD MCWA
0245 1037          AND K200   SZA
0246 0043          EXAM2,    /TEST 2 IF NO SKIP
0247 7440          TAD MCWA   CLA
0250 5450          AND K100   SZA
0251 7200          SZA         AND K100      /TEST 3 IF NO SKIP
0252 1037          EXAM3,    JMP I XTST2
0253 0044          CLA         TAD MCWA
0254 7440          AND K40   SZA
0255 5451          EXAM4,    CLA
0256 7200          TAD MCWA   AND K40      /TEST 4 IF NO SKIP
0257 1037          AND K40   SZA
0260 0045          SZA         JMP I XTST4
0261 7440          EXAM4,    JMS I XFIELD
0262 5452          CLA         /RESTORE DATA FIELD
0263 4503          LAS         LAS
0264 7604          AND K1   JMP RSTRY1
0265 0123          SZA         /CHECK SR 11
0266 7440          AND K20   LAS
0267 5211          SZA         /INHIBIT MOVE IF A 1
0270 7604          AND K20   JMP RTN1
0271 0046          SZA         /GO RELOCATE
0272 7440          EXAM4,    JMP I XMOVE
0273 5213          CLA
0274 5453          SETSR,   /XSTS,  SETSR
0275 2045          SZA

```

```

    / TEST 1, WRITE THE ADDRESS OF EACH LOCATION INTO
    / ITSELF, THEN READ EACH STACK FORWARDS AND BACKWARDS,
    /
TST1,   JMS I XSETU      /SET DF TO 1ST FIELD
        TAD K261      /TEST NUMBER
        DCA TNUM
        DCA MEMADR
        DCA PATR
        DCA PATR
        JMS I XBANK    /SEE IF FIELD HAS PROGRAM
        SKP           /NO, BEGIN WRITING
        JMP TST1A     /NEXT TEST
        JMS I XWRF      /WRITE FORWARDS
        JMS I XFWD      /READ FORWARDS
        JMS I XBAK      /READ BACKWARDS
        JMS I XTBNK     /SETUP FOR NEXT FIELD
        JMP TST1+3

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

```

```

        / TEST 2: WRITE COMPLEMENT ADDRESS
        / PATTERN FORWARD AND BACKWARD,
        /
        TST2,      JMS I XSETU      /SET OF TO 1ST FIELD
        03330      0454           /TEST NUMBER
        03331      1056           /SET ADDRESS COUNT TO 0
        03332      3065           /SET ADDRESS COUNT TO 0
        03333      3074           /SET ADDRESS COUNT TO 0
        03334      7040           /COMPARE WORD
        03335      3126           /SEE IF FIELD HAS PROGRAM
        03336      4466           /WRITE
        03337      7410           /NEXT TEST
        03340      5346           /SEE IF FIELD HAS PROGRAM
        03341      4534           /READ FORWARDS
        03342      4535           /READ BACKWARDS
        03343      4536           /SETUP FOR NEXT FIELD
        03344      4467           /SETUP FOR NEXT FIELD
        03345      5333           /NEXT TEST

        / NOW WRITE BACKWARDS
        /
        TST2A,     JMS I XSETU      /SET OF TO 1ST FIELD
        CLA CMA
        DCA MEMADR
        CMA
        DCA PATR
        JMS I XBANK
        SKP
        JMP TST2A
        JMS I XWRF2
        JMS I XFWD2
        JMS I XBAK2
        JMS I XTBNK
        JMP TST2A+3
        JMS I XWRB2
        JMS I XFWD2
        JMS I XBAK2
        JMS I XTBNK
        JMP EXAM3
        JMS I XWRB2
        JMS I XFWD2
        JMS I XBAK2
        JMS I XTBNK
        JMP TST2A+1
        JMP EXAM3
        /NEXT TEST

```

```

        / TEST3. WRITE ALL 1'S, THEN SLIDE A SINGLE 1 THRU EACH
        / WORD, READ AND WRITE IN THE FORWARD DIRECTION ONLY.
        /
        *400
        0400    4454      JMS 1 XSETU      /SET DF TO 1ST FIELD
        0401    1057      TAD K263
        0402    3065      DCA TNUM
        0403    7001      IAC PATR      /SET BIT 11
        0404    3126      DCA COMPR      /SAVE FIRST BIT
        0405    7001      DCA MEMADR   /SET ADDRESS COUNT TO 0
        0406    3127      JMS 1 XBANK   /SEE IF FIELD HAS PROGRAM
        0407    3074      SKP          /WRITE
        0410    4466      JMP EXT3      /WRITE 1'S
        0411    7410      JMS 1 XONES   /WRITE 1'S
        0412    5301      TAD M14      /-12 DECIMAL
        0413    4540      DCA SHIFT
        0414    1113      WROT,
        0415    3141      TAD PATR
        0416    1126      DCA 1 MEMADR /DONE 4K WHEN SKIP
        0417    3474      ISZ2 MEMADR
        0420    2074      SKP          /READ AND COMPARE
        0421    7410      JMP RSL1      /CHECK FOR 12 POSITIONS
        0422    5236      ISZ2 SHIFT
        0423    2141      JMP +4      /POSITION NEXT BIT
        0424    5230      TAD COMPR   /START OVER WITH 11
        0425    1127      DCA PATR
        0426    3426      JMP WROT
        0427    5214      TAD PATR
        0428    1126      CLL RAL
        0429    1126      SZL
        0430    1126      IAC PATR
        0431    7104      DCA PATR
        0432    7430      JMP WROT*2
        0433    7001
        0434    3126
        0435    5216

```

```

/ READ AND TEST ROUTINE FOR SLIDING ONE
/ RSL1, TAD COMPR          /FIRST POSITION
  0436 1127
  0437 3126      DCA PATR
  0440 3074      DCA MEMADR /SET ADDRESS COUNT TO 0
  0441 1113      TAD M14   /-12 DECIMAL
  0442 3141      DCA SHIFT
  0443 1474      TAD I MEMADR /SAVE
  0444 3143      DCA DATA
  0445 1143      TAD DATA
  0446 7041      CIA
  0447 1126      TAD PATR
  0450 7640      SZA CLA   /2'S COMPLEMENT
  0451 4473      JMS I XRROR /OK IF 0
  0452 2074      ISE MEMADR /PRINT ERROR
  0453 7410      SKP      /DONE IF SKIP
  0454 5270      JMP CKDN1 /DONE
  0455 2141      ISZ SHIFT /CHECK FOR 12 POSITIONS
  0456 5262      JMP I+4   /POSITION NEXT BIT
  0457 1127      TAD COMPR /START OVER WITH 11
  0460 3126      DCA PATH
  0461 5241      JMP RR0T
  0462 1126      TAD PATR
  0463 7104      CLL RAL
  0464 7430      S2L
  0465 7001      IAC
  0466 3126      DCA PATR
  0467 5243      JMP RR0T*2

