PDP-15
MEMORY

IDENTIFICATION

Product Code:	MAINDEC-15-DAMXA-A-D
Product Name:	PDP-15 Extended Memory Checkerboard (MXC15B)
Date Produced:	9 April 1972
Maintainer:	Diagnostic Group
Authors:	John W Richardson D K Macomber J M Graetz

The information in this document is subject to change without notice and should not be construed as a commitment by Digital Equipment Corporation.

Actual distribution of the software described in this specification will be subject to terms and conditions to be announced at some future date by Digital Equipment Corporation.

1, ABSTRACT

This program is a modified version of MAINDEC-15-D1BC(MXCH15). It is designed to provide a worst-case testing pattern for both 950 nanosecond and 800-nanosecond memory stacks. The program tests for switching failures in individual cores under worst-case half-select noise conditions. It also tests for the switching of unstable cores affected by repeated half- select currents.

Operation of MXC15B is almost identical to that of MXCH15. Only the ACS test selectors and the halt addresses are different.

2, REQUIREMENTS

2,1 Equipment

Standard PDP-15 with a minimum of 8192 words of core memory.

2,2 Program Storage

The program occupies the first 3400(8) words of the memory field in which it is operating.

3, LOADING PROCEDURE

The program is punched in ABS mode, with HRI loader.

3,1 Normally, the program is loaded into memory field 00.

a. Place the tape in the reader.
b. Set the Address Switches to 17700.
c. Press RESET, then READ IN.

When the tape has been read, the computer stops.

3,2 The program can be loaded into the lower half of any 8K memory bank, by placing the machine in BANK MODE, and setting Address Switches 3-4 to the desired bank number.

4. STARTING PROCEDURE

4,1 Starting Address: 00200. Set the Address Switches to 00200, and press START.

4,2 Restarting. To re-establish the initial conditions, start the program at address f0200, where f is the current memory bank in which the program is located.

To retain current program conditions, restart in location f0215.

5. OPERATING PROCEDURES

A memory field is one 4096-word segment. Fields are numbered sequentially from 00 through 37(8). Fields 00-07 are in memory block 0, fields 10-17 in block 1, and so on.

The operator communicates with the program via the AC Switches or the Teletype keyboard.

Typing errors can be eliminated by striking RUBOUT. The program will delete typed input for that line, and request a new entry.

5,1 Initial Operator Action

5,1,1 Test Limits. After starting at location 00200, the program types

TEST LIMITS

The area to be tested is specified by typing the octal numbers of the limiting memory fields. The

two numbers are separated by a comma, and followed by a carriage return. Any two fields, including that which contains the program, may be selected as limits, and either limit may be typed first. Examples:

```
TEST LIMITS
00,05
```

Memory fields 00 through 05, inclusive (addresses 000000 through 057777), will be tested. Note that two digits must be typed for each field number.

A single field may be chosen:

```
TEST LIMITS
03,03
```

If, however, the single field is that containing the program, the choice will be rejected:

```
TEST LIMITS
00,00
PROGRAM IS IN FIELD 00
TEST LIMITS
```

The program will proceed when acceptable limits have been chosen.

If only a single digit is used to specify a field, or if a field number greater than 37(8) is typed, the limits will not be accepted. An example of both errors:

```
TEST LIMITS
1,42
?
TEST LIMITS
```

CAUTION: Do not select more memory than is actually available, e.g., typing an upper limit of 17 when only fields 00-07 are provided. The action of the program is unpredictable when trying to address nonexistent memory locations.

5.1.2 Set up ACS. After the test limits have been accepted, the program types

SETUP ACS

Set the AC Switches as desired (see Table)! for normal operation, all the switches are set to Ø.

Strike any key on the keyboard. The message

PROGRAM IS IN FIELD ØØ

is printed, and the test begins. It will run continuously until stopped by the operator.

5.1.3 Table of AC Switch Settings

ACS	state	Function	Described In Section
0	0	Continuous operation (normal)	5.2.2
	1	Stop after testing current field or after error printout	
1	0	Print error data and continue (normal)	5.2.2
	1	Halt before error printout	
2	0	Proceed as selected by ACS1 (normal)	5.2.3
	1	Ring bell on error; no printout; ACS 1 is ignored.	
3-5	000	Do all three tests in succession (normal)	5.1.4
	---	Any switch set to 1 selects that test	
9	0	Automatic program relocation (normal)	5.3.3
	1	Do not relocate program.	
11	0	Print "PROGRAM IS IN FIELD ff" after each relocation (normal)	5.3.1
	1	Do not print message	
12	0	Normal program relocation sequence	5.3.4
	1	Force relocation to specified field	
13	0	Normal test sequence	5.4
	1	Request scope loop	
14	0	Test entire word (normal)	5.5
	1	Suppress printout for selected bits	

5.1.4 Test Selection: ACS 3-5. Three tests are provided. Tests 1 and 2 use storage patterns that establish worst-case noise conditions on the sense wire. Test 3 checks for switching failures during repeated half-selecting. The execution of the three tests is controlled by AC switches 3, 4, and

51

Test 1 (ACS 3), Worst-case pattern for 800-nsec memory stacks,
Test 2 (ACS 4), Worst-case pattern for 950-nsec stacks,
Test 3 (ACS 5), Switching failures in cores subject to repeated half-select currents;

If all three switches are set to 0, the three tests are done in succession over the entire test area, If any switch is set to 1, the test specified by that switch is performed,

5,2 Program Operation

5,2,1 In normal operation, the tests are performed in order; 1, 2, and 3, A description is given in Section 9,

Tests 1 and 2 are identical except for the pattern written in memory, The pattern is written in all fields, then each word, starting at the lowest address, is tested for failures, When all locations have been tested, the complement of the pattern is written and tested,

In test 3, the entire test area is filled with 1s, A single word is cleared to 0s, and repeatedly addressed, Other words affected by the half-select currents are then tested for switching failures,

After the selected tests have been executed, the program is relocated (see Section 5,3) and the tests are repeated,

5,2,2 Program-controlled halts: ACS 0-1,

To insure program continuity, the console STOP switch should not be used to halt the program, AC Switches 0 and 1 provide program-controlled stopping points, at which the operator may change the settings of the other AC Switches, The changes take effect immediately after resuming operation by pressing CONTINUE,

ACS 0 0: The program runs continuously.
1: The program will stop at either of two points:
a. After an error printout, Contents of bits 6-17 of the MQ register are 0734.
b. After testing has been completed in the current field, C(MQ(6-17)) = 0642.

