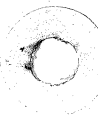
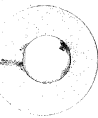
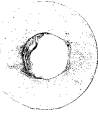


IDENTIFICATION

PRODUCT CODE: MAINDEC-08-DIEC-D
PRODUCT NAME: PDP-8, 8/I EXTENDED MEMORY
CHECKERBOARD
DATE CREATED: NOVEMBER 1, 1971
MAINTAINER: DIAGNOSTIC GROUP
AUTHOR: J. RICHARDSON - J. VROBEL

COPYRIGHT© 1971
DIGITAL EQUIPMENT
CORPORATION



1. ABSTRACT

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

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

2. REQUIREMENTS

2.1 Equipment

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

2.2 Storage

The program occupies locations 0010 to 3334.

2.3 Preliminary Programs

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

3. LOADING PROCEDURE

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

4. STARTING PROCEDURE

4.1 Starting Address

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

4.2 Restarting Address

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

4.3 Operator Action

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

The following rules govern the amount of memory to test:

- a. Type two octal numbers, separating the numbers with a comma. The first number signifies the lowest order 4K stack to test; the second signifies the highest order.
- b. The program expects the 4K stacks to be numbered sequentially starting with a stack 0.
- c. If the highest order stack to test is typed as the first stack, the program will interchange the two values so as to make the second value the first to test.
- d. After typing the second octal number, press the carriage return key to terminate the line.
- e. The program will test the lowest and highest order 4K stack specified, plus every stack between, starting with the lowest specified.
- f. Any single stack, or two or more sequential stacks may be specified.
- g. The stack containing the program may be included when specifying two or more stacks.

The stack containing the program will be tested after automatic program relocation takes place (see Section 5.3.1).

- h. If a typing error is made, press the RUB-OUT key. TEST LIMITS will be printed again. All previous input is disregarded.

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

Example A:

TEST LIMITS

0,7↵ (↵ denotes carriage return)

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

Example B:

TEST LIMITS

7,0

The program will perform exactly as Example A.

Example C:

TEST LIMITS

4,5

Only stacks 4 and 5 will be tested.

Example D:

TEST LIMITS

3,3

Stack 3 alone will be tested.

Example E:

TEST LIMITS

0,0 PROGRAM IS LOCATED IN FIELD 0

TEST LIMITS

0,1

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

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

4.3.1 Setup SR

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

5. OPERATING PROCEDURE

5.1 Program and Operator Action

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

5.2 Operational Switch Settings

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

5.3 Subroutine Abstracts

5.3.1 Program Relocation

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

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

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

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

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

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

5.3.2 The Checkerboard Patterns

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

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

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

- a. Write the pattern once in all stacks selected for testing; starting with the lowest order stack.
- b. Select the lowest order stack and perform a read, complement data, write sequence once on each location, until all 4K has been complemented.
- c. Repeat step b 31 more times. The stack will end up with the pattern originally loaded.

No error checking has been performed as yet.

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

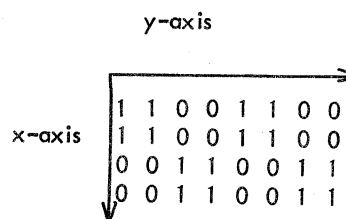
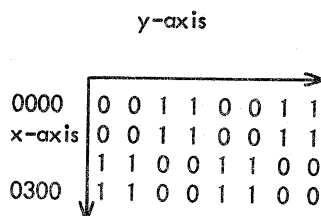
e. After checking the entire 4K stack for errors, repeat step d again. This time stall for a random period of time after reading and checking every 400₈ word block. The maximum stall is 18.4 ms; the minimum is 3 μs.

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

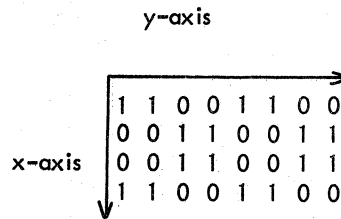
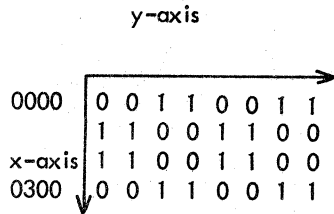
When all selected stacks have been checked the next test in sequence is executed, and steps a through f repeated. Program relocation takes place after the fourth test is executed in this manner.

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

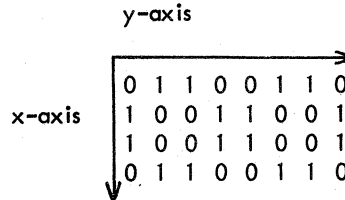
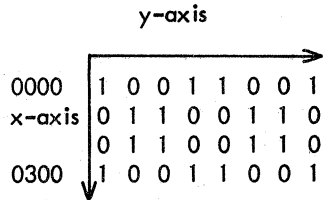
Test 1:



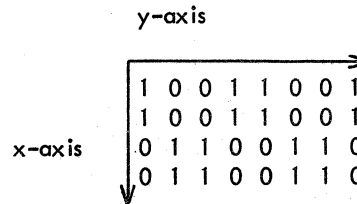
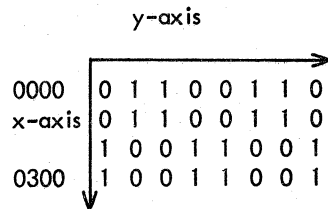
Test 2:



Test 3:



Test 4:



6. ERRORS

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

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

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

6.1 Error Halts and Description

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

7. RESTRICTIONS

7.1 Starting Restrictions

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

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

7.2 Operating Restrictions

None

8. MISCELLANEOUS

8.1 Execution Time

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

8.2 Applications

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

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

8.2.1 Halt after Test or Error - SR 0

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

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

8.2.2 Inhibit Error Print-out - SR 1

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

8.2.3 Bell on Error - SR 2

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

8.2.4 Test Selection SR 3 through 6

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

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

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

8.2.5 Inhibit Program Relocation - SR 7

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

8.2.6 SR 8, 9 and 10 - Not Used

8.2.7 Change TEST LIMITS and SR - SR 11

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

8.2.8 Loop on Address

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

Operating Procedure:

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

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

9. PROGRAM DESCRIPTION

The PDP-8, 8/1 Extended Memory Checkerboard diagnostic is designed to create worst case memory noise conditions on systems equipped with 8K to 32K words of memory. The program executes four checkerboard patterns, plus their complements, on each 4K memory field. In addition, the program automatically relocates from field to field in order to test all 4K fields from every 4K field. Under normal operation, the amount of core memory tested at one time is that specified by the operator minus the 4K field containing the program. A TTY keyboard input routine is provided to enable the operator to specify the exact number of 4K fields to be tested. A print-out is provided for each error detected by the program.

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

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

/ PDP-9, 81, 8S EXTENDED MEMORY CHECKBOARD TEST,
 / COPYRIGHT 1971, DIGITAL EQUIPMENT CORP., MAYNARD, MASS.
 / START AT 200, RESTART AT 211 TO SKIP HEADER.
 / MIN. OF 8K OF CORE REQUIRED.
 /
 / *1

0001			
	5000	0001	JMP .
	0	0002	0002
		0003	0003
		6201	CDF#6201
		6202	CIF#6202
		6214	RDF#6214
		6224	RIF#6224
	0004	0000	0
	0005	0000	0
	0006	0000	0
	0007	0000	0
	4535	0010	0
	4535	0011	0
	2402	0012	0
		0013	0
		0014	0
		0015	0
		0016	0
		0017	0
		0020	0
		0021	0
		0022	0
		0032	0
		0022	0
		0023	0
		0024	0
		0025	0
		0026	0
		0027	0
		0030	0
		0031	0
		0032	0
		0033	0
0740		0034	0
		0010	0
140		0740	0
		0400	0
		0037	0
		0040	0
		0041	0
		0042	0
		0043	0
		0044	0
		0045	0
		0046	0
		0047	0
		0053	0
		0051	0
		0052	0

DAYFLD,
 FLAGS,
 INSFLD,
 FLOAD,
 ERTBL,
 ENTBL,
 ERWRD,
 ERWRD,
 ERWRD*10
 ERWRD,
 ERWRD,
 ERWRD,
 ERWRD,
 ERWRD,
 ERWRD
 MCWA,
 K10,
 K740,
 K400,
 K200,
 K100,
 K40,
 K20,
 XLMTS,
 XTST1,
 XTST2,
 XTST3,
 XTST4,
 XMOVE,
 XSETU,
 K261,
 SLMTS
 TST1
 TST2
 TST3
 TST4
 CMOVE
 SETU1
 261