/ CKDN1, TAD COMPR          /DONE ALL 12 IF NO SKIP
  0470 1127
  0471 7104      CLL RAL
  0472 7430      S2L
  0473 5277      JMP I+4
  0474 3126      DCA PATR
  0475 1126      TAD PATR
  0476 5206      JMP WROT=6 /WRITE IN SAME FIELD AGAIN
  /
  0477 4467      JMS I XTBANK /SETUP FOR NEXT FIELD
  0500 5203      JMP TST3*3
  0501 5702      JMP I+1   /NEXT TEST
  0502 0256      EXAM4

```

```

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

        TST4,   JMS I XSETU      /SET OF TO 1ST FIELD
                  TAD K264      /TEST NUMBER
                  DCA TNUM
                  IAC
                  DCA COMPR
                  TAD COMPR
                  CMA
                  DCA PATR
                  DCA MEMADR     /SET ADDRESS COUNT TO 0
                  JMS I XBANK     /SEE IF FIELD HAS PROGRAM
                  SKP
                  JMP I XT4
                  JMS I XONES      /WRITE 1'S
                  STL
                  TAD M14         /-12 DECIMAL
                  DCA SHIFT
                  TAD PATR
                  DCA I MEMADR
                  ISZ MEMADR     /DONE 4K WHEN SKIP
                  SKP
                  JMP I XSL0
                  ISZ SHIFT
                  JMP I+5
                  TAD COMPR
                  CMA
                  DCA PATR
                  JMS WRIT
                  TAD PATR
                  RAL
                  SNL
                  JMP I+3
                  DCA PATR
                  JMP WRIT+3
                  CLA
                  IAC
                  CMA
                  STL
                  JMP , -6
                  /XT4,
                  XSL0,
                  EXAM4+5
                  RSL0

```

0503 4454
 0504 1060
 0505 3065
 0506 7001
 0507 3127
 0510 1127
 0511 7040
 0512 3126
 0513 3074
 0514 4466
 0515 7410
 0516 5751
 0517 4540
 0520 7120
 0521 1113
 0522 3141
 0523 1126
 0524 3474
 0525 2074
 0526 7410
 0527 5752
 0530 2141
 0531 5336
 0532 1127
 0533 7040
 0534 3126
 0535 5329
 0536 1126
 0537 7004
 0540 7420
 0541 5344
 0542 3126
 0543 5323
 0544 7200
 0545 7001
 0546 7040
 0547 7120
 0550 5342
 0551 0263
 0552 0600

```

*600
/READ AND TEST ROUTINE FOR SLIDING ZERO
/RSLD, TAD COMPR          /1ST POSITION
  CMA PATR
  DCA MEMADR             /SET ADR, COUNT TO 0
  DCA TAD M14             /=12 DECIMAL
  DCA SHIFT
  TAD I MEMADR
  TAD DATA               /READ
  DCA DATA               /SAVE
  TAD DATA
  CMA PATR
  TAD PATR
  CMA
  SZA CLA
  JMS I XRROR
  ISE MEMADR
  SKP
  JMP CKDN0
  ISE SHIFT
  JMP '+5
  TAD COMPR
  CMA
  DCA PATR
  JMP RR1T
  TAD PATR
  STL RAL
  SNL
  JMP '+3
  DCA PATR
  JMP RR1T+2
  CLA
  IAC
  CMA
  JMP '+5
  /CKDN0, TAD COMPR
  CLL RAL
  SNL
  JMP I XRT4
  /
  JMS I XTBNK
  JMP I XST4
  JMP I '+1
  EXAM4+5
  /SETUP FOR NEXT FIELD
  EXT4,
  /DONE ALL TESTS
  /
  XRT4, TST4+4
  XST4, TST4+3
  0642 1127
  0643 7104
  0644 7420
  0645 5652
  0646 4467
  0647 5653
  0650 5651
  0651 0263
  0652 0507
  0653 0506

```

```

/
/ WRITE FORWARD ROUTINE FOR TST1
/
WFR1,    0000          / WRITE C(MEMADR) INTO
                      TAD MEMADR / SAME ADDRESS
                      DCA I MEMADR / DONE WHEN SKIP
                      ISZ MEMADR
                      JMP WFR1+1
                      JMP I WFR1 / EXIT

/
/ WRITE FORWARD ROUTINE FOR TST2
/
WFR2,    0000          / WRITE COMPLEMENT OF
                      TAD MEMADR / C(MEMADR) INTO SAME
                      CMA
                      DCA I MEMADR / DONE 4K WHEN SKIP
                      ISZ MEMADR
                      JMP WFR2+1
                      JMP I WFR2 / EXIT

/
/ WRITE BACKWARD ROUTINE FOR TST1A
/
WRB1,    0000          / 4K COUNTER
                      DCA LOOP / INITIALLY=7777
                      TAD MEMADR
                      DCA I MEMADR
                      ISZ LOOP
                      SKP
                      JMP I WRB1 / EXIT
                      TAD MEMADR
                      TAD M1
                      DCA MEMADR
                      JMP WRB1+2 / SUBTRACT 1 FROM ADDRESS

/
/ WRITE BACKWARDS ROUTINE FOR TST2A
/
WRB2,    0000          / 4K COUNTER
                      DCA LOOP / INITIALLY=7777
                      TAD MEMADR
                      CMA
                      DCA I MEMADR
                      ISZ LOOP
                      SKP
                      JMP I WRB2 / EXIT
                      TAD MEMADR
                      TAD M1
                      DCA MEMADR
                      JMP WRB2+2 / SUBTRACT 1 FROM ADDRESS