In both cases, the ACS may be changed. Press CONT to proceed. If the ACS settings have been changed, the test then being executed is terminated, and the new switch settings take effect. If the ACS have not been changed, the test resumes from the point at which it was stopped.

ACS 1 0: Error data are printed as usual.
1: The program halts before the error printout. C(MQ(6-17)) = 0730.

To proceed, press CONT. ACS changes are accepted immediately.

EXCEPTION: After error printouts have been inhibited (see Section 6.1.3), ACS 1 no longer has any effect, and ACS 0 will cause a halt only after a field has been tested (case b above).

5.2.3 Ringing the Error Bell: ACS 2.

If ACS 2 is set to 1, a bell rings at the occurrence of an error, the error printout is suppressed, and ACS 1 is ignored. The bell function is useful when an intermittent error is suspected: the program does not stop, and the frequency and/or regularity of the bell gives an index to the performance of the core stack.

Note, however, that ACS 0 remains in effect: the program will halt after the bell if ACS 0 is set to 1.

5.3 Program Relocation

5.3.1 Automatic Relocation. When all tests have been completed in the selected fields, the program is

relocated to the highest tested field in which no error was found during the preceding run. From this new location, the tests are repeated in the other selected fields. After each test run, the program is relocated in the next lower error-free field, until it reaches the bottom of the test area. From there, it goes back to the top. For example, assume fields 02-10 are chosen as test limits. After the first test run, the program relocates from field 00 to field 10. From there it will go to field 07, then 06, 05, 04, 03, and 02, skipping any field(s) in which an error was detected during the previous run. From field 02, the program will relocate to field 10 once more.

Immediately after each relocation, the message

PROGRAM IS IN FIELD ff

is printed. The message may be suppressed by setting AC Switch 11 to 1. The printout will resume when ACS 11 is set to 0.

5.3.2 Exceptions. The program will not relocate automatically under any of the following conditions:

a. Only one field has been selected for testing.

b. There are no error-free tested fields.

c. ACS 9 is set to 1 (see Section 5.3.4).

d. A forced relocation has been made (see Section 5.3.5).

In each of these cases, the program remains where it is, and the the test run is repeated.

5.3.3 Program protection. To protect against program failures, two internal checks are made. First, the program will not relocate to any field in which an error was detected during the previous test run. Second, when relocation does take place, the transfer is checked word by word with the source field; errors in transfer are tabulated and printed (see Section 6.2.1).

5.3.4 Relocation prevented: ACS 9. The operator can prevent automatic relocation by using ACS 9.

ACS 9 0: Relocation occurs normally.

1: Program does not relocate, but continues testing from its current location.

When ACS 9 is set to 0 again, automatic relocation will resume.

5.3.5 Forced Relocation: ACS 12. The operator can force the program to relocate to any specified field, by using ACS 12, as follows:

a. Set ACS 0 to 1, and wait for the program to halt.

b. Set ACS 0 to 0 and ACS 12 to 1. Press CONTINUE. The program types

PUT ACS 12 ON A 0

and waits for this to be done.

c. Set ACS 12 to 0. Immediately, the program types

GO TO FIELD

and waits for a field number.

d. Type a two-digit field number. The program relocates to the new field, and proceeds with the test.

After a forced relocation, the program will no longer relocate itself automatically. To restore the automatic function, restart the program from address 200, setting new test limits.

Forced relocation will not occur under either of the following conditions:

a. An error was detected in the destination field during the preceding test run. In this case, the message

ERROR IN SELECTED 4K
GO TO FIELD

is typed. Type a new field number, or a carriage Return to retain the program in the current field.

b. An error occurs during relocation. The error information is printed (see Section 6.2.1), followed by "GO TO FIELD". Proceed as described in case A above.

If the requested destination field is the only one that has been selected for testing, the program will relocate, and then restart at program address 200, with the message "TEST LIMITS", so that new limits can be chosen.

5.4 Scope Loops: ACS 13.

This program option allows the operator to examine signal traces on a scope under the conditions imposed by the three tests. The test functions are performed on one or more memory locations, but no error checking is done, no messages are printed, and there are no halts. For tests 1 and 2, any number of consecutive locations, from 1 to 4096, may be included in the loop. For test 3, only one register is used.

To initiate a scope loop, proceed as follows:

a. Halt the program by setting ACS 0 to 1.

b. Set ACS 13 to 1, and press CONTINUE. The message "TEST" is printed, and the program waits for a request.

c. Type the number of the desired test. An automatic CR follows, and the message

FIRST ADR

is printed.

d. Type the address of the first location of the group to be included in the loop; this must be a six-digit octal integer. The program responds with a CR and the message

LAST ADR

f. Type the address of the last location. This may be higher or lower than the first address.

The scope loop is begun as soon as the last address is typed. The loop will continue until the STOP switch is pressed, to request another loop, or to go back to the normal test program, restart at program location 200.

Examples of scope loop requests:

```
TEST 2
FIRST ADR 010100
LAST ADR 010100
```

The pattern for Test 2 is written throughout field 1, but the loop includes only location 100 in that field.

```
TEST 1
FIRST ADR 030102
LAST ADR 037777
```

The pattern for test 1 is written in field 03. The loop includes locations 0102 through 7777 in that field.

The area included in the loop may overlap memory fields, but the test pattern will be written only in the field containing the first address:

```
TEST 2
FIRST ADR 024000
LAST ADR 016770
```

The pattern will be written only in field 2, though the loop will include addresses 016770 through 024000. Note that the higher limit may be specified first.