0053	0262	K262,	262
0054	0263	K263,	263
0055	0264	K264,	264
0056	7760	X20,	7760
0057	7740	M40,	7740
0060	7774	M4,	7774
0061	7773	M5,	7773
0062	0000	TNUM,	0
0063	1607	XBANK,	CBANK
0064	0652	X0011,	W0011
0065	0667	X1100,	W1100
0066	0704	X0110,	W0110
0067	0721	X1001,	W1001
0070	1600	XKBANK,	CKBANK
0071	1624	XTBANK,	NXTBANK
0072	0000	COUNT,	0
0073	0000	FLOUNT,	0
0074	0000	LOOP,	0
0075	0736	XRALL,	RDALL
0076	1037	XCHK1,	RCHK1
0077	1054	XCHK1C,	RCHK1C
0100	1071	XCHK2,	RCHK2
0101	1106	XCHK2C,	RCHK2C
0102	1123	XCHK3,	RCHK3
0103	1140	XCHK3C,	RCHK3C
0104	1200	XCHK4,	RCHK4
0105	1217	XCHK4C,	RCHK4C
0106	1056	TDM20,	TAD M20
0107	1097	TDM40,	TAD M40
0110	4515	JMS1,	JMS I XRD1
0111	4516	JMS2,	JMS I XRD2
0112	4517	JMS3,	JMS I XRD3
0113	4520	JMS4,	JMS I XRD4
0114	4592	JMS5,	JMS I XSALL
0115	1245	RD1,	RD1
0116	1322	RD2,	RD2
0117	1400	RD3,	RD3
0120	1455	RD4,	RD4
0121	2000	ERROR,	ERROR
0122	0000	MEMADR,	0
0123	0000	FIRST1,	0
0124	0000	LAST1,	0
0125	6201	KCDF,	6201
0126	6202	KCIF,	6202
0127	2641	XHLT,	HALT
0130	0213	XRTN,	RTN1
0131	1646	XFILD,	FEILD
0132	2146	XPRER,	PRERR
0133	0007	K7,	7
0134	0000	CHAR,	0
0135	2474	XHDR,	PHDR
0136	2146	XPERR,	PRERR
0137	2115	XPING,	SPING

0261

7253
4536

10

0245

0140 7764
 0141 7770
 0142 0260
 0143 0215
 0144 0377
 0145 0370
 0146 0277
 0147 2154
 0150 0001
 0151 0000
 0152 2702
 0153 0000
 0154 2166

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

RTU

0320

0200
 0201 6002
 0202 7200
 0203 3015
 0204 6224
 0205 3016
 0206 4443
 0207 4677
 0210 4535
 0213 5213

*200
 BEGIN,
 IOF
 CLA FLAGS
 DCA FLAGS
 RIF
 DCA INSFLD
 JMS I XLMTS
 JMS I XLSTSR
 JMS I XHDR
 JMP RTN1

/PI OFF

/CLEAR PROGRAM FLAGS

/SETUP TEST LIMITS
 /SETUP SR
 /PRINT HEADER

/RESTART HERE

0211 4443
 0212 4677
 0213 6224
 0214 3016
 0215 4531
 0216 1141
 0217 3074
 0220 7600
 0221 1220
 0222 2022
 0223 3422
 0224 2074
 0225 5220
 0226 1220
 0227 3554
 0230 1020
 0231 3022

RSTR1, JMS I XLMTS
 RTN1, JMS I XLSTSR
 RIF
 DCA INSFLD
 JMS I XFILD
 TAD M10
 DCA LOOP
 7600
 TAD ALAW
 ISZ ERWRD
 DCA I ERWRD
 ISZ LOOP
 JMP ALAW
 TAD ALAW
 DCA I LASTX
 TAD ERTBL
 DCA ERWRD

/SET TEST LIMITS
 /SETUP SR INSTRUCTION FIELD
 /READ INSTRUCTION FIELD
 /CURRENT FIELD
 /-10

/EXAMINE SR

0232 1033
 0233 0035
 0234 7440
 0235 5241
 0236 1033
 0237 1035

TAD MCWA
 AND K740
 SZA
 JMP EXAM1
 TAD MCWA
 TAD K740

/DO ALL IF 0
 /MADK 3,4,5 AND 6

/SET ALL TEST BITS

```

0240 3033 DCA MCWA /SAVE
0241 7200 CLA
0242 1033 TAD MCWA
0243 0036 AND K400
0244 7440 SEA
0245 5444 JMP I XTST1
0246 7200 CLA MCWA
0247 1033 TAD MCWA
0250 0037 AND K200
0251 7440 SEA
0252 5445 JMP I XTST2
0253 7200 CLA MCWA
0254 1033 TAD MCWA
0255 0040 AND K100
0256 7440 SEA
0257 5446 JMP I XTST3
0260 7200 CLA MCWA
0261 1033 TAD MCWA
0262 0041 AND K40
0263 7440 SEA
0264 5447 JMP I XTST4
0265 4531 JMS I XFILE

```

/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

XSTSR, SETSR

/TEST 1. WRITE CHECKER PATTERN #1.

```

0300 4451 JMS I XSETU /SET DF TO 1ST FIELD
0301 7200 CLA /TEST NUMBER
0302 1052 TAD K261 /SET ADDRESS COUNT TO 7777
0303 3062 DCA TNUM /SEE IF FIELD HAS PROGRAM
0304 7040 CMA /NO. BEGIN WRITING
0305 3010 DCA 10 /DONE ALL. NOW READ ALL
0306 4463 JMS I XBANK
0307 7410 SKP EXT1
0310 5322 JMP EXT1
0311 1345 TAD KXT1
0312 3153 DCA EXIT
0313 4464 JMS I X0011 /WRITE 0011
0314 4464 JMS I X0011 /WRITE 0011 64 TIMES
0315 4465 JMS I X1100 /WRITE 1100 128 TIMES
0316 4465 JMS I X1100

```


0317 5313
 0320 4471
 0321 5304
 0322 4476
 0323 5324

JMP :=4 /KEEP WRITING
 JMS I XTBNK /SETUP FOR NEXT FIELD
 JMP TST1+4
 XIT1,
 EXT1, JMS I XCHK1 /READ EACH FIELD AND CHECK
 /FOR ERRORS,
 /NOW WRITE COMPLEMENT
 JMP TST1C

/WRITE COMPLEMENT OF PATTERN 1

0324 4451
 0325 7240
 0326 3010
 0327 4463
 0330 7410
 0331 5343
 0332 1346
 0333 3133
 0334 4465
 0335 4465
 0336 4464
 0337 4464
 0340 5334
 0341 4471
 0342 5325

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

0343 4477
 0344 5246
 0345 0320
 0346 0341

EXT1C, JMS I XCHK1C /READ EACH BANK AND CHECK
 /FOR ERRORS,
 /SEE IF TEST 2 IS SELECTED
 JMP EXAM2
 KXT1, XIT1
 KXT1C, XIT1C

/TEST 2. WRITE CHECKER PATTERN #2

0400
 0400 4451
 0401 7200
 0402 1053
 0403 3062
 0404 7240
 0405 3010
 0406 4463
 0407 7410
 0410 5223
 0411 1250
 0412 3153
 0413 4464
 0414 4465
 0415 4464
 0416 4464
 0417 4464

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

0420 JMP ;=4
 0421 JMS I XTBNK /SETUP FOR NEXT FIELD
 0422 JMP TST2+4
 0423 JMS I XCHK2 /READ EACH FIELD AND CHECK
 0424 JMP TST2C /NOW WRITE COMPLEMENT

/WRITE COMPLEMENT OF PATTERN 2

/TST2C: JMS I XSETU /SET OF FOR FIRST FIELD

0425 CLA CMA

0426 DCA 10 /SET ADR. COUNT TO 7777

0427 JMS I XBANK /SEE IF FIELD HAS PROGRAM

0428 SKP EXT2C /WRITE

0429 JMP EXT2C /GO READ

0430 TAD KXT2C

0431 DCA EXIT

0432 JMS I X1100 /WRITE 1100

0433 JMS I X0011 /WRITE 0011 128 TIMES

0434 JMS I X0011

0435 JMS I X1100 /WRITE 1100 128 TIMES

0436 JMS I X1100

0437 JMP ;=4

0438 JMS I XTBNK /SETUP FOR NEXT FIELD

0439 JMP TST2C+1

0440

0441

0442

0443

0444

0445

0446

0447

0448

0449

0450

0451

0452

0453

0454

0455

0456

0457

0458

0459

0460

0461

0462

0463

0464

0465

0466

0467

0468

0469

0470

0471

0472

0473

/TEST 3. WRITE CHECKER PATTERN #3

/TST3: JMS I XSETU /SETUP FOR 1ST FIELD

0452 CLA K263 /TEST NUMBER

0453 DCA TNUM

0454 CLA CMA

0455 DCA 10 /SET ADR. COUNT TO 7777

0456 JMS I XBANK /SEE IF FIELD HAS PROGRAM

0457 SKP EXT3 /GO WRITE

0458 JMP EXT3 /GO READ

0459 TAD KXT3

0460 DCA EXIT

0461 JMS I X1001 /WRITE 1001

0462 JMS I X0110 /WRITE 0110 128 TIMES

0463 JMS I X0110

0464 JMS I X1001 /WRITE 1001 128 TIMES

0465 JMS I X1001

0466 JMP ;=4

0467 JMS I XTBNK /SETUP FOR NEXT FIELD

0468

0469

0470

0471

0472

0473

PAL10 V141
2474 5256

2 NOV 71

2119

PAGE 1 OF 6

EXT3, JMS I XCHK3 /READ EACH FIELD AND CHECK
JMP TST3+4 /WRITE COMPLEMENT

/WRITE COMPLEMENT OF PATTERN 3

TST3C, JMS I XSETU /SETUP DF FOR 1ST FIELD
CLA CMA /SET ADR, COUNT TO 7777
DCA 10 /SEE IF FIELD HAS PROGRAM
JMS I XBANK /WRITE
SKP /READ ALL