0654 0000
0655 1074
0656 3474
0657 2074
0660 5255
0661 5664

0662 0000
0663 1074
0664 7040
0665 3474
0666 2074
0667 5263
0670 5662

0671 0000
0672 3072
0673 1074
0674 3474
0675 2072
0676 7410
0677 5671
0700 1074
0701 1142
0702 3074
0703 5273

0704 0000
0705 3072
0706 1074
0707 7040
0710 3474
0711 2072
0712 7410
0713 5704
0714 1074
0715 1142
0716 3074
0717 5306

```

/READ FORWARD ROUTINE FOR TST1 AND TST1A

/RFR1, 0 TAD I MEMADR

DCA DATA

TAD DATA

CIA

TAD PATR

SZA CLA

/COMPARE WORD

/MUST EQUAL 0

JMS I XRROR

/PRINT ERROR

/DONE 4K WHEN SKIP

SKP

JMP I RFR1

/EXIT

ISZ PATR

JMP RFR1*1

JMP I RFR1

/SHOULD NEVER GET HERE

/READ FORWARD ROUTINE FOR TST2 AND TST2A

/RFR2, 0 TAD I MEMADR

DCA DATA

TAD DATA

CIA

TAD PATR

SZA CLA

/COMPARE WORD

/MUST=0

JMS I XRROR

/PRINT ERROR

/DONE 4K WHEN SKIP

SKP

JMP I RFR2

/EXIT

TAD PATR

CIA

/SUBTRACT 1

CHA

DCA PATR

JMP RFR2*1

0000

0720 0721 1474

0722 3143

0723 1143

0724 7041

0725 1126

0726 7640

0727 4473

0730 2074

0731 7410

0732 5720

0733 2126

0734 5321

0735 5720

0000

0736 0737 1474

0740 3143

0741 1143

0742 7041

0743 1126

0744 7640

0745 4473

0746 2074

0747 7410

0750 5736

0751 1126

0752 7041

0753 7040

0754 3126

0755 5337

```

1000 *1000
      / READ BACKWARD ROUTINE FOR TST1 AND TST1A
      /
      RBK1,   0          DCA LOOP           /4K COUNTER
      0000    3072          CMA
      1001    7040          DCA MEMADR     /SET ADR. COUNT TO 7777
      1002    3074          CMA
      1003    7040          DCA PATR       /COMPARE WORD
      1004    3126          JMS I X$ALL
      1005    4665          TAD I MEMADR
      1006    1474          /READ           /SAVE
      1007    3143          DCA DATA
      1010    1143          TAD DATA
      1011    1143          CIA
      1012    7041          TAD PATR
      1013    1126          SZA CLA        /MUST=0
      1014    7640          JMS I X$0000
      1015    4473          /PRINT ERROR
      1016    2071          JMS I X$0000
      1017    5207          ISZ FLCNT
      1018    1143          JMP LBK1
      1020    2072          ISZ LOOP
      1021    7410          SKP
      1022    5600          JMP I RBK1
                           EXIT

      TAD MEMADR
      TAD M1
      DCA MEMADR
      TAD MEMADR
      JMP LBK1=2

1023    1074
1024    1142
1025    3074
1026    1074
1027    5205

```

/READ BACKWARD ROUTINE FOR TST2 AND TST2A

```

1030 0000 0000
1031 3072 0000
1032 7040 CMA LOOP /4K COUNTER
1033 3074 DCA MEMADR /SET ADR, COUNT TO 7777
1034 3126 DCA PATR /COMPARE WORD
1035 4665 JMS I XSTALL
1036 1474 TAD I MEMADR /READ
1037 3143 DCA DATA /SAVE
1040 1143 TAD DATA
1041 7041 CIA
1042 1126 TAD PATR
1043 7640 SZA CLA /MUST=0
1044 4473 JMS I XRORR /PRINT ERROR
1045 2071 ISZ FLCNT /RANDOM LOOP
1046 5236 JMP LBK2 /DONE 4K WHEN SKIP
1047 2072 ISZ LOOP
1050 7410 SKP /EXIT /COMPARE WORD
1051 5630 JMP I RBK2
1052 2126 ISZ PATR
1053 1074 TAD MEMADR
1054 1142 TAD M1
1055 3074 DCA MEMADR
1056 5235 JMP LBK2*1 /LOOP

```

/ROUTINE TO WRITE ONES IN ONE FIELD

```

1057 0000 0000
1060 7240 CLA CMA
1061 3474 DCA I MEMADR
1062 2074 ISZ MEMADR /DONE 4K WHEN SKIP
1063 5260 JMP ONES*1
1064 5657 JMP I ONES /EXIT
1065 2102 XSTALL, STALL

```

4/1/68 16:50,17 PAGE 16

PAUSE

4/1/68 16:50,17 PAGE 17

/EXXTENDED MEMORY ADDRESS TEST - TAPE 2
ROUTINE TO SEE IF TESTED FIELD HAS PROGRAM
/
CBANK, \emptyset RIF /READ INST, FIELD
0000 DCA SAVIF /SAVE
1066 6224
1067 3302
1070 6214
1071 7041
1072 1302
1073 1074 7640
1075 5666
1076 4467
1077 5666
1100 2266
1101 5666
1102 0000
/CBANK, \emptyset RIF /READ INST, FIELD
DCA SAVIF /SAVE
RDF /READ DATA FIELD
CIA
TAD SAVIF
SZA CLA /EQUAL IF AC=0
JMP I CBANK /DOESN'T HAVE PROGRAM
JMS I XTBANK /INCREMENT DATA FIELD
JMP I CBANK /TEST NEW FIELD
ISZ CBANK /DONE ALL CAUSE PROGRAM NOW
/IN HIGHEST FIELD
JMP I CBANK /EXIT
SAVIF, \emptyset

```

/ROUTINE TO SET DF FOR NEXT FIELD
/
NXTBNK, 0
    CLA          /READ DATA FIELD
    RDF
    CIA          /C(LAST1) = LAST1 TO TEST
    TAD LAST1
    SZA CLA      /ALL DONE IF 0
    JMP ,+3
    ISZ NXTBNK
    JMP ,+6
    RDF          /EXIT
    TAD K10      /INCREMENT DATA FIELD
    TAD KCDF
    ADD ,6201
    DCA ,+1
    CDF 00      /CHANGE TO NEW DATA FIELD