For Test 3, only the first address is used, as this is a single register test. However, both addresses must be given. The example illustrates what happens when a selected address is in the program area:

```
TEST 3
FIRST ADR 002534
FIRST ADR IS WITHIN PROGRAM
PROGRAM IS IN FIELD 00
FIRST ADR 012534
LAST ADR 012534
```

If anything other than 1, 2, or 3 is typed for a test number, it will be rejected:

```
TEST 4
/?
TEST
```

5.4.1 Program loop for Tests 1 and 2. The loop performs the complement-recomplement operation on each of the selected registers in turn. The loop is 12 instructions long:

```
SCP1    EEM             /ENABLE EXTEND MODE
        LAC* MEMADR     /READ
        CMA             /COMPLEMENT DATA
        DAC* MEMADR     /WRITE
        LAC MEMADR      /ADDRESS
        SAD LTST        /COMPARE TO LAST
        JMP STSCP       /DONE
        ISZ MEMADR      /INCREMENT ADR
        JMP SCP1        /LOOP
STSCP   LAC ADRA        /FIRST ADR
        DAC MEMADR      /RESTORE COUNTER
        JMP SCP1        /GO TO TOP OF LOOP
```

5.4.2 Program loop for test 3. After writing a pattern of all 1s in the selected field, the test location is cleared to 0s, and repeatedly read. The loop is two instructions long:

```
LAC* ADRA
JMP .-1
```

5.5 Bit Suppression: ACS 14

Once a particular bit position (and hence its core plane) has demonstrated a consistent failure, error printouts due to this bit can be eliminated from subsequent passes. Proceed as follows:

a. Halt the program by setting ACS 0 to 1.

b', Set ACS 0 to 0 and ACS 14 to 1'. The message

SUPPRESS

is printed, and the program waits for input.

c', Set ACS 14 to 0'. Type the number of the bit position to be suppressed, as a decimal integer (0-17). If more than one bit is to be suppressed, separate the numbers by commas.

d', Strike the CR key'. The tests will start from the beginning again; the selected bits will be ignored for printouts.

Requests for bit suppression may be made at any time, and as frequently as is desired. Each succeeding request wipes out the effect of the previous one; to retain the suppression of a bit, its number must be typed again.

To remove all suppression requests, proceed as for a new request'. After the program types "SUPPRESS", type a Carriage Return.

Examples:

SUPPRESS 0 <CR>

SUPPRESS 4,9,13 <CR>

SUPPRESS <CR> [removes all suppression]

6. ERRORS

6.1 Test Data Errors

6.1.1 Data column header. Immediately after the first error is detected, the following header is printed, identifying the columns of error data:

```
TEST   OCTAL ADR   GOOD   BAD   PAT CONTROL WORD
```

TEST — The number---1, 2, or 3---of the test which detected the error.

OCTAL ADR — The octal address of the location containing the data in error

GOOD — What the contents should have been. This is always either 000000 or 777777.

BAD — The data as read. One or more bits will be in error. The datum is printed as an octal integer.

PAT CONTROL WORD — The word used to generate the checkerboard pattern for the current test. For test 1, this word is 776000 or (for the complement pattern) 003774. For test 2 it is 525250 or 252524. No control word is used for test 3.

Once printed, the header will not be printed again until the program is restarted from 200 or 215.

6.1.2 Data printout. Error data are printed as soon as an error has been detected. A typical printout might look like this:

```
TEST  OCTAL ADR  GOOD    BAD     PAT CONTROL WORD

1     014000     000000  000001  776000
1     060200     777777  776777  003774
2     014000     000000  000001  252524
3     037555     777777  377777
```

In the example, tests 1 and 2 detected a "picked up" bit in location 4000 of field 1, and test 1 found a "dropped" bit in location 0200 of field 6. Test 3 caught a switched core in location 7555 of field 3.

6.1.3 Printouts Inhibited. During long test runs, excessive error printouts would use too much time and paper. To avoid this, the program automatically suppresses test data printouts after 64 errors. The message

PRINTOUTS INHIBITED

is typed; the printouts will not be restored until the program is restarted at address 0200. Printouts for relocation errors are not affected.

6.2 Other errors

6.2.1 Program relocation errors. As errors are detected during the relocation of the program from one field to the next, the data are printed. Column heads are the same as for data printouts. Example:

```
TEST  OCTAL ADR  GOOD    BAD     PAT CONTROL WORD

      031000     741000  740000
      031001     611005  601005
      031002     760027  760007

NO MORE ERRORS
```

The errors occurred while relocating to field 03. If this happens during an automatic relocation, the program will attempt to relocate to the next lower field, in this case 02, continuing until an error-free transfer is obtained. If the errors occur during a forced relocation, the program will return to the control routine after the printout, and type "GO TO FIELD", to request another destination.

6.2.2 Error in Selected Field. If, in a forced relocation, the destination field had produced an error during the previous test run, the message

```
ERROR IN SELECTED FIELD
GO TO FIELD
```

is printed. A new destination must be chosen.

6.2.3 Address within program (Scope loop request). If, in specifying limits for a scoping loop, the operator types an address that is within the area occupied by the program, the message

```
FIRST [or LAST] ADDRESS IS WITHIN PROGRAM
PROGRAM IS IN FIELD ff.
```

is typed, and the program repeats the request for whichever address is at fault.

7. RESTRICTIONS

7.1 Hardware. The test will not operate in a PDP-15 with less than 8K of core memory.

7.2 Program.

7.2.1 In selecting test limits (see Section 5.1.1), the operator should not specify more memory than is actually available. The program will behave unpredictably otherwise.

7.2.2 During normal test execution, the console STOP switch should not be used to halt the program for the purpose of resetting AC switches. ACS 2 and 1 provide program-controlled halts.

8. MISCELLANEOUS

8.1 Execution time.

All times are approximate.

8.1.1 950-nsec memory stack. Tests 1 and 2 take 4.8 seconds to complete one 4K field. Test 3 requires 2.4 seconds per field.

8.1.2 800-nsec stack. Tests 1 and 2 take 4 seconds per field. Test 3 takes 2 seconds per field.

9. DESCRIPTION

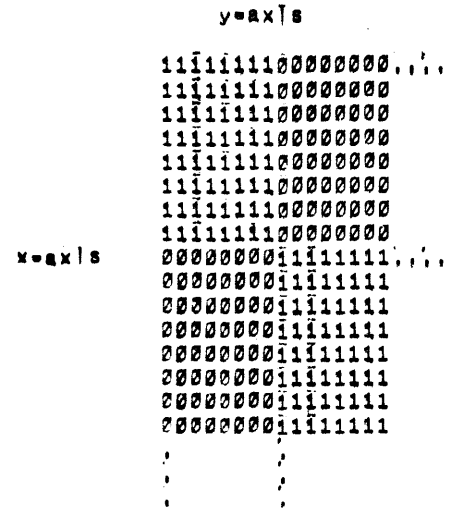
9.1 Tests 1 and 2.