JMP EXT3C /WRITE 0110
TAD KXT3C /WRITE 1001 128 TIMES
DCA EXIT /WRITE 0110 128 TIMES

XIT3C, JMS I XIBNK /SETUP FOR NEXT FIELD
JMP TST3C+1

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

KXT3, XIT3
KXT3C, XIT3C

/TEST 4. WRITE PATTERN #4

*600

TST4, JMS I XSETU /SET DF FOR 1ST FIELD

CLA K264 /TEST NUMBER
DCA TNUM

CLA CMA /SET ADR, COUNT TO 7777
DCA 10 /SEE IF FIELD HAS PROGRAM
JMS I XBANK /WRITE
SKP /GO READ

JMP EXT4 /WRITE 0110
TAD KXT4 /WRITE 0110 64 TIMES
DCA EXIT /WRITE 1001 128 TIMES

XIT4, JMS I XIBNK /SETUP FOR NEXT FIELD

0600 4451
0601 7200
0602 1055
0603 3062
0604 7240
0605 3010
0606 4463
0607 7410
0610 5223
0611 1245
0612 3153
0613 4466
0614 4466
0615 4467
0616 4467
0617 4466
0620 5214
0621 4471

```

0622 5204 JMP TST4*4
0623 4504 /EXT4, JMS I XCHK4 /READ EACH FIELD AND CHECK
0624 5225 JMP TST4C /WRITE COMPLEMENT

/WRITE COMPLEMENT OF PATTERN 4
TST4C, JMS I XSETU /SET DF FOR FIRST
        CLA CMA
        DCA I 10 /SET ADR. COUNT TO 7777
        JMS I XBANK /SEE IF FIELD HAS PROGRAM
        SKP /WRITE
        JMP EXT4C /READ
        TAD KXT4C
        DCA EXT4
        JMS I X1001
        JMS I X1001
        JMS I X0110 /WRITE 1001 64 TIMES
        JMS I X0110 /WRITE X0110 128 TIMES
        JMS I X1001
        JMP :+4
        JMS I XTBNK /SETUP FOR NEXT FIELD
        JMP TST4C*1

/
KXT4, XIT4
KXT4C, XIT4C
    
```

```

0647 4505 /EXT4, JMS I XCHK4C /READ EACH FIELD AND CHECK
0650 5651 JMP I :+1 /SEE IF READY TO MOVE
0651 0265 EXAM4*5

/ROUTINE TO WRITE 0011
W0011, 0
        TAD M20
        DCA COUNT
        DCA I 10
        DCA I 10
        CMA
        DCA I 10
        CMA
        DCA I 10
        ISZ COUNT /COUNT = 16 OR -32
        JMP W0011*3 /LOOP
        JMS I XBANK /SEE IF END OF FIELD
        JMP I W0011 /EXIT
    
```

```

/ROUTINE TO WRITE 1100
W1100, 0
        TAD M20
        DCA COUNT
        CMA
        DCA I 10
    
```

PAL10 V141 0674 7040
0675 3410
0676 3410
0677 3410
0700 2072
0701 5272
0702 4470
0703 5667

2-NOV-71

0119

PAGE 1-8

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

/ROUTINE TO WRITE 0110

W0110, 0
0704 0000
0705 1056
0706 3072
0707 3410
0710 7040
0711 3410
0712 7040
0713 3410
0714 3410
0715 2072
0716 5307
0717 4470
0720 5704

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

/ROUTINE TO WRITE 1001

W1001, 0
0721 0000
0722 1056
0723 3072
0724 7040
0725 3410
0726 3410
0727 3410
0730 7040
0731 3410
0732 2072
0733 5324
0734 4470
0735 5721

TAD M20
DCA COUNT /1
DCA I 10 /0
DCA I 10 /0
CMA
DCA I 10 /1
DCA I 10 /-16 TO -32
ISZ COUNT /LOOP
JMP W1001+3 /SEE IF END OF FIELD
JMS I XKBK /EXIT
JMP I W1001

/ROUTINE TO READ ALL OF MEMORY 8 TIMES, COMPLEMENTING
/THE PATTERN EACH PASS. NO ERROR CHECKING IS DONE.

RDALL, 0
0736 0000
0737 7200
0740 1057
0741 3072
0742 7240
0743 3010
0744 7040

CLA
TAD M40 /-32 DECIMAL
DCA COUNT /COUNTS PASSES THRU MEMORY
CLA CMA
DCA 10 /SET ADR. REGS. TO 777
CMA

```

0745 3011 DCA I1
0746 3073 DCA FLCNT
0747 4463 JMS I XBANK /SEE IF FIELD HAS PROGRAM
0750 7410 SKP /READ
0751 5360 JMP CCON1 /DONE
0752 7200 CLA /READ ONE
0753 1410 TAD I 10
0754 7040 CMA
0755 3411 DCA I 11 /WRITE BACK
0756 2073 ISZ FLCNT /DONE 1 FIELD WHEN SKIP
0757 5353 JMP RDLOP
0760 2072 ISZ COUNT /DONE 32 PASSES WHEN SKIP
0761 5342 JMP RDALL*4 /DO ANOTHER PASS
0762 5736 JMP I RDALL /EXIT

```

RDLOP:

CCON1:

```

/READ AND CHECK FOR ERROR ROUTINE
/
*1000
/

```

1000

RCHKA:

```

1000 0000 /SET OF TO 1ST FIELD
1001 4451 JMS I XSETU /READ ALL, DON'T CHECK
1002 4475 JMS I XRALL /SET ADR, COUN TO 0
1003 3122 DCA MEMADR /SEE IF FIELD HAS PROGRAM
1004 4463 JMS I XBANK
1005 7410 SKP
1006 5600 JMP I RCHKA
1007 1235 TAD KRXT
1010 3153 DCA EXIT
1011 0000
1012 0000
1013 0000
1014 0000
1015 0000
1016 7000
1017 5212
1020 1216
1021 1236
1022 7640
1023 5227
1024 1114
1025 3216
1026 5203
1027 7000
1030 1227
1031 3216
1032 4471
1033 5202
1034 5600

```

RLOPA:

0 0 0 0 0 0 /WILL 0 JMS I XRD1, 2, 3, OR 4

RXIT:

0 NOP RLOPA+1 /WILL 0 NOP OR JMS STALL

KRXT:

K1K: 1000

/SETUP ROUTINES FOR RCHKA

/SETUP FOR NEXT FIELD

1037 0000 /
 1040 1110 /
 1041 3211 /
 1042 1110 /
 1043 3212 /
 1044 1111 /
 1045 3213 /
 1046 1111 /
 1047 3214 /
 1050 1110 /
 1051 3215 /
 1052 4200 /
 1053 5637 /

Ø
 TAD JMS1 /
 DCA RLOPA /
 TAD JMS1 /
 DCA RLOPA+1 /
 TAD JMS2 /
 DCA RLOPA+2 /
 TAD JMS2 /
 DCA RLOPA+3 /
 TAD JMS1 /
 DCA RLOPA+4 /
 JMS RCHKA /
 JMP I RCHK1 /

/JMS1 = JMS I XRD1
 /JMS1 = JMS I XRD1

/GO READ
 /EXIT

1054 0000 /
 1055 1111 /
 1056 3211 /
 1057 1111 /
 1060 3212 /
 1061 1110 /
 1062 3213 /
 1063 1110 /
 1064 3214 /
 1065 1111 /
 1066 3215 /
 1067 4200 /
 1070 5654 /

Ø
 TAD JMS2 /
 DCA RLOPA /
 TAD JMS2 /
 DCA RLOPA+1 /
 TAD JMS1 /
 DCA RLOPA+2 /
 TAD JMS1 /
 DCA RLOPA+3 /
 TAD JMS2 /
 DCA RLOPA+4 /
 JMS RCHKA /
 JMP I RCHKIC /

/JMS2 = JMS I XRD2

/GO READ
 /EXIT

1071 0000 /
 1072 1110 /
 1073 3211 /
 1074 1111 /
 1075 3212 /
 1076 1111 /
 1077 3213 /
 1100 1110 /
 1101 3214 /
 1102 1110 /
 1103 3215 /
 1104 4200 /
 1105 5671 /

Ø
 TAD JMS1 /
 DCA RLOPA /
 TAD JMS2 /
 DCA RLOPA+1 /
 TAD JMS2 /
 DCA RLOPA+2 /
 TAD JMS1 /
 DCA RLOPA+3 /
 TAD JMS1 /
 DCA RLOPA+4 /
 JMS RCHKA /
 JMP I RCHK2 /

/JMS1 = JMS I XRD1
 /JMS I XRD2
 /JMS I XRD1

/GO READ
 /EXIT

1106 0000 /
 1107 1111 /
 1110 3211 /
 1111 1110 /
 1112 3212 /
 1113 1110 /
 1114 3213 /
 1115 1111 /
 1116 3214 /
 1117 1111 /
 1120 3215 /
 1121 4200 /