/CHECK SWITCH REGISTER
LAS
SPA CLA          /CHEC HALT
JMS ,XHLT        /GO HALT, SR0=1
JMP ,NXTBNK     /EXIT

/RESTORE DATA FIELD AND CHECK SR
/
FIELD, 0
    CLA
    RDF
    DCA ,DATFLD   /SAVE TESTED FIELD#
    RIF
    TAD KCDF
    DCA ,+1
    CDF 00      /MAKE DATA AND INST FIELD EQUAL
    CLA
    JMP ,FIELD

```

1103	00000	
1104	72000	
1105	6214	
1106	7041	
1107	1076	
1110	7640	
1111	5314	
1112	2303	
1113	5321	
1114	6214	
1115	1040	
1116	1077	
1117	3320	
1120	6201	
1121	7604	
1122	7710	
1123	4501	
1124	5703	
1125	00000	
1126	72000	
1127	6214	
1130	30200	
1131	6224	
1132	1077	
1133	3334	
1134	6201	
1135	7200	
1136	5725	

```

/
/ START HERE TO LOOP ON ADDRESS
/
    1137    7200      CLA      LAS      /READ LOWER LIMIT
    1140    7604      DCA      FIRST1   /NOW SETUP UPPER LIMIT
    1141    3075      HLT
    1142    7402      LAS
    1143    7604      DCA      LAST1
    1144    3076      TAD      FIRST1
    1145    1075      DCA      MEMADR
    1146    3074      TAD      I MEMADR
    1147    1474      DCA      I MEMADR
    1150    3474      TAD      MEMADR
    1151    1074      CIA
    1152    7041      TAD      LAST1
    1153    1076      SNA      CLA
    1154    7650      JMP      OVER
    1155    5345      ISZ      MEMADR
    1156    2074      JMP      WRLOP
    1157    5347      HLT
    1160    7402      CLA      DATF0D
    1161    0000      CLA      DATF0D
    1162    7200      TAD      KCDF
    1163    1020      DCA      *1
    1164    1077      CDF      00
    1165    3366      CLA      A
    1166    6201      JMP      1 CFID
    1167    7200      /RESTORE TEST FIELD
    1170    6201      /EXIT

```

```

/PRINT ERROR ROUTINE
/
*1200, V
1200 0000
1201 1143 TAD DATA /SAVE BAD DATA
1202 3363 DCA BAD
1203 1126 TAD PATR
1204 3364 DCA GOOD /SAVE GOOD DATA
1205 1074 TAD MEMADR
1206 3365 DCA OCADR /OCTAL ADDRESS
1207 7040
1210 3071 CMA FLCNT JNS 1 XFILD /RESTORE DATA FIELD
1211 4503 TAD DATFLD /DATA FIELD
1212 1020 CIA LAST /LAST = FIELD WITH LAST ERROR
1213 7041 TAD SNA CLA /SAME IF 0
1214 4366 7650 JMP SW2 /DON'T STORE
1215 5233 TAD DATFLD
1216 1020 DCA LAST /TABLE POINTER
1217 3366 TAD ERWRD CIA
1220 1026 TAD ENLBL SZA CLA /END OF TABLE IF = 0
1221 1021 TAD ERWRD JMP +3
1222 7041 TAD ERTBL DCA ERWRD
1223 1025 TAD DATFLD
1224 7640 TAD ERWRD ISEZ ERWRD /INCREMENT POINTER
1225 5230 TAD DATFLD DCA I ERWRD /STORE IN TABLE
1226 1024
1227 3026
1230 1020
1231 2026
1232 3426
1233 7604
1234 7006
1235 7700
1236 5242
1237 1367
1240 4346
1241 5311
1242 7604
1243 7004
1244 7700
1245 5254
1246 1020
1247 1077
1248 3251
1251 6201
1252 7200
1253 5600

/ SW2, LAS RAL /SR2 ON A 1 = RING BELL
RTL SMA CLA
JMP SW1
TAD K207
JMS PRRR
JMP SW0
LAS
SMA CLA /SR1 A 1 = NO PRINT
JMP EPRNT
TAD DATFLD
TAD KCDF
DCA *+1
CDF 00 /SET TO TESTED FIELD
CLA
JMP 1 ERROR

```

```

/
EPRINT, JMS CRLF /CR,LF
          TAD DATFLD /TEST NUMBER
          RAK
          TAD K260 /PRINT
          JMS PRERR /-12 DECIMAL
          TAD M14
          DCA LOOP /SPACE 12
          JMS SPING /OCTAL ADR,
          TAD OCADR /SAVE
          DCA CHAR /PRINT
          JMS PROCTL /-8 DECIMAL
          TAD M10
          DCA LOOP /SPACE 8
          JMS SPING /GOOD
          TAD GOOD
          DCA CHAR /PRINT
          JMS PROCTL /-5
          TAD M5
          DCA LOOP /SPACE 5
          JMS SPING /BAD
          TAD BAD
          DCA CHAR /PRINT
          JMS PROCTL /TEST NUMBER
          TAD M5
          DCA LOOP /PRINT
          JMS SPING /PRINT
          TAD TNUM /PRINT
          JMS PRERR /PRINT
          /
SW0, LAS /CH CK SR0
          SPA CLA /GO HALT
          JMS IXHLT /EIT
          JMP EREXT /PRINT SPACES
          /
SPING, Ø TAD K240 /PRINT
          TLS
          TSF
          JMP :-1
          ISZ LOOP
          JMP SPING+2
          CLA /EXIT
          JMP I SPING

```