These tests are identical except for the pattern of 1's and 0's stored in memory.

The principal fault for which a checkerboard program tests is that of a spurious "1" signal generated as the result of excessive "noise" current in a core array sense winding. The noise is a consequence of the path which the sense wire takes through the array. Each time the sense wire passes from a core in the 0 state through one in the 1 state, and vice versa (that is, each time there is a change in the magnetic field), a small voltage is induced in the wire. If the combination of these tiny bits of noise produces a voltage high enough to trigger the sense amplifier, a spurious "1" is detected.

The checkerboard pattern is designed to produce the maximum number of 0-1 and 1-0 junctions in the sense wire's path, thus generating the greatest amount of noise current. This "worst-case" pattern is different for core arrays with different winding paths. In the PDP-15, two different core stacks are used, requiring two worst-case patterns.

The pattern of Test 1 provides a worst case for the 800-nsec memory stack. It is shown below as it would appear in one portion of one core array (64x64 cores). The pattern shown repeats throughout the array.



In terms of core address, the array would be as follows:

Addresses	Contents (octal)
0000-0007	777777
0010-0017	000000
0020-0027	777777
.....	.....
0070-0077	000000
0100-0107	000000
0110-0117	777777
.....	.....

Test 2 uses a worst-case pattern for the 950-nsec memory stack. This pattern is shown below for a portion of one core array:

		y-axis
	0	10101010,,,,
	1	10101010
	2	10101010
	3	10101010
		,,,,,
x-axis	36	10101010
	37	10101010
	40	01010101,,,,
	41	01010101
	42	01010101
		,,,,
	76	01010101
	77	01010101,,,,

In terms of address:

Addresses	Contents
0000-0037	777777
0040-0077	000000
0100-0137	000000
0140-0177	777777
0200-0237	777777
0240-0277	000000
,,,,	,,,

Both tests proceed in this manner: The pattern is written throughout the entire test area. Then, starting with the lowest address, each memory location is tested in turn. The contents of the location are read into the AC, complemented, and deposited in the test location, then read, recomplemented, and deposited once more. The contents are read once more and compared with a control word; if any bits are in error, the information is printed, and the test proceeds to the next location. If no error is detected, the same location is tested further by performing the complement-recomplement operation on each bit of the contents in turn, starting with bit 17. As soon as an error is caught, the data are printed, and the test goes to the next location.

When all locations in the entire test area have been examined, the complement of the pattern is written, and the test is repeated.

9.2 Test 3

This test will catch switching failures in cores subjected to repeated half-select currents.