Ø
 TAD JMS2 /
 DCA RLOPA /
 TAD JMS1 /
 DCA RLOPA+1 /
 TAD JMS1 /
 DCA RLOPA+2 /
 TAD JMS2 /
 DCA RLOPA+3 /
 TAD JMS2 /
 DCA RLOPA+4 /
 JMS RCHKA /

/JMS I XRD1

/GO READ

1122 5706 JMP I RCHK2C /EXIT

```

/ RCHK3, 0
1123 0000 TAD JMS4 /JMS I XRD4
1124 1113 DCA RLOPA /JMS I XRD3
1125 3211 TAD JMS3 /JMS I XRD3
1126 1112 DCA RLOPA+1 /JMS I XRD3
1127 3212 TAD JMS3 /JMS I XRD3
1130 1112 DCA RLOPA+2 /JMS I XRD4
1131 3213 TAD JMS4 /JMS I XRD4
1132 1113 DCA RLOPA+3 /GO READ
1133 3214 TAD JMS4 /EXIT
1134 1113 DCA RLOPA+4 /EXIT
1135 3215 JMS RCHKA /EXIT
1136 4200 JMP I RCHK3 /EXIT
1137 5723

```

```

/ RCHK3C, 0
1140 0000 TAD JMS3 /JMS I XRD3
1141 1112 DCA RLOPA /JMS I XRD4
1142 3211 TAD JMS4 /JMS I XRD4
1143 1113 DCA RLOPA+1 /JMS I XRD4
1144 3212 TAD JMS4 /JMS I XRD4
1145 1113 DCA RLOPA+2 /JMS I XRD4
1146 3213 TAD JMS3 /JMS I XRD4
1147 1112 DCA RLOPA+3 /JMS I XRD4
1150 3214 TAD JMS3 /JMS I XRD4
1151 1112 DCA RLOPA+4 /GO READ
1152 3215 JMS RCHKA /EXIT
1153 4200 JMP I RCHK3C /EXIT
1154 5740

```

```

*1200
/ RCHK4, 0
1200 0000 JMS I XFILD /JMS I XRD3
1201 4531 TAD JMS3 /JMS I XRD3
1202 1112 DCA I XLOPA /JMS I XRD3
1203 3637 TAD JMS3 /JMS I XRD3
1204 1112 DCA I XLOPB /JMS I XRD3
1205 3640 TAD JMS4 /JMS I XRD3
1206 1113 DCA I XLOPC /JMS I XRD4
1207 3641 TAD JMS4 /JMS I XRD4
1210 1113 DCA I XLOPD /JMS I XRD4
1211 3642 TAD JMS3 /JMS I XRD4
1212 1112 DCA I XLOPE /JMS I XRD4
1213 3643 JMS I XCFL /GO READ
1214 4644 JMS I XCHKA /EXIT
1215 4636 JMP I RCHK4 /EXIT
1216 5600

```

```

/ RCHK4C, 0
1217 0000 JMS I XFILD /JMS I XRD4
1220 4531 TAD JMS4 /JMS I XRD4
1221 1113 DCA I XLOPA /JMS I XRD4
1222 3637 TAD JMS4 /JMS I XRD4
1223 1113

```



```

1224 3640 DCA I XLOPB
1225 1112 TAD JMS3
1226 3641 DCA I XLOPC
1227 1112 TAD JMS3
1230 3642 DCA I XLOPD
1231 1113 TAD JMS4
1232 3643 DCA I XLOPE
1233 4644 JMS I XCFL
1234 4636 JMS I XCHKA
1235 5617 JMP I RCHK4C /EXIT
                                           /GO READ

```

```

1236 1000 XCHKA,
1237 1011 XLOPA,
1240 1012 XLOPB,
1241 1013 XLOPC,
1242 1014 XLOPD,
1243 1015 XLOPE,
1244 1722 XCFL,

```

PAUSE

/01-08 EXTENDED CHECKERBOARD - TAPE 2
 /READ ROUTINES FOR 0011; 1100; 0110 AND 1001

```

1245 0000 R01,
1246 1036 TAD M20 /016
1247 3072 DCA COUNT
1250 1000 TAD M4 /04
1251 3073 DCA FLCNT /08
1252 1141 TAD M10
1253 3074 DCA LOOP
1254 1522 TAD I MEMADR
1255 7040 CMA
1256 3522 DCA I MEMADR
1257 2074 ISZ LOOP
1260 5254 JMP ,04
1261 2073 ISZ FLCNT
1262 7410 SKP ,+3
1263 5266 JMP ,+3
1264 2122 ISZ MEMADR
1265 5252 JMP CLOP1
                                           /COMPLEMENT 8 TIMES
                                           /DONE 4 ADRS; WHEN SKIP

1266 1122 TAD MEMADR
1267 1060 TAD M4 /SUBTRACT 4
1270 3010 DCA 10 /NOW USE AUTO=INDEX
1271 7100 CLL
1272 1410 TAD I 10
1273 7440 SZA
1274 4521 JMS I XRROR /0
1275 7100 CLL /PRINT ERROR
1276 1410 TAD I 10
1277 7440 SZA /0
1300 4521 JMS I XRROR /PRINT ERROR

```

1301	7120	STL	
1302	1410	TAD I 10	/1
1303	7040	CMA	
1304	7440	SZA	
1305	4521	JMS I XRROR	/PRINT ERROR
1306	7120	STL	
1307	1410	TAD I 10	/1
1310	7040	CMA	
1311	7440	SZA	
1312	4521	JMS I XRROR	/PRINT ERROR
1313	2072	ISE COUNT	
1314	5320	JMP I +4	
1315	4470	JMS I XKBK	/SEE IF END OF FIELD
1316	2122	ISE MEMADR	
1317	5645	JMP I RD1	
1320	2122	ISE MEMADR	/KEEP READING
1321	5250	JMP RD1+3	

1322	0000	RD2, 0	
1323	1056	TAD M20	/-16
1324	3072	DCA COUNT	
1325	1060	TAD M4	/-4
1326	3073	DCA FLCNT	
1327	1141	TAD M10	/-8
1330	3074	DCA LOOP	
1331	1522	TAD I MEMADR	/READ
1332	7040	CMA	
1333	3522	DCA I MEMADR	/COMPLEMENT 4 TIMES
1334	2074	ISE LOOP	
1335	5331	JMP I-4	
1336	2073	ISE FLCNT	/DONE 4 ADRS. WHEN SKIP
1337	7410	SKP	
1340	5343	JMP I+3	
1341	2122	ISE MEMADR	/INCREMENT ADDRESS
1342	5327	JMP CLOP2	

1343	1122	TAD MEMADR	
1344	1060	TAD M4	
1345	3010	DCA 10	/NOW USE AUTO-INDEX
1346	7120	STL	
1347	1410	TAD I 10	/1
1350	7040	CMA	
1351	7440	SZA	
1352	4521	JMS I XRROR	/PRINT ERROR
1353	7120	STL	
1354	1410	TAD I 10	/1
1355	7040	CMA	
1356	7440	SZA	
1357	4521	JMS I XRROR	/PRINT ERROR
1360	7100	CLL	
1361	1410	TAD I 10	

```

1362 7440 SZA /0
1363 4521 JMS I XRROR /PRINT ERROR
1364 7100 CLL
1365 1410 TAD I 10
1366 7440 SZA /0
1367 4521 JMS I XRROR /PRINT ERROR
1370 2072 ISZ COUNT
1371 5375 JMP :+4
1372 4470 JMS I XKBK
1373 2122 ISZ MEMADR
1374 5722 JMP I RD2 /SEE IF END OF FIELD

1375 2122 ISZ MEMADR /KEEP READING
1376 5325 JMP RD2+3

```

1400

```

1400 0000 /
1401 1056 RDS,
1402 3072 /-16
1403 1060 /+4
1404 3073 DCA FLCNT
1405 1141 TAD M10 /-8
1406 3074 DCA LOOP /READ
1407 1322 TAD I MEMADR
1410 7040 CMA I MEMADR
1411 3522 ISZ LOOP
1412 2074 JMP :+4 /COMPLEMENT 8 TIMES
1413 5207 ISZ FLCNT /DONE 4 IF 0
1414 7410 SKP :+3
1415 7410 JMS MEMADR
1416 5221 ISZ MEMADR
1417 2122 JMP CLOPS /DO NEXT
1420 5205

```

```

1421 1122 TAD MEMADR
1422 1060 TAD M4
1423 3010 DCA I 10 /USE AUTO=INDEX
1424 7100 CLL
1425 1410 TAD I 10
1426 7440 SZA /0
1427 4521 JMS I XRROR /PRINT ERROR
1430 7120 STL
1431 1410 TAD I 10 /1
1432 7040 CMA /PRINT ERROR
1433 7440 SZA /1
1434 4521 JMS I XRROR /PRINT ERROR
1435 7120 STL
1436 1410 TAD I 10 /1
1437 7040 CMA /PRINT ERROR
1440 7440 SZA /1
1441 4521 JMS I XRROR /PRINT ERROR
1442 7100 CLL

```

1443 1410 TAD I 10
 1444 7440 SZA
 1445 4521 JMS I XRROR
 1446 2072 ISZ COUNT
 1447 5253 JMP I +4
 1450 4470 JMS I XKBNK
 1451 2122 ISZ MEMADR
 1452 5600 JMP I RD3
 1453 2122 ISZ MEMADR
 1454 5203 JMP RD3+3