```

/
/PRINT OCTAL
/PROC TL, 0          TAD M4          /--4
                                         DCA LOOP
                                         POSITN, TAD CHAR
                                         CLL RAL
                                         RTL
                                         DCA CHAR
                                         TAD CHAR
                                         RAL
                                         AND K7
                                         TAD K260
                                         JMS PRERR
                                         ISZ LOOP
                                         JMP POSITN
                                         CLA
                                         JMP ! PROC TL  /EXIT
                                         /PRINT A NUMBER
                                         PRERR, 0
                                         TLS
                                         TSF
                                         JMP ,=1
                                         CLA
                                         JMP ! PRERR  /EXIT
                                         /CARRIAGE RETURN, LINE FEED
                                         CRLF, 0
                                         CLA
                                         TAD K212
                                         JMS PRERR
                                         TAD K212
                                         JMS PRERR
                                         JMP ! CRLF
                                         /BAD, 0
                                         GOOD, 0
                                         OCADR, 0
                                         LAST, 0
                                         K207, 207
                                         K240, 240
                                         K212, 212

```

1326 0000
 1327 1063
 1330 3072
 1331 1106
 1332 7104
 1333 7006
 1334 3106
 1335 1106
 1336 7004
 1337 0105
 1340 1115
 1341 4346
 1342 2072
 1343 5331
 1344 7200
 1345 5726

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

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

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

```

1400          *1400
               /ROUTINE TO SET OF 10 FIRST TEST FIELD
               /SETU1,   CLA      /FIRST TO TEST
               TAD FIRST1
               TAD KCDF
               DCA ,+1
               CDF 00      /CHANGE TO TEST FIELD
               JMP 1 SETU1  /EXIT
               /ROUTINE TO ACCEPT TEST LIMITS FROM
               /KEYBOARD INPUT
               /SLMTS,   0
               JMS 1 XFIELD
               JMS 1 XCRLF
               JMS 1 XTLIM
               JMS 1 XCRLF
               JMS KEYIN
               JMS LEGAL
               TAD CHAR
               AND K7      /MASK AC 9~11
               CLL RAL
               RTL      /POSITION TO AC 6~8
               DCA FIRST1
               JMS KEYIN
               TAD CHAR
               CIA
               TAD K254
               SNA      /OK IF 0
               JMP *3      /PRINT QUESTION MARK
               JMS QUERY
               JMP SLMTS*2
               /WAIT FOR 2ND
               JMS KEYIN
               JMS LEGAL
               TAD CHAR
               AND K7      /MASK AC 9~11
               CLL RAL
               RTL      /POSITION TO AC 6~8
               DCA LAST1
               TAD FIRST1
               CIA
               TAD LAST1
               SMA      /LAST IS > LAST IF NEG
               JMP OKAS
               CLA

```

```

1450    1075      TAD FIRST1
        3106      DCA CHAR
        1076      TAD LAST1      /LAST NOW IS FIRST
        3075      DCA FIRST1
        1106      TAD CHAR
        3076      DCA LAST1      /FIRST IS NOW LAST
        7200      OKAS,
        1076      TAD LAST1
        7041      CIA
        1075      TAD FIRST1      /SEE IF EQUAL
        7440      SZA      /YES IF 0
        5273      JMP ALOK
        1075      TAD FIRST1      /NOW SEE IF IT HAS PROGRAM
        7041      CIA
        1023      TAD INSFLD      /CURRENT FIELD
        7640      SZA CLA      /NO IF A 1
        5273      JMP ALOK
        4752      JMS I XLCAT      /PRINT PROGRAM LOCATION
        5211      JMP SLMTS+2      /AND START OVER
        4303      JMS KEYIN      /WAIT FOR C.R.
        1106      TAD CHAR
        7041      CIA
        7041      TAD K215      /NOT A C.R; IF A SKIP
        1116      SNA      /SLMTS
        7450      JMP I SLMTS
        5607      JMS QUERY      /PRINT QUESTION MARK
        4344      JMP SLMTS+2      /START OVER
        5211      /
        0000      KEYIN, 0
        6032      KCC
        6031      KSF
        5305      JMP ,=1
        6036      KRB
        3106      DCA CHAR
        1106      TAD CHAR
        4510      JMS I XPERR
        5703      JMP I KEYIN
        1503      /
        1504      /
        1505      /
        1506      /
        1507      /
        1510      /
        1511      /
        1512      /
        1513      /

```

```

/
  LEGAL,   0 TAD CHAR
  1514 0200 CIA
  1515 1106 TAD K377
  1516 7041 SNA CLA /RUB=QUIT IF 0
  1517 1117 JMP SLMTS+2
  1520 7650 TAD CHAR
  1521 5211 AND K370
  1522 1106 CIA
  1523 0120 TAD K260
  1524 7041 SNA CLA
  1525 1115 CIA
  1526 7650 TAD CHAR
  1527 5714 JMP I LEGAL
  1530 1106 CIA
  1531 7041 TAD K254
  1532 1353 SNA CLA /A COMMA IF 0
  1533 7650 CIA
  1534 5714 TAD CHAR
  1535 1106 JMP I LEGAL
  1536 7041 CIA
  1537 1116 TAD K215
  1540 7650 SNA CLA /A C,R, IF 0
  1541 5714 JMP I LEGAL
  1542 4344 JMS QUERY /QUERY
  1543 5211 JMP SLMTS+2 /START OVER

/
  QUERY,   0 JMS XCRLF
  1544 0000 TAD K277
  1545 4522 JMS XPERR /PRINT QUERY MARK
  1546 1121 JMP I QUERY
  1547 4510
  1550 5744
  1551 1646 /XTIM, TLIMT
  1552 1600 XLCAT, LOCAT
  1553 0254 K254,

```

```

/PRINT FIELD PROGRAM IS IN
/*1600
/
LOCAT,    CLA      /CURRENT FIELD
1600 0000
1601 7200
1602 1023
1603 7012
1604 7010
1605 0105
1606 1115
1607 3244
1610 1217
1611 3012
1612 1412
1613 7450
1614 5600
1615 4510
1616 5212
1617 1617
1620 0320
1621 0322
1622 0317
1623 0307
1624 0322
1625 0301
1626 0315
1627 0240
1630 0311
1631 0323
1632 0240
1633 0311
1634 0316
1635 0240
1636 0306
1637 0311
1640 0305
1641 0314
1642 0304
1643 0240
1644 0000
1645 0000

/LOCAT,    TAD  INSFLD
           RTR
           RAR
           AND K7
           TAD K260
           DCA FLDN
           TAD PRGAM
           DCA 12
           PLOCT, TAD I 12
           SNA
           JMP I LOCAT
           JMS I XPERR
           JMP PLOCT
           /
           PRGAM,
           '320
           /P
           /R
           /S
           /G
           /R
           /A
           /M
           /I
           /S
           /F
           /T
           /E
           /L
           /D
           /X
           FLDN,
           @
           /
TERMINATOR
/FLDN = PRGAM+25

```