The entire memory field is filled with 1s. The contents of location 0000 are cleared to 0s, then read into the AC 1024 times in rapid succession. The contents of every other register half-selected by x-axis line 0 are then examined to see if any of the bits on that x-line have switched from 1 to 0. In like manner, the remaining 63 x-axis lines are tested. The test if performed in each field in turn.

```
/
/PDP-15 EXTENDED MEMORY CHECKERBOARD,
/8K MINIMUM CORE REQUIRED, S.A, = 200,
/
/COPYRIGHT 1972, DIGITAL EQUIPMENT CORP.,
/MAYNARD, MASS, 01754
/
/J. RICHARDSON
/D. MACOMBER
/J. M. GRAETZ
/
```

```
                                .TITLE MXC15B
                                .ABS
                        /
    20001                       .LOC    1
                        /
    00001   600001              JMP     1
    00002   777777              LAW     -1
    00003   777777              LAW     -1
    00004   777777              LAW     -1
    00005   777777              LAW     -1
                        /
            720406      TLS=720406
            720401      TSF=720401
            720301      KSF=720301
            720312      KRB=720312
            721000      PAX=721000
            707764      EBA=707764
            707762      EPA=707762
            707761      SBA=707761
            707741      EXBA=707741
            735000      CLX=735000
            707702      EEM=707702
            707704      LEM=707704
                        /
                                .EJECT
```

```
00200                       .LOC    200
                    /
00200   707752     BEGIN    EPA                 /ENTER PDP-15 MODE
00201   143121               DZM     FLAGS       /CLEAR PROGRAM FLAGS
00202   143176               DZM     PINX
00203   100646               JMS     WHERE       /SEE WHERE PROGRAM IS
00204   043122               DAC     INSFLD      /SAVE FIELD NO,
00205   101655               JMS     SLMTS       /SETUP TEST LIMITS
00206   102010               JMS     SETAC       /SETUP ACS
00207   777777               LAW     -1
00210   043170               DAC     BITSUP      /MASK FOR BIT SUPPRESSION
00211   777700               LAW     -100
00212   043247               DAC     MAXERR      /COUNT FOR TOTAL ERRORS
00213   777770               LAW     -10
00214   043117               DAC     SIXT4       /PASS COUNTER
00215   143120               DZM     NOPRNT
00216   201620     RTN1      LAC     GETAD-1
00217   041572               DAC     LOCAT+4
00220   707762               EPA
00221   101566               JMS     LOCAT       /TELL WHERE PROGRAM IS
00222   142616               DZM     PHDR
00223   203150               LAC     LAST1       /FIRST FIELD TO TEST
00224   543147               SAD     FIRST1      /LAST TO TEST
00225   741000               SKP
00226   600231               JMP     .+3
00227   543122               SAD     INSFLD      /INSFLD = FIELD WITH PROGRAM
00230   600202               JMP     BEGIN+2
00231   202674               LAC     ERTBL       /ERROR TABLE POINTER
00232   043250               DAC     ERWRD
00233   777740               LAW     -40
00234   043144               DAC     CT16
00235   760000               LAW
00236   063250               DAC*    ERWRD       /LAW = NO ERROR IN TABLE
00237   443250               ISZ     ERWRD
00240   443144               ISZ     CT16
00241   600236               JMP     .-3
                    /
00242   043123               DAC     LAST        /NO ERROR IN LAST
00243   100646               JMS     WHERE
00244   043122               DAC     INSFLD
00245   202674               LAC     ERTBL       /ERROR TABLE POINTER
00246   043250               DAC     ERWRD
                    /
                             .EJECT
```

```
                        /RETURN TO STOVER AFTER ANY ACS CHANGES WHILE RUNNING
                        /
00247   750004  STOVER  LAS                     /READ TEST PARAMETERS
00250   503243          AND     K177
00251   043124          DAC     MCWA
00252   503211          AND     K40
00253   744200          SZA!CLL                 /BIT 12 A 1 = FORCE RELOCATE
00254   602307          JMP     FCDMV           /RELOCATE
00255   750004          LAS
00256   503210          AND     K20
00257   740200          SZA                     /BIT 13 A 1 = KEYBOARD INPUT
00260   601034          JMP     KYBRD           /WAIT FOR INPUT
00261   750004          LAS
00262   503207          AND     K10
00263   740200          SZA                     /BIT 14 A 1 = BIT SUPPRESSION
00264   101410          JMS     SUPBIT
00265   203124          LAC     MCWA            /PARAMETERS
00266   503240          AND     K74K            /MASK BITS 3 TO 6
00267   741200          SNA                     /ALL 0 = DO ALL TESTS
00270   600317          JMP     DOALL
                        /
                        /EXAMINE TEST SWITCHES 3 TO 6
                        /
00271   203124  EXTST   LAC     MCWA
00272   503236          AND     K40K
00273   740200          SZA                     /BIT 3 A 1 = TEST 1
00274   600322          JMP     TST1
00275   203235  EXAM2   LAC     K20K
00276   503124          AND     MCWA
00277   740200          SZA                     /BIT 4 A 1 = TEST 2
00300   600336          JMP     TST2
00301   203233  EXAM3   LAC     K10K
00302   503124          AND     MCWA
00303   740200          SZA                     /BIT 5 A 1 = TEST 3
00304   600353          JMP     TST3
00305   443117  EXREL   ISZ     SIXT4           /64 PASSES IF SKIP
00306   600312          JMP     .-4
00307   143120          DZM     NOPRNT          /CLEAR NO PRINT FLAG
00310   777770          LAW     -10             /RESTORE COUNT
00311   043117          DAC     SIXT4
00312   750004          LAS
00313   503227          AND     K400
00314   740200          SZA                     /BIT 9 A 1 = DON'T MOVE
00315   600216          JMP     RTN1
00316   602125          JMP     CMOVE           /DONE ALL TESTS, SETUP
                                                /FOR RELOCATION
                        /
                        /SETUP TO RUN ALL TESTS
                        /
00317   203124  DOALL   LAC     MCWA
00320   243240          XOR     K74K            /SET ALL TEST BITS
00321   043124          DAC     MCWA            /RESTORE
                        /
                        .EJECT
```

```
                        /TEST 1, WRITE CHECKER PATTERN #1: SLOW MEMORY WORST CASE
                        /
00322   223127  TST1    LAC     PCWA            /TEST 1 PAT, CONTROL WORD
00323   343125          DAC     PCW
00324   343126          DAC     CNTRL
00325   760261          LAW     261             /ASCII 1
00326   343136          DAC     TNUM            /TEST NUMBER
00327   203206          LAC     K7              /SLOW MEMORY WORST-CASE CONSTANTS
00330   343202          DAC     XMSK
00331   143203          DZM     XCOM
00332   100357          JMS     NETWK           /GO WRITE IN ALL FIELDS
00333   100366          JMS     CREAD           /NOW GO READ AND TEST