RD4,
 CLOP4,
 0
 1455 0000 TAD M20 /-16
 1456 1056 DCA COUNT
 1457 3072 TAD M4 /-4
 1460 1060 DCA FLCNT /-8
 1461 3073 TAD M10
 1462 1141 DCA LOOP
 1463 3074 TAD I MEMADR
 1464 1522 CMA I MEMADR
 1465 7040 DCA I MEMADR
 1466 3522 ISZ LOOP
 1467 2074 JMP I +4
 1470 5264 ISZ FLCNT
 1471 2073 SKP I +3
 1472 7410 JMP I +3
 1473 5276 ISZ MEMADR
 1474 2122 JMS CLOP4
 1475 5262 JMS CLOP4
 1476 1122 TAD MEMADR
 1477 1060 TAD M4
 1500 3010 DCA 10
 1501 7120 STL
 1502 1410 TAD I 10
 1503 7040 CMA
 1504 7440 SZA
 1505 4521 JMS I XRROR
 1506 7100 CLL
 1507 1410 TAD I 10
 1510 7440 SZA
 1511 4521 JMS I XRROR
 1512 7100 CLL
 1513 1410 TAD I 10
 1514 7440 SZA
 1515 4521 JMS I XRROR
 1516 7120 STL
 1517 1410 TAD I 10
 1520 7040 CMA
 1521 7440 SZA
 1522 4521 JMS I XRROR
 1523 2072 ISZ COUNT
 1524 5330 JMP I +4
 1525 4470 JMS I XKBNK

/COMPLEMENT 8 TIMES

/DONE 4 ADRS. WHEN SKIP

/INCREMENT ADDRESS

/USE AUTO=INDEX

/1

/PRINT ERROR

/0

/PRINT ERROR

/0

/PRINT ERROR

/1

/PRINT ERROR

/SEE IF END OF FIELD

FAL10 V141
1526 2122
1527 5655
1530 2122
1531 5260

2-NOV-71 0119 PAGE 1-16

ISZ MEMADR
JMP I RD4
ISZ MEMADR
JMP RD4+3

/ROUTINE TO CHECK FOR END OF FIELD
/

1600
1600 0000
1601 7200
1602 1010
1603 7040
1604 7640
1605 5600
1606 5553

*1600
CKBNK, 0
CLA
TAD 10
CMA
SZA CLA
JMP I CKBNK /NOT DONE
JMP I EXIT /DONE

/ROUTINE TO SEE IF TESTED FIELD HAS PROGRAM
/

1607 0000
1610 6224
1611 3223
1612 6214
1613 7041
1614 1223
1615 7640
1616 5607
1617 4471
1620 5607
1621 2207
1622 5607
1623 0000

CBANK, 0
RIF
DCA SAVIF /READ INST. FIELD
RDF /SAVE
CIA /READ DATA FIELD
TAD SAVIF
SZA CLA /EQUAL IF AC#0
JMP I CBANK /DOESN'T HAVE PROGRAM
JMS I XTBNK /INCREMENT DATA FIELD
JMP I CBANK /TEST NEW FIELD
ISZ CBANK /DONE ALL CAUSE PROGRAM NOW
JMP I CBANK /IN HIGHEST FIELD
SAVIF, 0 /EXIT

/ROUTINE TO SET DF FOR NEXT FIELD
/

1624 0000
1625 7200
1626 6214
1627 7041
1630 1124
1631 7640
1632 5235
1633 2224
1634 5242
1635 6214
1636 1034
1637 1125
1640 3241
1641 6201

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

/CHECK SWITCH REGISTER

1642 7634
1643 7712
1644 4527
1645 5624

LAS
SPA CLA /CHECK HALT
JMS I XHLT /GO HALT, SRC=1
JMP I NXTBANK /EXIT

/RESTORE DATA FIELD AND CHECK SR

1646 2000
1647 7200
1650 6214
1651 3014
1652 6224
1653 1125
1654 3255
1655 6201
1656 7200
1657 5646

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

/ *1700

1700

/START HERE TO LOOP ON ADDRESS

1700 7200
1701 7604
1702 3123
1703 7402
1704 7604
1705 3124
1706 1123
1707 3122
1710 1522
1711 3522
1712 1122
1713 7041
1714 1124
1715 7650
1716 5306
1717 2122
1720 5310
1721 7402

CLA
LAS
DCA FIRST1 /READ LOWER LIMIT
HLT /NOW SETUP UPPER LIMIT
LAS LAST1
TAD FIRST1
DCA MEMADR /READ
TAD I MEMADR /WRITE
DCA I MEMADR
TAD MEMADR
CIA
TAD LAST1
SNA CLA
JMP OVER
ISZ MEMADR
JMP WRLOP
HLT

/ CFLD,

1722 2000
1723 7200
1724 1014
1725 1125
1726 3327
1727 6201
1730 7200
1731 5722

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

/PRINT ERROR ROUTINE

```

2000 /
2001 /
2002 /
2003 /
2004 /
2005 /
2006 /
2007 /
2010 /
2011 /
2012 /
2013 /
2014 /
2015 /
2016 /
2017 /
2020 /
2021 /
2022 /
2023 /
2024 /
2025 /
2026 /
2027 /
2030 /
2031 /
2032 /
2033 /
2034 /
2035 /
2036 /
2037 /
2040 /
2041 /
2042 /
2043 /
2044 /
2045 /
2046 /
2047 /
2050 /
2051 /
2052 /
2053 /

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

SZA
CMA
DCA BAD
SZL
CMA
DCA GOOD
TAD 10
DCA OCADR
JMS I XFILD
TAD DATFLD
CIA
TAD LAST
SNA CLA
JMP SW2
TAD DATFLD
DCA LAST
TAD ERWRD
CIA
TAD ENTBL
SZA CLA
JMP +3
TAD ERIBL
DCA ERWRD
TAD DATFLD
ISZ ERWRD
DCA I ERWRD

LAS
RTL
SMA CLA
JMP SW1
TAD K207
JMS PRERR
JMP SW0
LAS
RAL
SMA CLA
JMP EPRNT
TAD DATFLD
TAD KCDF
DCA +1
CDF 00
CLA
JMP I ERROR

SW2,
SW1,
EREXT,

EPRNT,
JMS CRLF
TAD DATFLD
RAR
RAR

```

```

2054 /CR,LF
2055 /TEST NUMBER
2056 /
2057 /

```

2060 1142 TAD K260 /PRINT
 2061 4346 JMS PRERR /-12 DECIMAL
 2062 1142 TAD M14 /SPACE 12
 2063 3074 DCA LOOP /OCTAL ADR;
 2064 4315 JMS SPING /SAVE
 2065 1365 TAD OCADR /PRINT
 2066 3134 DCA CHAR /-8 DECIMAL
 2067 4325 JMS PROCTL /SPACE 8
 2070 1141 TAD M10 /PRINT
 2071 3074 DCA LOOP /-5
 2072 4315 JMS SPING /SPACE 5
 2073 1364 TAD GOOD /PRINT
 2074 3134 DCA CHAR /-5
 2075 4326 JMS PROCTL /PRINT
 2076 1061 TAD M5 /SPACE 5
 2077 3074 DCA LOOP /PRINT
 2100 4315 JMS SPING /SPACE 5
 2101 1363 TAD BAD /PRINT
 2102 3134 DCA CHAR /PRINT
 2103 4326 JMS PROCTL /PRINT
 2104 1061 TAD M5 /SPACE 5
 2105 3074 DCA LOOP /TEST NUMBER
 2106 4315 JMS SPING /PRINT
 2107 1062 TAD TNUM /PRINT
 2110 4346 JMS PRERR /PRINT

/ SW0, LAS SPA CLA /CH CK SR0 /GO HALT
 2111 7604 JMS I XHLT /EXIT
 2112 7710 JMP EREXT /EXIT
 2113 4527 /PRINT SPACES
 2114 5246 / SPING, 0 TAD K240 /SPACE
 2115 0000 TLS TSP /PRINT
 2116 1370 JMP ,=1 /PRINT
 2117 6046 ISE LOOP /PRINT
 2120 6041 JMP SPING+2 /PRINT
 2121 5320 CLA /PRINT
 2122 2074 JMP I SPING /EXIT
 2123 5317 /PRINT OCTAL
 2124 7200 /
 2125 5715 PROCTL, 0 TAD M4 /-4
 POSITN, TAD LOOP /DIGIT COUNTER
 CLL RAL
 RTL
 DCA CHAR

2126 0000
 2127 1060
 2130 3074
 2131 1134
 2132 7104
 2133 7006
 2134 3134

2135 1134 TAD CHAR
 2136 7004 RAL
 2137 0133 AND K7
 2140 1142 TAD K260 /MAKE ASCII
 2141 4346 JMS PRERR /PRINT ONE
 2142 2074 ISZ LOOP
 2143 5331 JMP POSITN /DO NEXT
 2144 7200 CLA
 2145 5726 JMP I PROCTL /EXIT

/PRINT A NUMBER
 /PRERR, 0
 2146 0000 TLS
 2147 6046 TSF
 2150 6041 JMP I=1
 2151 5350 CLA
 2152 7200 JMP I PRERR /EXIT
 2153 5746