```

    / TLIMT,   CLA          /PRINT TEST LIMITS
    1646 0000          TAD TSL
    1647 7200          DCA 12
    1650 1257          TAD 1 12
    1651 3012          SNA
    1652 1412          JMP I TLIMT
    1653 7450          JMS I XPERR
    1654 5646          JMP PLIMT
    1655 4510
    1656 5252

    / TSL,           'I
    1657 1657          324
    1660 0324          'I
    1661 0305          305
    1662 0323          'E
    1663 0324          323
    1664 0240          'S
    1665 0314          324
    1666 0311          'I
    1667 0315          315
    1670 0311          'M
    1671 0324          311
    1672 0323          'I
    1673 0000          323
                           'S
                           /TERMINATOR

    / HEADER ROUTINE
    PHDR,             JMS I XCRLF /CR, LF
    TAD FILD          TAD FILD
    DCA 12
    PFILD,            TAD I 12 /PRINT FIELD
    SNA
    /DONE IF 0
    JMP '+3
    JMS I XPERR
    JMP PFILD
    TAD M5
    DCA LOOP          DCA LOOP
    JMS I XPING        JMS I XPING
    TAD OTLDR         /SPACE 5
    DCA 12

```

```

1712 1412          POCDR, TAD 1 12      /PRINT OCTAL ADR
1713 7450          SNA           /DONE IF 0
1714 5317          JMP 1 *3
1715 4510          JMS 1 XPERR
1716 5312          JMP  POCUR
/
1717 1064          TAD M5
1720 3072          DCA LOOP
1721 4511          JMS 1 XPING
1722 1355          TAD GODD
1723 3012          DCA 12
1724 1412          TAD 1 12      /PRINT GOOD
1725 7450          PGOOD,
1726 5731          SNA           /DONE IF 0
1727 4510          JMP 1 *3
1730 5324          JMS 1 XPERR
1731 2000          JMP  PGOOD
                                BSPCE /NEXT PAGE
/
1732 1732          FILD,
1733 0306          306
1734 0311          311
1735 0305          305
1736 0314          314
1737 0304          304
1740 0000          0
/
1741 1741          OTLDR,
1742 0317          317
1743 0303          303
1744 0324          324
1745 0301          301
1746 0314          314
1747 0240          240
1750 0301          301
1751 0304          304
1752 0322          322
1753 0256          256
1754 0000          0
/
1755 1755          GODD,
1756 0307          307
1757 0317          317
1760 0317          317
1761 0304          304
1762 0000          0
/
1763 5674          EXHDR, JMP 1 PHDR

```

```

        /
        /*2700
        /
        BSPACE,      TAD M5
        DCA LOOP
        JMS I XPING
        TAD BADD
        DCA 12
        /
        PBAUD,      TAD I 12      /PRINT BAD
        SNA          /DONE IF 0
        JMP !*3
        JMS I XPIERR
        JMP PBAD
        /
        TAD M5
        DCA LOOP
        JMS I XPING
        /
        TAD TSTN
        DCA 12
        /
        PTSTN,      TAD I 12      /PRINT TEST
        SNA          /DONE IF 0
        JMP !*3
        JMS I XPIERR
        JMP PTSTN
        JMS I XCRLF
        JMP I XPHDR
        /
        TAD TSTN
        DCA 12
        /
        PTSTN,      TAD I 12      /PRINT TEST
        SNA          /DONE IF 0
        JMP !*3
        JMS I XPIERR
        JMP PTSTN
        JMS I XCRLF
        JMP I XPHDR
        /
        TSTN,       !324
        3026      /T
        3027      /E
        3028      /S
        3029      /T
        3030      /T
        3031      /A
        3032      /D
        3033      /T
        /
        BAUD,       !302
        3034      /B
        3035      /A
        3036      /A
        3037      /D
        3038      /T
        3039      /T
        3040      /T
        /
        HALT,       @
        HLT
        /
        JMP I HALT      /RESTART HEREOR RTRN1
        /
        XPHDR,      EXHDR
        /
        2000      1064
        2001      3072
        2002      4511
        2003      1234
        2004      3012
        2005      1412
        2006      7450
        2007      5212
        2008      4510
        2009      5205
        2010      5205
        2011      5205
        2012      1064
        2013      3072
        2014      4511
        2015      1226
        2016      3012
        2017      1412
        2018      7450
        2019      5224
        2020      4510
        2021      5217
        2022      4522
        2023      5217
        2024      4522
        2025      5644
        2026      2026
        2027      0324
        2028      0305
        2029      0323
        2030      0323
        2031      0324
        2032      0324
        2033      0000
        2034      2034
        2035      0302
        2036      0301
        2037      0304
        2038      0000
        2039      0000
        2040      0000
        2041      0000
        2042      7402
        2043      5641
        2044      1763
    
```

/ / WAIT HERE TO SETUP SR. TYPE CARRIAGE RETURN
 / AFTER SETTING SR,
 /

```

2045 0000           SETSR,    JMS I XFIELD      /RESTORE DATA FIELD
2046 4503           JMS I XCRLF      /CR, LF
2047 4522           TAD STSR
2050 1270           DCA 12
2051 3012           /PRINT SETUP SR
2052 1412           PSTSR,   TAD 1 12      /DONE IF @
2053 7450           SNA          JMP +3
2054 5257           JMS I XPERR
2055 4510           JMP PSTSR
2056 5252           KRB
2057 6036           KSF
2060 6031           JMP +4
2061 5265           KRB
2062 6036           JMS I XPERR
2063 4510           JMP I SETSR
2064 5645           LAS
2065 7604           DCA MCWA
2066 3037           JMP WTCR
2067 5260           /STSR,
2070 2070           'S
2071 0323           323
2072 0305           'E
2073 0324           324
2074 0325           'T
2075 0320           325
2076 0249           320
2077 0323           'S
2100 0322           322
2101 0000           @
2102 0000           STALL,   0
2103 4503           JMS I XFIELD
2104 4317           JMS GENRAN
2105 0316           AND K17
2106 7040           CMA
2107 3071           DCA FLCNT
2110 1020           TAD DATFLD
2111 1077           TAD KCDF
2112 3313           DCA +1
2113 6201           COF 00
2114 7200           CLA
2115 5702           JMP I STALL
2116 0017           K17,    0017

```

GENRAN, 2

2117	0000	TAD RANTAB
2120	1355	CIA
2121	7041	TAD RANDEX
2122	1343	SEA CLA
2123	7640	JMP RANTAO-1
2124	5334	TAD TBLRAN
2125	1356	DCA RANDEX
2126	3343	TAU RANCON
2127	1342	CLL RAL
2130	7104	SEL
2131	7430	TAD K1
2132	1123	DCA RANCON
2133	3342	TAD I RANDEX
2134	1743	RANTAU, TAD RANCON
2135	1342	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, 2154
2156	2144	TBLRAN, RANTBL