00334   600275          JMP     EXAM2           /SEE IF TEST 2 WANTED
00335   600332          JMP     .-3             /DO COMPLEMENT
                        /
                        /TEST 2, WRITE CHECKER PATTERN #2: FAST MEMORY WORST-CASE
                        /
00336   223130  TST2    LAC     PCWB            /TEST 2 PAT, CONTROL WORD
00337   343125          DAC     PCW
00340   343126          DAC     CNTRL
00341   760262          LAW     262             /ASCII 2
00342   343136          DAC     TNUM            /TEST NUMBER
00343   203212          LAC     K77
00344   343202          DAC     XMSK
00345   203211          LAC     K40             /COMPLEMENT EVERY 40(8)X-LINES
00346   343203          DAC     XCOM
00347   100357          JMS     NETWK           /WRITE IN ALL FIELDS
00350   100366          JMS     CREAD           /READ AND TEST EACH FIELD
00351   600301          JMP     EXAM3           /SEE IF TEST 3 WANTED
00352   600347          JMP     .-3             /DO COMPLEMENT
                        /
                        /TEST 3, TEST ALL XY COORDINATES
                        /
00353   760263  TST3    LAW     263             /ASCII 4
00354   343136          DAC     TNUM            /TEST NUMBER
00355   100524          JMS     BURST           /WRITE IN ALL FIELDS
00356   600325          JMP     EXREL           /PREPARE TO RELOCATE
                        /
                        .EJECT
```

```
                        /ROUTINE TO SETUP ADDRESSES FOR WRITE LOOP
                        /
 00357  000000  NETWK   0
 00360  100604          JMS     SETU1           /SETUP 1ST FIELD TO TEST
 00361  100613          JMS     CBANK           /SEE IF IT HAS PROGRAM
 00362  741000          SKP                     /NO
 00363  620357          JMP*    NETWK           /EXIT
 00364  100417          JMS     WRITE           /ACTUALLY WRITE ONE FIELD
 00365  600637          JMP     NXTBNK          /SETUP FOR NEXT FIELD
                        /
                        /ROUTINE TO SETUP ADDRESSES FOR READ LOOP
                        /
 00366  000000  CREAD   0
 00367  100424          JMS     READ            /ACTUALLY READ AND TEST 1 FIELD
 00370  777774          LAW     -4
 00371  243125          XOR     PCW             /AC=COMPLEMENT OF PCW
 00372  543126          SAD     CNTRL           /ALL DONE IF EQUAL
 00373  620366          JMP*    CREAD           /EXIT
 00374  043126          DAC     CNTRL           /CNTRL=COMPLEMENT PATTERN
 00375  440366          ISZ     CREAD           /RETURN+1
 00376  620366          JMP*    CREAD           /EXIT AND WRITE COMPLEMENT
                        /
                        /PATTERN ROUTINE FOR TESTS 1 THRU 3
                        /
 00377  000000  GENPAT  0
 00400  203126          LAC     CNTRL           /CURRENT PATTERN CONTROL WORD
 00401  043135          DAC     PATN            /SAVE
 00402  777700          LAW     -100
 00403  043146          DAC     CT04            /COUNTS Y AXIS
 00404  203135          LAC     PATN            /CONTROL WORD
 00405  043132          DAC     PATR            /SAVE
 00406  777760  WCNT    LAW     -20
 00407  043144          DAC     CT16            /COUNTS 16 SHIFTS
 00410  203132          LAC     PATR
 00411  744010          RCL
 00412  043132          DAC     PATR
 00413  751402          SZL!CLA                 /NO SKIP SAYS WRITE 777777
 00414  740001  COMPL   CMA                     /AC = 7777
 00415  043153          DAC     GOOD1           /SAVE
 00416  620377          JMP*    GENPAT          /EXIT TO READ OR WRITE
                        /
                        /WRITE ROUTINE FOR TESTS 1 THRU 3
                        /
 00417  000000  WRITE   0
 00420  100377          JMS     GENPAT          /GET A WORD
 00421  050000          DAC     X               /WRITE
 00422  100471          JMS     CKXY            /CHECK FOR PATTERN INVERSION
 00423  620417          JMP*    WRITE           /DONE 4K
                        .EJECT
```

```
                        /
                        /READ AND TEST ROUTINE FOR TESTS 1 THRU 3
                        /
 22424  000000  READ    0
 22425  100604          JMS     SETU1           /SETUP FOR FIRST FIELD
 22426  100613          JMS     CBANK           /SEE IF IT HAS PROGRAM
 22427  741000          SKP                     /NO
 22430  620424          JMP*    READ            /NO MORE CORE TO READ
 22431  100377          JMS     GENPAT          /GET A WORD
 22432  210000          LAC     X               /TEST FULL-WORD COMPLEMENT
 22433  740001          CMA
 22434  050000          DAC     X
 22435  210000          LAC     X
 22436  740001          CMA                     /RECOMPLEMENT
 22437  050000          DAC     X
 22440  210000          LAC     X               /READ
 22441  543153          SAD     GOOD1           /IS WORD OK?
 22442  741000          SKP                     /YES, GO ON TO BITWISE TEST.
 22443  600466          JMP     ERSET           /NO.
 22444  203204          LAC     K1
 22445  043137          DAC     BITCON          /USED FOR BIT INVERSION
 22446  203137  RCOM    LAC     BITCON
 22447  250000          XOR     X               /COMPLEMENT A BIT
 22450  050000          DAC     X               /WRITE
 22451  203137          LAC     BITCON
 22452  250000          XOR     X               /RE-COMPLEMENT
 22453  050000          DAC     X               /RE-WRITE
 22454  210000          LAC     X               /READ
 22455  543153          SAD     GOOD1           /COMPARE
 22456  741000          SKP                     /OK
 22457  600466          JMP     ERSET           /PRINT INFO
 22460  203137          LAC     BITCON
 22461  744010          RCL                     /SETUP FOR NEXT BIT
 22462  740400          SNL                     /DONE 18 IF LINK = 1
 22463  600445          JMP     RCOM-1          /DO NEXT BIT POSITION
 22464  100471  CKAL    JMS     CKXY            /CHECK FOR PATTERN INVERSION
 22465  600637          JMP     NXTBNK          /SETUP FOR NEXT FIELD
 22466  043151  ERSET   DAC     BAD1
 22467  100666          JMS     ERROR           /PRINT INFO
 22470  600464          JMP     CKAL
                        /
                        .EJECT
```

```
                        /
                        /ROUTINE TO CHECK FOR PATTERN INVERSION
                        /
00471   000000  CKXY    0
00472   443143          ISZ     CT4K            /DONE 4K IF SKIP
00473   741000          SKP
00474   620471          JMP*    CKXY            /EXIT TO WRITE OR READ
00475   443146          ISZ     CT04            /DONE WITH Y AXIS IF SKP
00476   741000          SKP
00477   600506          JMP     Y64             /DONE 64 Y LINES
00500   724000          PXA
00501   343213  N64     TAD     K100            /INCREMENT Y ADDRESS BY 1
00502   721000          PAX
00503   443144          ISZ     CT16            /CHECK FOR 16 LOCATIONS
00504   620412          JMP     WCNT+2          /NOT YET
00505   600404          JMP     WCNT-2          /RESTORE COUNT
                        /