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

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

/+2200
 /ROUTINE TO SET DF TO FIRST TEST FIELD

/SETUI, 0
 2200 0000 CLA
 2201 7200 TAD FIRST1 /FIRST TO TEST
 2202 1123 TAD KCDF
 2203 1125 DCA I+1
 2204 3205

/CDF 00 /CHANGE TO TEST FIELD
 /JMP I SETUI /EXIT
 /ROUTINE TO ACCEPT TEST LIMITS FROM
 /KEYBOARD INPUT

2205 6201
 2206 5600

2207 0000
 2210 4531
 2211 4547
 2212 4751
 2213 4547
 2214 4303
 2215 4314
 2216 1134
 2217 0133
 2220 7104
 2221 7006
 2222 3123
 2223 4303
 2224 1134
 2225 7041
 2226 1353
 2227 7450
 2230 5233
 2231 4344
 2232 5211
 2233 4303
 2234 4314
 2235 1134
 2236 0133
 2237 7104
 2240 7006
 2241 3124
 2242 1123
 2243 7041
 2244 1124
 2245 7500
 2246 5256
 2247 7200

SLMTS: 2
 JMS I XFILD
 JMS I XCRLF
 JMS I XTLLIM
 JMS I XCRLF
 JMS KEYIN
 JMS LEGAL
 TAD CHAR
 AND K7
 CLL RAL
 RTL
 DCA FIRST1
 JMS KEYIN
 TAD CHAR
 CIA
 TAD K254
 SNA
 JMP :+3
 JMS QUERY
 JMP SLMTS+2
 JMS KEYIN
 JMS LEGAL
 TAD CHAR
 AND K7
 CLL RAL
 RTL
 DCA LAST1
 TAD FIRST1
 CIA
 TAD LAST1
 SMA OKAS
 JMP OKAS
 CLA

/CR, LF
 /PRINT TEST LIMITS
 /CR, LF
 /GO ACCEPT INPUT
 /SEE IF IT'S LEGAL
 /MASK AC 9=11
 /POSITION TO AC 6=8
 /FIRST TO TEST
 /WAIT FOR COMMA
 /GET INPUT
 /OK IF 0
 /PRINT QUESTION MARK
 /WAIT FOR 2ND
 /SEE IF IT'S LEGAL
 /MASK AC 9=11
 /POSITION TO AC 6=8
 /LAST TO TEST
 /1ST IS > LAST IF NEG

2250 1123
 2251 3134
 2252 1124
 2253 3123
 2254 1134
 2255 3124
 2256 7200
 2257 1124
 2260 7041
 2261 1123
 2262 7440
 2263 5273
 2264 1123
 2265 7041
 2266 1016
 2267 7640
 2270 5273
 2271 4752
 2272 5211
 2273 4303

OKAS,
 TAD FIRST1
 DCA CHAR
 TAD LAST1
 DCA FIRST1
 TAD CHAR
 DCA LAST1
 CLA
 TAD LAST1
 CIA
 TAD FIRST1
 SEA
 JMP ALOK
 TAD FIRST1
 CIA
 TAD INSFLD
 SEA CLA
 JMP ALOK
 JMS I XLCAT
 JMP SLMTS+2
 JMS KEYIN
 ALOK,

/LAST NOW IS FIRST
 /FIRST IS NOW LAST
 /SEE IF EQUAL
 /YES IF 0
 /NOW SEE IF IT HAS PROGRAM
 /CURRENT FIELD
 /NO IF A 1
 /PRINT PROGRAM LOCATION
 /AND START OVER
 /WAIT FOR C.R.

2274 1134 TAD CHAR
 2275 7041 CIA
 2276 1143 TAD K215
 2277 7450 SNA
 2300 5607 JMP I SLMTS
 2301 4344 JMS QUERY
 2302 5211 JMP SLMTS+2
 / KEYIN,
 2303 0000
 2304 6032 KCC
 2305 6031 KSF
 2306 5305 JMP ,=1
 2307 6036 KRB CHAR
 2310 3134 DCA CHAR
 2311 1134 TAD CHAR
 2312 4536 JMS I XPERR
 2313 5703 JMP I KEYIN

/NOT A C.R. IF A SKIP

/PRINT QUESTION MARK
/START OVER

2314 0000
 2315 1134 TAD CHAR
 2316 7041 CIA
 2317 1144 TAD K377
 2320 7650 SNA CLA
 2321 5211 JMP SLMTS+2
 2322 1134 TAD CHAR
 2323 0145 AND K370
 2324 7041 CIA
 2325 1142 TAD K260
 2326 7650 SNA CLA
 2327 5714 JMP I LEGAL
 2330 1134 TAD CHAR
 2331 7041 CIA
 2332 1353 TAD K254
 2333 7650 SNA CLA
 2334 5714 JMP I LEGAL
 2335 1134 TAD CHAR
 2336 7041 CIA
 2337 1143 TAD K215
 2340 7650 SNA CLA
 2341 5714 JMP I LEGAL
 2342 4344 JMS QUERY
 2343 5211 JMP SLMTS+2

/RUB-OUT IF 0

/A COMMA IF 0

/A C.R. IF 0

/QUERY
/START OVER

2344 0000
 2345 4547 JMS I XCRLF
 2346 1146 TAD K277
 2347 4536 JMS I XPERR
 2350 5744 JMP I QUERY
 / QUERY,
 2351 2446 XTLIM,
 2352 2400 XLCAT,
 2353 0254 K254,

/PRINT QUERY MARK

TLIMT
LOCAT
254

6211

2455 4536
 2456 5252
 2457 2457
 2460 0324
 2461 0305
 2462 0323
 2463 0324
 2464 0240
 2465 0314
 2466 0311
 2467 0315
 2470 0311
 2471 0324
 2472 0323
 2473 0000

JMS I XPERR
 JMP PLIMT

/TSTL

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

/HEADER ROUTINE

2474 0000
 2475 4547
 2476 1332
 2477 3012
 2500 1412
 2501 7450
 2502 5305
 2503 4536
 2504 5300
 2505 1061
 2506 3074
 2507 4537
 2510 1341
 2511 3012

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

/PHDR
 /CR, LF
 /PFILD
 /DONE IF 0
 /SPACE 5

2512 1412
 2513 7450
 2514 5317
 2515 4536
 2516 5312
 2517 1061
 2520 3074
 2521 4537
 2522 1355
 2523 3012
 2524 1412
 2525 7450
 2526 5731
 2527 4536
 2530 5324
 2531 2600
 2532 2532
 2533 0306

TAD I 12
 SNA
 JMP I +3
 JMS I XPERR
 JMP POCOR
 TAD M5
 DCA LOOP
 JMS I XPING
 TAD GOOD
 DCA 12
 TAD I 12
 SNA
 JMP I +3
 JMS I XPERR
 JMP PGOOD
 BSPCE

/POCOR
 /DONE IF 0
 /PRINT OCTAL ADR
 /SPACE 5
 /PGOOD
 /DONE IF 0
 /PRINT GOOD
 /FIELD
 /NEXT PAGE
 /F

2534 V141 2-NOV-71 0119 /I
 2535 0311 /E
 2536 0305 /L
 2537 0314 /D
 2540 0304 0
 2541 0311 /O
 2542 0317 /C
 2543 0303 /T
 2544 0324 /A
 2545 0301 /L
 2546 0314 /A
 2547 0240 /D
 2550 0301 /R
 2551 0304 /G
 2552 0322 /O
 2553 0256 /O
 2554 0000 /D
 2555 2555 /G
 2556 0307 /O
 2557 0317 /O
 2560 0317 /O
 2561 0304 /D
 2562 0000 /D
 2563 5674 /D

/ OTLDR:
 317
 303
 324
 301
 314
 240
 301
 304
 322
 256 0
 307
 317
 317
 304 0

/ GOOD:
 307
 317
 317
 304 0

/ EXHDR: JUMP I PHDR
 307
 317
 317
 304 0

/ *2600
 / BSPCE:
 TAD M5
 DCA LOOP
 JMS I XPING /SPACE 5
 TAD BADD
 DCA 12
 TAD I 12 /PRINT BAD
 SNA :+3 /DONE IF 0
 JMP :+3
 JMS I XPERR
 JMP PBAD
 TAD M5
 DCA LOOP /SPACE 5
 JMS I XPING

/ PTSTN:
 TAD TSTN
 DCA 12
 TAD I 12
 SNA :+3
 JMP :+3
 JMS I XPERR
 JMP PTSTN
 JMS I XPERR
 JMP I XCRLF
 JMS I XPHDR

2600 1061
 2601 3074
 2602 4537
 2603 1234
 2604 3012
 2605 1412
 2606 7450
 2607 5212
 2610 4536
 2611 5205
 2612 1061
 2613 3074
 2614 4537

2615 1226
 2616 3012
 2617 1412
 2620 7450
 2621 5224
 2622 4536
 2623 5217
 2624 4547
 2625 5644

/CR, LF
 /EXIT

```

2626 2626 TSTN,
2627 0324 /T
2630 0305 /E
2631 0323 /S
2632 0324 /T
2633 0000
2634 2634 BADD,
2635 0302 /B
2636 0301 /A
2637 0304 /D
2640 0000
2641 0000 HALT,
2642 7402 HLT
2643 5641 JMP I HALT /RESTART HERE OR RTRN1
2644 2563 XPHDR, EXHDR