```

    /
    / ROUTINE TO DETERMINE FIELD FOR RELOCATION
    /
    *2200
    /
    CMOVE, JMS,1 XFLD   /SET DF TO CURRENT 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

    7600 TAD ERTBL      /SETUP ERROR TABLE POINTER
    DCA ERWD      /FIRST TESTED FIELD
    TAD FIRST1
    CIA
    TAD LAST1      /LAST TESTED FIELD
    SNA CLA      /DON'T MOVE IF EQUAL
    JMP 1 XRTN      /START OVER
    TAD FLAGS
    RAR
    SZL      /FIRST MOVE IF A SKIP
    JMP 1 XTMV      /SETUP FOR NEXT MOVE
    IAC
    DCA FLAGS

    TAD LAST1      /LAST TO TEST = 1ST MOVE
    DCA INSLD      /NEW CURRENT FIELD
    TAD INSLD
    TAD M10      /SUBTRACT1 FROM NEW CURRENT
    DCA NXLOC      /NXLOC=DOESTN FOR NEXT TIME
    RIF
    CIA
    TAD INSLD
    SNA CLA      /IS NEXT SAME AS CURRENT
    JMP SUB1      /YES, TRY NEXT LOWER FIELD

```

```

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

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

)
) 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 IXRTN /START OVER
) 2260 1426 TAD I ERWRD
) 2261 7650 SNA CLA /IS IT FIELD ??
) 2262 5266 JMP SUB1 /YES
) 2263 1023 TAD INSFLD /CURRENT NEXT
) 2264 1114 TAD M10 /SUBTRACT 1 FROM OF
) 2265 3124 DCA NXLOC

)
) 2266 1024 SUB1, TAD ERTBL
) 2267 3026 DCA ERWRD /RESTORE TABLE POINTER
) 2270 1124 TAD NXLOC
) 2271 7041 CIA
) 2272 1023 TAD INSFLD
) 2273 7650 SNA CLA /NEXT = CURRENT NEXT IF 0
) 2274 5253 JMP EQUAL
) 2275 1124 TAD NXLOC
) 2276 3023 DCA INSFLD /NEW CURRENT FIELD
) 2277 1023 TAD INSFLD
) 2300 7041 CIA /IS IT = LOWEST FIELD
) 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 OF
) 2306 3124 DCA NXLOC /NEXT FIELD LOWER
) 2307 5231 JMP CKERR
)
```

4/1/68 16:50.35

PAGE 34

/ STMV,
2310 7200 CLA
2311 1024 TAD ERTBL
2312 3026 DCA ERWRD
2313 6224 RIF
2314 3723 DCA I XSRCE
2315 1723 TAD I XSRCE
2316 7041 CIA
2317 1023 TAD INSFLD
2320 7650 SNA CLA /DON'T MOVE IF EQUAL
2321 5502 JMP I XRTN /START OVER
2322 5724 JMP I XMVE /GO MOVE

/ XSRCE,
2323 2522 SOURCE,
2324 2507 MOVE,
2325 2400 XTMV,
/ NXTMV,

```

*2400
  /      /600
  NXTMV, HIF      DCA SOURCE
  2401 6224      ISZ ERWD   /CURRENT FIELD
  2402 3322      TAD NXTMV /POINTER +1
  2403 2026
  2404 1200
  2405 7041
  2406 1426      TAD I ERWRD /NO ERRORS RECORDED IF 0
  2407 7650      SNA CLA   /INITIALIZE MOVE
  2410 5225      JMP STNXT
  2411 1426      TAD I ERWRD
  2412 7041      CIA
  2413 1124      TAD NXLOC /ERROR IN NEW FIELD IF 0
  2414 7650      SNA CLA   /TRY NEXT LOWER FIELD
  2415 5255      JMP SUB2
  2416 1026      TAD ERWRD
  2417 7041      CIA
  2420 1025      TAD ENTBL /DONE WITH TABLE IF 0
  2421 7650      SNA CLA   /INITIALIZE MOVE
  2422 5225      JMP STNXT /POINTER +1
  2423 2026      ISZ ERWD
  2424 5211      JMP CKNXT

  /      TAD ERTBL
  STNXT, DCA ERWD /RESTORE TABLE POINTER
  2425 1024      TAD NXLOC
  2426 3026      CIA
  2427 1124      TAD INSFLD /NEXT=CURRENT IF 0
  2430 7041      SNA CLA   /NEXT LOWER FIELD
  2431 1023      JMP CKNT
  2432 7650      TAD NXLOC
  2433 5242      CIA
  2434 1124      TAD FIRST1 /NEXT = LOWEST IF 0
  2435 7041      SZA CLA   /MOVE TO LOWEST TEST FIELD
  2436 1075      JMP STNXT
  2437 7640      JMP MVBK
  2440 5247      TAD NXLOC
  2441 5302      CIA
  2442 1124      TAD FIRST1 /NEXT = LOWEST IF 0
  2443 7041      SNA CLA   /SETUP TO MOVE TO HIGHEST
  2444 1075      JMP NXTHI /NEXT LOWER FIELD
  2445 7650      TAD NXLOC /IS NOW CURRENT FIELD
  2446 5275      DCA INSFLD
  2447 1124      TAD M10   /SUBTRACT 1 FROM NEW
  2450 3023      DCA NXLOC /NEW NEXT LOWER FIELD
  2451 1023      JMP MOVE /GO MOVE
  2452 1114
  2453 3124
  2454 5307
  /

```