00506   737001  Y64     AXR+1                   /INCREMENT X LINE BY 1
00507   777700          LAW     -100
00510   043146          DAC     CT04            /RESTORE Y LINE COUNTER
00511   724000          PXA
00512   503202          AND     XMSK
00513   543203          SAD     XCOM            /COMPLEMENT PATTERN IF EQUAL
00514   600520          JMP     .+4
00515   724000          PXA
00516   343311          TAD     (770000
00517   600501          JMP     N64             /START WITH NEW X-Y COMBO
00520   203135          LAC     PATN            /PATTERN CONTROL WORD
00521   740001          CMA
00522   043135          DAC     PATN            /COMPLEMENTED CONTROL WORD
00523   600515          JMP     .-6
                        /
                        .EJECT
```

```
                        /TEST 3 WRITE AND READ ROUTINE
                        /
00524   000000  BURST   0
00525   100604          JMS     SETU1           /SETUP FOR FIRST FIELD
00526   100613          JMS     CBANK           /SEE IF IT HAS PROGRAM
00527   741000          SKP                     /NO
00530   600567          JMP     DOXY            /READ XY COORDINATES
00531   777777  WONS    LAW     -1
00532   073201          DAC*    MEMADR,X        /WRITE 1'S INTO ALL FIELDS
00533   443201          ISZ     MEMADR          /ADDRESS+1
00534   443143          ISZ     CT4K            /DONE 4K WHEN SKIP
00535   600531          JMP     WONS
00536   600637          JMP     NXTBNK          /SETUP FOR NEXT FIELD
00537   100604  DOXY    JMS     SETU1           /SETUP FOR FIRST FIELD
00540   100613          JMS     CBANK           /SEE IF IT HAS PROGRAM
00541   741000          SKP                     /NO
00542   620524          JMP*    BURST           /EXIT
00543   203201          LAC     MEMADR          /ADDRESS 0 OF FIELD X
00544   343213          TAD     K100            /ADD Y LINE 01 TO IT
00545   043132          DAC     PATR            /SAVE
00546   203132  BRSTA   LAC     PATR
00547   043143          DAC     CT4K            /Y LINE ADDRESS
00550   776000          LAW     -2000           /-1024 DECIMAL
00551   043144          DAC     CT16
00552   173201          DZM*    MEMADR,X        /CLEAR LINE XN
00553   233201          LAC*    MEMADR,X        /READ 000000 1024 TIMES TO
                                                /TRY TO SWITCH OTHER LINES
00554   443144          ISZ     CT16
00555   600553          JMP     .-2
00556   777777  BUST    LAW     -1
00557   273143          XOR*    CT4K,X          /LINE Y + X MUST = 777777
00560   741200          SNA                     /SHOULD NOT SKIP
00561   600567          JMP     CEND            /OK
00562   740001          CMA
00563   043151          DAC     BAD1            /SAVE BAD DATA
00564   200556          LAC     BUST
00565   043153          DAC     GOOD1           /SAVE GOOD DATA
00566   100666          JMS     ERROR           /PRINT INFO
00567   203213  CEND    LAC     K100
00570   343143          TAD     CT4K            /Y AXIS PLUS 1
00571   043143          DAC     CT4K
00572   503231          AND     K7700           /MASK Y ADDRESS
00573   740200          SZA                     /ALL DONE IF SKIP
00574   600556          JMP     BUST            /READ NEXT Y ON CURRENT X
00575   443201          ISZ     MEMADR          /INCREMENT X ADDRESS
00576   443132          ISZ     PATR            /INCREMENT X+Y ADDRESS
00577   203212          LAC     K77
00600   503201          AND     MEMADR
00601   740200          SZA                     /DONE 64 X LINES IF 0
00602   600546          JMP     BRSTA           /TEST NEW X WITH Y01 TO Y63,
00603   600637          JMP     NXTBNK          /SETUP FOR NEXT FIELD
                        /
                        .EJECT
```

```
                        /
                        /SETUP FOR FIRST 4K FIELD
                        /
00604   000000  SETU1   0
00605   203147          LAC     FIRST1          /FIRST TO TEST
00606   721000          PAX                     /ADDRESS COUNTER
00607   043140          DAC     SVADR
00610   770000          LAW     -10000
00611   043143          DAC     CT4K            /4K COUNTER
00612   620604          JMP*    SETU1           /EXIT
                        /
                        /ROUTINE TO SEE IF TESTED FIELD HAS PROGRAM
                        /
00613   000000  CBANK   0
00614   100646          JMS     WHERE           /CURRENT PROGRAM FIELD
00615   722000          PAL
00616   543140          SAD     SVADR           /NEXT TO TEST
00617   600637          JMP     NXTBNK          /SEE IF CURRENT IS LAST
00620   740031          CMA!IAC
00621   343140          TAD     SVADR
00622   721000          PAX
00623   730000          PLA
00624   043201          DAC     MEMADR
00625   620613          JMP*    CBANK           /EXIT
00626   440613  NOMOR   ISZ     CBANK           /RETURN +1
00627   620613          JMP*    CBANK
00630   203140          LAC     SVADR
00631   343233          TAD     K10K            /CURRENT +4K
00632   043140          DAC     SVADR           /NEW FIELD
00633   721000          PAX
00634   770000          LAW     -10000          /-4K
00635   043143          DAC     CT4K            /4K COUNTER
00636   620614          JMP     CBANK+1         /EXIT AND TEST NEW FIELD
                        /
                        /ROUTINE TO CHECK FOR LAST FIELD
                        /
00637   750004  NXTBNK  LAS                     /CHECK ACS0 FOR HALT
00640   741100          SPA
00641   100653          JMS     HALT            /GO HALT
00642   203140          LAC     SVADR
00643   543150          SAD     LAST1           /ALL DONE IF EQUAL
00644   600626          JMP     NOMOR
00645   600630          JMP     NOMOR+2
                        /
                        /ROUTINE TO DETERMINE WHERE PROGRAM IS
                        /
00646   000000  WHERE   0                       /CONTAINS EPC
00647   200646          LAC     .-1
00650   503237          AND     K70K            /CLEAR ALL BUT BITS 3,4,5
00651   243176          XOR     PINX
00652   620646          JMP*    WHERE           /EXIT
                        /
                        .EJECT
```

```
                        /HALT ROUTINE.  PRESS CONTINUE TO RESUME
                        /TESTING, OR IF ACS CHANGES, TO EXECUTE
                        /NEW PARAMETERS.
                        /
020653  000000  HALT    0
020654  740040          HLT                     /PRESS CONTINUE
020655  750004          LAS
020656  740010          RAL
020657  741100          SPA
020660  620653          JMP*    HALT
020661  740020          RAR
020662  503243          AND     K177
020663  543124          SAD     MQWA
020664  620653          JMP*    HALT            /RESUME WHERE LEFT OFF
020665  600247          JMP     STOVER          /EXECUTE NEW PARAMETERS
                        /
                        /ERROR PRINT-OUT ROUTINE,  PLACE ACS0 UP FOR
                        /HALT AFTER PRINT-OUT,  PRESS CONTINUE TO GO ON.
                        /
020666  000000  ERROR   0
020667  724000          PXA                     /SAVE BAD DATA