```

```

2645 0000 SETSR, 0
2646 4531 JMS I XFILD /RESTORE DATA FIELD
2647 4547 JMS I XCRLF /CR, LF
2650 1270 TAD STSR
2651 3012 DCA 12
2652 1412 TAD I 12 /PRINT SETUP SR
2653 7450 SNA /DONE IF 0
2654 5257 JMP :+3
2655 4536 JMS I XPERR
2656 5252 JMP PSTSR
2657 6036 KRB
2660 6031 KSF
2661 5265 JMP :+4
2662 6036 KRB
2663 4536 JMS I XPERR
2664 5645 JMP I SETSR
2665 7604 LAS
2666 3033 DCA MCWA
2667 5260 JMP WTCR
2670 2670 SETSR,
2671 0323 /S
2672 0305 /E
2673 0324 /T
2674 0325 /U
2675 0320 /P
2676 0240 /S
2677 0323 /S

```

2-NOV-71 0:19 /R

2700 2322
 2701 2000
 2702 2000
 2703 4531
 2704 4316
 2705 3074
 2706 2074
 2707 5306
 2710 1014
 2711 1125
 2712 3313
 2713 6201
 2714 7200
 2715 5702

/STALL,
 JMS I XFILD
 JMS GENRAN /GET ANOTHER
 DCA LOOP
 ISZ LOOP /18.5 MS MAX.
 JMP I-1
 TAD DATFLD
 TAD KCOF
 DCA I-1
 CDF 00
 CLA
 JMP I STALL
 /RESTORE DATA FIELD
 /EXIT

GENRAN, 0
 TAD RANTAB
 CIA
 TAD RANDEX
 SZA CLA
 JMP RANTAD-1
 TAD TBLRAN
 DCA RANDEX
 TAD RANCON
 CLL RAL
 SZL
 TAD K1
 DCA RANCON
 TAD I RANDEX
 RANTAD, TAD RANCON
 DCA I RANDEX
 TAD I RANDEX
 ISZ RANDEX
 JMP I GENRAN

2716 0000
 2717 1354
 2720 7041
 2721 1342
 2722 7640
 2723 5333
 2724 1355
 2725 3342
 2726 1341
 2727 7104
 2730 7430
 2731 1150
 2732 3341
 2733 1742
 2734 1341
 2735 3742
 2736 1742
 2737 2342
 2740 5716

RANCON, 1234
 RANDEX, RANTBL+10
 RANTBL, 4321
 1416
 5363
 6060
 3035
 2572
 3237
 0214
 0
 RANTAB, I-1
 TBLRAN, RANTBL
 K177, 177

3412-2741 1234
 -2742 2753
 2353-2743 4321
 7150-2744 1416
 3415-2745 5363
 4112-2746 6060
 1007-2747 3035
 0624 2750 2572
 1271 2751 3237
 6246 2752 0214
 2753 0000
 2754 2753
 2755 2743
 2756 0177


```

3000
3001 4531
3002 7600
3003 1020
3004 3022
3005 1123
3006 7041
3007 1124
3008 7650
3009 5530
3010 1015
3011 7010
3012 7430
3013 5725
3014 7001
3015 3015
3016
3017 1124
3018 3016
3019 1016
3020 1141
3021 3151
3022 6224
3023 7041
3024 1016
3025 7650
3026 5266
3027
3028
3029
3030
    JMS I XFILE /SET DF TO CURRENT FIELD
    TAD ERTBL /SETUP ERROR TABLE POINTER
    DCA ERWRD /FIRST TESTED FIELD
    TAD FIRST1
    CIA
    TAD LAST1 /LAST TESTED FIELD
    SNA CLA /DON'T MOVE IF EQUAL
    JMP I XRTN /START OVER
    TAD FLAGS
    RAR
    SEL /FIRST MOVE IF A SKIP
    JMP I XTMV /SETUP FOR NEXT MOVE
    IAC /SET BIT 11
    DCA FLAGS
    TAD LAST1 /LAST TO TEST # 1ST MOVE
    DCA INSFLD /NEW CURRENT FIELD
    TAD INSFLD
    TAD M10 /SUBTRACT 1 FROM NEW CURRENT
    DCA NXLOG /NXLOG=DESTIN FOR NEXT TIME
    RIF
    CIA
    TAD INSFLD
    SNA CLA /IS NEXT SAME AS CURRENT
    JMP SUB1 /YES, TRY NEXT LOWER FIELD
  
```

/CHECK FOR ERROR IN NEW FIELD

```

3031 2022
3032 1201
3033 7041
3034 1422
3035 7650
3036 5310
3037 1422
3038 7041
3039 1016
3040 7650
3041 5253
3042 1022
3043 7041
3044 1021
3045 7650
3046 5310
3047 2022
3048 5237
    CKERR, ISZ ERWRD /POINTER+1
    TAD CMOVE+1
    CIA
    TAD I ERWRD
    SNA CLA /NO ERRORS RECORDED IF 0
    JMP STMV /INITIALIZE MOVE
    CNXT, TAD I ERWRD
    CIA
    TAD INSFLD
    SNA CLA /ERROR IN NEW FIELD IF 0
    JMP EQUAL
    TAD ERWRD
    CIA
    TAD ENTBL
    SNA CLA /ENTBL=ERWRD+10
    JMP STMV /TABLE DONE IF 0
    ISZ ERWRD /INITIALIZE MOVE
    JMP CNXT /POINTER+1
  
```

```

3053 1422 / EQUAL, TAD I ERWRD /GET ERROR FIELD
3054 7041 CIA /DON'T MOVE IF = TO FIRST
3055 1123 TAD FIRST1 /START OVER
3056 7650 SNA CLA /IS IT FIELD 0?
3057 5530 JMP I XRTN /YES
3060 1422 TAD I ERWRD /CURRENT NEXT
3061 7650 SNA CLA /SUBTRACT 1 FROM DF
3062 5266 JMP SUB1
3063 1016 TAD M10
3064 1141 DCA NXLOC
3065 3151

```

```

SUB1. TAD ERTBL /RESTORE TABLE POINTER
3066 1020 DCA ERWRD
3067 3022 TAD NXLOC
3070 1151 CIA
3071 7041 TAD INSFLD
3072 1016 SNA CLA /NEXT = CURRENT NEXT IF 0
3073 7650 JMP EQUAL /NEW CURRENT FIELD
3074 5253 TAD NXLOC
3075 1151 DCA INSFLD
3076 3016 TAD INSFLD
3077 1016 CIA /IS IT = LOWEST FIELD
3100 7041 TAD FIRST1 /YES
3101 1123 SNA CLA /CURRENT NEW FIELD
3102 7650 JMP CKERR /SUBTRACT 1 FROM DF
3103 5251 TAD INSFLD /NEXT FIELD LOWER
3104 1016 TAD M10
3105 1141 DCA NXLOC
3106 3151 JMP CKERR
3107 5231

```

```

STMV. CLA ERTBL /RESTORE TABLE POINTER
3110 7200 TAD ERWRD
3111 1020 DCA ERWRD
3112 3022 RIF
3113 6224 DCA I XSRCE
3114 3723 TAD I XSRCE
3115 1723 CIA
3116 7041 TAD INSFLD
3117 -1016 SNA CLA /DON'T MOVE IF EQUAL
3120 7650 JMP I XRTN /START OVER
3121 5530 JMP I XMVE /GO MOVE
3122 5724

```

```

XSRCE, SOURCE
3123 3323 XHVE, MOVE
3124 3307 XTMV, NXTMV
3125 3200
*3200
3200 7600 / NXTMV, 7600

```

3201 6224 RIF
 3202 3323 DCA SOURCE /CURRENT FIELD
 3203 2022 ISZ ERWRD /POINTER +1
 3204 1200 TAD NXTMV
 3205 7041 CIA
 3206 1422 TAD I ERWRD
 3207 7650 SNA CLA
 3210 5225 JMP STNXT
 3211 1422 TAD I ERWRD
 3212 7041 CIA
 3213 1151 TAD NXLOC
 3214 7650 SNA CLA
 3215 5255 JMP SUB2
 3216 1022 TAD ERWRD
 3217 7041 CIA
 3220 1021 TAD ENTBL
 3221 7650 SNA CLA
 3222 5225 JMP STNXT
 3223 2022 ISZ ERWRD
 3224 5211 JMP CKNXT

/NO ERRORS RECORDED IF 0
/INITIALIZE MOVE

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

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

/ STNXT,
 3225 1020 TAD ERTBL
 3226 3022 DCA ERWRD /RESTORE TABLE POINTER
 3227 1151 TAD NXLOC /NEXT LOWER FIELD
 3230 7041 CIA
 3231 1016 TAD INSFLD
 3232 7650 SNA CLA
 3233 5242 JMP CKNT
 3234 1151 TAD NXLOC
 3235 7041 CIA
 3236 1123 TAD FIRST1
 3237 7640 SZA CLA
 3240 5247 JMP STNX
 3241 5302 JMP MVBK
 3242 1151 TAD NXLOC
 3243 7041 CIA
 3244 1123 TAD FIRST1
 3245 7650 SNA CLA
 3246 5275 JMP NXTHI
 3247 1151 TAD NXLOC
 3250 3016 DCA INSFLD
 3251 1016 TAD INSFLD
 3252 1141 TAD M10
 3253 3151 DCA NXLOC
 3254 5307 JMP MOVE