```

2455 1024      TAD ERTBL      / RESTORE TABLE POINTER
2456 3026      DCA ERWDL     /NEXT LOWER FIELD
2457 1124      TAD NXLOC
2460 7450      SNA
2461 5502      JMP I XRTN    /FIELD 0 IF 0
2462 1114      TAD M10     /START OVER CAN'T MOVE
2463 3124      DCA NXLOC   /SUBTRACT 1
2464 1124      TAD NXLOC   /NOW = 2 FIELDS LOWER
2465 7041      CIA
2466 1023      TAD INSFID  /CURRENT FIELD
2467 7640      SZA CLA     /ARE THEY EQUAL
2470 5203      JMP CHNXT   /NO
2471 1124      TAD NXLOC   /YES
2472 7450      SNA
2473 5203      JMP CHNXT   /DOES IT = FIELD 0
2474 5262      JMP SUB2+5  /YES
2475 1076      JMP CHNXT   /NO
2476 3124      TAD LAST1   /VERY LAST TO TEST
2477 1076      DCA NXLOC   /MAKE IT NEXT FIELD
2500 3023      TAD LAST1
2501 5203      DCA INSFID
2502 1124      JMP CHNXT
2503 3023      MVBK,      TAD NXLOC
2504 6224      DCA INSFID
2505 3322      RIF
2506 3021      DCA SOURCE
                           DCA FLAGS /CLEAR BIT 11

```

```

/
/ROUTINE TO RELOCATE 4K FIELDS
/
MOVE, TAU KCDF          /6201
2507 1077               /CURRENT FIELD
2510 1322               /SOURCE NOW = CDF N
2511 3322               /SOURCE NOW = CDF N
2512 1077               /6201
2513 1023               /NEW FIELD
2514 3324               /DESTN NOW = CDF V
2515 7040               /DESTN COUNT
2516 3010               /SOURCE COUNT
2517 7040               /DESTINATION COUNT
2520 3011               /4K COUNTER
2521 3072               /4K COUNTER
2522 0000               /WILL = CDF N
2523 1410               /TAKE FROM HERE
2524 0000               DESTN, 0
2525 3411               DCA I 11           /PUT IN HERE
2526 2072               ISZ LOOP          /DONE 4K WHEN SKIP
2527 5322               JMP SOURCE        /KEEP MOVING
2530 1100               TAD KCIF          /6202
2531 3023               TAD INSFLD        /NEW FIELD
2532 3333               DCA *1
2533 6202               CIF 00           /CHANGE TO NEW FIELD
2534 5502               JMP I XRTN         /EXIT TO RTN1 IN
                                         /NEW FIELD
$
```

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
CKDNO	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
EPRINT	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
MCM	0037
MEMADR	0074
MOVE	2507
MVBK	2502
M1	0142
M10	0114
M13	0112
M14	0113
M20	0061
M4	0063
M40	0062
M5	0064
NXLLOC	0124
NXTBNK	1103
NXTHI	2475
NXTHV	2400
OACDR	1365
OKAS	1456
ONES	1057
OTLDR	1741
OVER	1145
PATR	0126

SYMBOL TABLE

PBAD	2005
PFILD	1700
PG000	1724
PHDR	1674
PLIMT	1652
PLOCT	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	6214
RFR1	0720
RFR2	0736
RIF	6224
RROT	0441
RR1T	0604
RSL0	0600
RSL1	0436
RSTRTR1	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
STSX	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
WRIT	0520
WTCR	2060
XBAK	0132
XBAK2	0136
XBANK	0066
XCRLF	0122
XFILD	0103
XFWD	0131
XFWD2	0135
XHDR	0107
XHLT	0101
XLCAT	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
XSLD	0552
XSRCE	2323
XSTS R	0275
XST4	0653
XTB NK	0067
XTLIM	1551
XTMV	2325
XTST1	0047
XTST2	0050
XTST3	0051
XTST4	0052
XT4	0551
XWRB	0133
XWRB2	0137
XWRF	0130
XWRF2	0134

4/1/68 16:51.2

PAGE 42

SYMBOL TABLE

SYMBOL TABLE

DATFLD	0020
FLAGS	0021
XLMTS	0022
INSFLD	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
XTBNK	0067
COUNT	0070
FLCNT	0071
LOOP	0072
XRROR	0073
MEMADR	0074
FIRST1	0075
LAST1	0076
KCDF	0077
KCIF	0100
XHLT	0101
XRTN	0102
XFILE	0103
XPRER	0104
K7	0105
CHAR	0106
XHDR	0107
XPER	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
XFW02	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
WR0T	0414
RSL1	0436
RR0T	0441
CKDN1	0470
EXT3	0501
TST4	0503
WR1T	0520
XT4	0551
XSL0	0552
RSL0	0600
RR1T	0604
CKDN0	0642
EXT4	0650
XRT4	0652
XST4	0653
WFRI	0654

SYMBOL TABLE

WFR2	0662
WRB1	0671
WRB2	0724
RFR1	0720
RFR2	0736
RHK1	1000
LBK1	1007
RBK2	1030
LBK2	1036
ONES	1057
X\$ALL	1065
CBANK	1066
SAVIF	1102
NXTBNK	1103
FIELD	1125
OVER	1145
WRLOP	1147
CFLD	1161
ERROR	1200
SW1	1233
SW2	1242
EREXT	1246
EPRNT	1254
SW@	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
PLOC	1612
PRGM	1617
FLDN	1644
TLIMIT	1646
PLINT	1652
TSTL	1657

SYMBOL TABLE

PHDR	1674
PFILE	1720
POCDR	1712
PGOOD	1724
FILD	1732
OTLDR	1741
GODD	1755
EXHDR	1763
BSPCE	2000
PBAD	2005
PTSTN	2017
TSTN	2026
BADD	2034
HALT	2041
XPHDR	2044
SETSR	2045
PSTSR	2052
WTCR	2060
STSRR	2070
STALL	2102
K17	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
NXTH1	2475
MVBK	2502
MOVE	2507
SOURCE	2522
DESTN	2524
CDF	6201
CIF	6202
RDF	6214
RIF	6224

SYMBOL TABLE

4/1/68 16:51,27

PAGE 47