/RESTORE TABLE POINTER
/NEXT LOWER FIELD

/NEXT=CURRENT IF 0

/NEXT = LOWEST IF 0

/MOVE TO LOWEST TEST FIELD

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

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

3255 1020 TAD ERTBL
 3256 3022 DCA ERWRD /RESTORE TABLE POINTER
 3257 1151 TAD NXLOC /NEXT LOWER FIELD
 3260 7450 SNA
 3261 5530 JMP I XRTRN
 3262 1141 TAD M10 /FIELD 0 IF 0
/START OVER CAN'T MOVE
/SUBTRACT 1

/RESTORE TABLE POINTER
/NEXT LOWER FIELD
/FIELD 0 IF 0
/START OVER CAN'T MOVE
/SUBTRACT 1

```

3263 DCA NXLOC
3264 TAD NXLOC
3265 CIA
3266 TAD INSFLO
3267 SZA CLA
3270 JMP CHNXT
3271 TAD NXLOC
3272 SNA
3273 JMP CHNXT
3274 JMP SUB2+5

3275 NAXHI, TAD LAST1
3276 DCA NXLOC
3277 TAD LAST1
3300 DCA INSFLO
3301 JMP CHNXT

3302 MVBK, TAD NXLOC
3303 DCA INSFLO
3304 RIF
3305 DCA SOURCE
3306 DCA FLAGS

```

/ROUTINE TO RELOCATE 4K FIELDS

```

3307 MOVE, TAD KCOF /6201
3310 TAD SOURCE /CURRENT FIELD
3311 DCA SOURCE /SOURCE NOW * CDF N
3312 TAD KCOF /NEW FIELD
3313 TAD INSFLO /DESTN NOW * CDF N
3314 DCA DESTN
3315 TAD SOURCE
3316 CIA
3317 TAD DESTN
3320 SNA CLA
3321 JMP I XRTN
3322 DCA LOOP
3323 SOURCE, 0
3324 TAD I LOOP
3325 DCA SAVGD
3326 TAD SAVGD
3327 DESTN, 0
3330 DCA I LOOP
3331 TAD I LOOP
3332 CIA
3333 TAD SAVGD
3334 SNA CLA
3335 JMP I +3
3336 HLT
3337 JMP SOURCE
3340 ISZ LOOP
3341 JMP SOURCE
3342 TAD KCIF

```

```

/NOW * 2 FIELDS LOWER
/CURRENT FIELD
/ARE THEY EQUAL
/NO
/YES
/DOES IT * FIELD 0
/NO
/VERY LAST TO TEST
/MAKE IT NEXT FIELD
/CLEAR BIT 11
/4K COUNTER
/WILL * CDF N
/TAKE FROM HERE
/SAVE INSTRUCTION
/GET IT BACK
/PUT IN HERE
/GET INFORMATION STORED
/COMPARE TO THIS VALUE
/WERE THEY THE SAME
/YES CONTINUE
/NO, RELOCATION ERROR
/TRY SAME AGAIN
/DONE 4K WHEN SKIP
/KEEP MOVING
/6202

```

PAL10 V141 2-NOV-71 0119 PAGE 1-32
3343 1016 TAD INSFLD /NEW FIELD
3344 3345 DCA :+1
3345 6202 CIF 00 /CHANGE TO NEW FIELD
3346 5530 JMP I XRTN /EXIT TO RTN1 IN
3347 0000 SAVGD, 0 /NEW FIELD
\$

4000
4100
4200
4300
4400
4500
4600
4700
5000
5100
5200
5300
5400
5500
5600
5700
6000
6100
6200
6300
6400
6500
6600
6700
7000
7100
7200
7300
7400
7500
7600
7700

ALAW	2220	FLCNT	0073	LEGAL	2314	RD4	1455
ALOK	2273	FLOD	2444	LOCAT	2400	RDALL	0736
BAD	2153	FLOAD	0017	LOOP	0074	ROF	6214
BADD	2634	GENRAN	2716	M10	0141	ROLOP	0753
BEGIN	3200	GOOD	2555	M14	0140	RIF	6224
BSPCE	2600	GOOD	2164	M20	0056	RLOPA	1011
CBANK	1607	HALT	2641	M4	0060	RSTRT1	0211
CDF	6201	INSFLD	0016	M40	0057	RTRN1	0213
CDDN1	0760	JMS1	0110	M5	0061	RXT	1020
CFLD	1722	JMS2	0111	MCHA	0033	SAVGD	3347
CHAR	0134	JMS3	0112	MEMADR	0122	SAVIF	1623
CHNXT	3203	JMS4	0113	MOVE	3307	SETSR	2645
CIF	6202	JMS5	0114	MVBK	3302	SETU1	2200
CKBNK	1600	K1	0150	NXLOC	0151	SLMTS	2207
CKERR	3031	K10	0034	NXTBNC	1624	SOURCE	3323
CKNT	3242	K100	0040	NXTHI	3275	SPING	2115
CKNXT	3211	K177	2756	NXTMV	3200	STALL	2702
CLOP1	1252	K1K	1036	OCADR	2195	STMV	3110
CLOP2	1327	K20	0042	OKAS	2256	STNX	3247
CLOP3	1405	K200	0037	OTLDR	2541	STNXT	3225
CLOP4	1462	K207	2167	OVER	1706	STR	2670
CMOVE	3000	K212	2171	PBAD	2605	SUB1	3066
CNXT	3037	K215	0143	PFILO	2500	SUB2	3255
COUNT	0072	K240	2170	PGOOD	2524	SW0	2111
CRLF	2194	K254	2353	PHDR	2474	SW1	2042
DATELD	0014	K260	0142	PLIMT	2452	SW2	2033
DESTN	3327	K261	0092	PLOCT	2412	TBLRAN	2755
ENTBL	0021	K262	0033	POCDR	2512	TDM20	0106
EPRNT	2054	K263	0034	POSITN	2131	TDM40	0107
EQUAL	3053	K264	0035	PRERR	2146	TLMY	2446
ERROR	2000	K277	0146	PRGAM	2417	TNUM	0062
ERTBL	0020	K370	0145	PROCTL	2126	TST1	0300
ERWRD	0022	K40	0041	PSTSR	2692	TST10	0324
EXAM1	0241	K400	0036	PTSTN	2617	TST2	0400
EXAM2	0246	K7	0133	QUERY	2344	TST20	0425
EXAM3	0253	K740	0035	RANCON	2741	TST3	0452
EXAM4	0260	KCDF	0125	RANDEX	2742	TST30	0477
EXHDR	2563	KCIF	0126	RANTAB	2754	TST4	0600
EXIT	0153	KEYIN	2303	RANTAD	2734	TST4C	0625
EXT1	0322	KRXT	1035	RANTBL	2743	TSTL	2457
EXT10	0343	KXT1	0345	RCHK1	1037	TSTN	2626
EXT2	0423	KXT10	0346	RCHK2	1054	W0011	0652
EXT20	0445	KXT2	0450	RCHK2C	1071	W0110	0704
EXT3	0475	KXT2C	0451	RCHK3	1106	W1001	0721
EXT3C	0517	KXT3	0522	RCHK3C	1123	W1100	0667
EXT4	0623	KXT3C	0523	RCHK4	1140	WRLOP	1710
EXT4C	0647	KXT4	0645	RCHK4C	1200	WPCR	2660
FEILD	1646	KXT4C	0646	RCHKA	1217	X0011	0064
FILD	2532	LAST	2166	RD1	1000	X0110	0066
FIRST1	0123	LAST1	0124	RD2	1245	X1001	0067
FLAGS	0015	LASTX	0154	RD3	1322	X1100	0065
					1400	XBANK	0063

XCFL	1244
XCHK1	0076
XCHK1C	0077
XCHK2	0100
XCHK2C	0101
XCHK3	0102
XCHK3C	0103
XCHK4	0104
XCHK4C	0105
XCHKA	1236
XCRLF	0147
XFILD	0131
XHDR	0135
XHLT	0127
XITI	0320
XI11C	0341
XI12	0421
XI12C	0443
XI13	0473
XI13C	0515
XI14	0621
XI14C	0643
XKBNK	0070
XLGAT	2352
XLMTS	0043
XLOPA	1237
XLOPB	1240
XLOPC	1241
XLOPD	1242
XLOPE	1243
XMOVE	0050
XMVE	3124
XPERR	0136
XPHDR	2644
XPING	0137
XPRER	0132
XRALL	0075
XRD1	0115
XRD2	0116
XRD3	0117
XRD4	0120
XRROR	0121
XRTN	0130
XSALL	0152
XSETU	0051
XSRCE	3123
XSTSR	0277
XTBNK	0071
XTLIM	2351
XTMV	3125
XTST1	0044
XTST2	0045

XTST3	0046
XTST4	0047

ERRORS DETECTED: 0

LINKS GENERATED: 0

RUN-TIME: 13 SECONDS

2K CORE USED