020670  503232          AND     K7777
020671  243140          XOR     SVADR
020672  043152          DAC     OCADR           /SAVE FAILING ADDRESS
020673  203250          LAC     ERWRD           /ERROR TABLE POINTER
020674  542675          SAD     ENERR           /LAST ADDRESS OF TABLE
020675  741000          SKP
020676  600702          JMP     .+4
020677  202674          LAC     ERTBL           /FIRST ADDRESS OF TABLE
020700  043250          DAC     ERWRD           /PUT POINTER TO TOP OF TABLE
020701  600711          JMP     SW2             /CHECK AC2 FOR BELL
020702  203152          LAC     OCADR           /FAILING ADDRESS
020703  503244          AND     K370K           /MASK 3, 4 AND 5
020704  543123          SAD     LAST            /NEW ERROR FIELD IF SKIP
020705  600711          JMP     .+4             /SAME FIELD AS LAST ERROR
020706  043123          DAC     LAST
020707  063250          DAC*    ERWRD           /STORE FIELD# IN TABLE
020710  443250          ISZ     ERWRD           /INCREMENT POINTER
                        /
                        .EJECT
```

```
00711  760000  SW2   LAW              /PRINT INHIBIT IF = LAW
00712  543120        SAD    NOPRNT
00713  620666        JMP*   ERROR     /DON'T PRINT
00714  750004        LAS
00715  742010        RTL
00716  740100        SMA              /BELL IF SKIP
00717  600723        JMP    SW1       /CHECK ACS 1
00720  760207        LAW    207       /ASCII BELL
00721  102027        JMS    PCHAR     /PRINT
00722  600731        JMP    SW0       /CHECK ACS 0 FOR HALT
00723  750004  SW1   LAS
00724  740010        RAL
00725  740100        SMA              /NO SKIP = PRINT INFO
00726  600735        JMP    DOERR     /PRINT
00727  100653        JMS    HALT      /HALT
00730  600735        JMP    DOERR     /PRINT INFO
00731  750004  SW0   LAS
00732  741100        SPA              /NO SKIP = HALT
00733  100653        JMS    HALT
00734  620666        JMP*   ERROR     /RETURN TO READ ROUTINE
                /
                      .EJECT
```

```
                        /SETUP TO PRINT ERROR
                        /
00735   203151  DOERR   LAC     BAD1            /BAD DATA
00736   741200          SNA
00737   600752          JMP     STER-6          /FULL WORD ERROR
00740   740001          CMA
00741   741200          SNA
00742   600752          JMP     STER-6          /FULL WORD ERROR
00743   740001          CMA
00744   503170          AND     BITSUP          /MASK SUPPRESSED BITS
00745   740200          SZA
00746   740001          CMA
00747   503170          AND     BITSUP
00750   741200          SNA                     /NEW ERROR IF SKIP
00751   620666          JMP*    ERROR           /ERROR IS SUPPRESSED
00752   202616          LAC     PHDR
00753   741200          SNA                     /PRINT HEADER IF 0
00754   102616          JMS     PHDR
00755   203136          LAC     TNUM            /ASCII TEST NUMBER
00756   102027          JMS     PCHAR           /PRINT TEST NO.
00757   777767          LAW     -11             /-9
00760   043142  STER    DAC     CT32            /USED FOR SPACING COUNT
00761   102101          JMS     SPING           /SPACE 9
00762   203152          LAC     OCADR           /OCTAL ADDRESS
00763   042072          DAC     CRLF            /SAVE TEMPORARILY
00764   102107          JMS     PROCTL          /PRINT FAILING ADDRESS
00765   777772          LAW     -6
00766   043142          DAC     CT32
00767   102101          JMS     SPING           /SPACE 7
00770   203153          LAC     GOOD1           /WHAT DATA SHOULD BE
00771   042072          DAC     CRLF
00772   102107          JMS     PROCTL          /PRINT THE GOOD
00773   777776          LAW     -2
00774   043142          DAC     CT32
00775   102101          JMS     SPING           /SPACE 5
00776   203151          LAC     BAD1            /DATA READ
00777   042072          DAC     CRLF            /SAVE
01000   102107          JMS     PROCTL          /PRINT THE BAD
01001   760264          LAW     264
01002   543136          SAD     TNUM
01003   601012          JMP     INDY+4
                        /
                        .EJECT
```

```
21224   777773          LAW     -5
21225   043142          DAC     CT32
21226   102101  INDY    JMS     SPING           /SPACE 5
21227   203126          LAC     CNTRL           /CURRENT CONTROL WORD
21210   042072          DAC     CRLF            /SAVE
21211   102107          JMS     PROCTL          /PRINT PATTERN CONTROL WORD
21212   102072          JMS     CRLF            /CR, LF
21213   443247          ISZ     MAXERR          /CHECK FOR MAX, PRINT-OUTS
21214   620731          JMP     SW0             /CHECK ACS 0
21215   777700          LAW     -100            /-64 DECIMAL
21216   043247          DAC     MAXERR
21217   760000          LAW
21220   043120          DAC     NOPRNT          /NO MORE ERROR PRINT-OUTS
                                                /UNTIL RESTART FROM 100
21221   202676          LAC     PTO
21222   043154          DAC     PRNT
21223   102034          JMS     PNXT            /PRINT-OUTS INHIBITED
21224   102072          JMS     CRLF            /CR,LF
21225   777766          LAW     -12             /-10 DECIMAL
21226   043142          DAC     CT32
21227   760212          LAW     212             /LF
21230   102027          JMS     PCHAR
21231   443142          ISZ     CT32            /10 LINE FEEDS
21232   621027          JMP     .-3
21233   620731          JMP     SW0
                /
                        .EJECT
```

```
                        /
                        /KEYBOARD INPUT ROUTINES
                        /
01234   735000  KYBRD   CLX
                        /
                        /TYPE TEST# AND WAIT FOR INPUT,  CARRIAGE
                        /RETURN ONLY MEANS USE LAST PATTERN WRITTEN
                        /
01235   202653  TSTNO   LAC     TSNX            /POINTER FOR TEST#
01236   043154          DAC     PRNT
01237   102072          JMS     CRLF            /CR, LF
01240   102034          JMS     PNXT            /PRINT TEST#
01241   101544          JMS     KEYIN           /WAIT FOR INPUT
01242   543226          SAD     K377            /START OVER IF A RUB-OUT
01243   601034          JMP     KYBRD
01244   543216          SAD     K215            /CHECK FOR C,R,
01245   101067          JMS     ADR1            /USE LAST PATTERN WRITTEN
01246   777517          LAW     -261            /-1 ASCII
01247   343142          TAD     CT32
01250   740100          SMA                     /MINUS = INPUT <1
01251   601054          JMP     .+3
01252   102607          JMS     WOTIS           /PRINT QUERY AND RESTART
01253   601035          JMP     TSTNO
01254   203142          LAC     CT32            /TEST# IN ASCII
01255   740001          CMA
01256   343204          TAD     K1
01257   343221          TAD     K263            /SUBTRACT ASCII 3
01260   740100          SMA                     /MINUS = TEST# >3
01261   601064          JMP     .+3
01262   102607          JMS     WOTIS           /PRINT QUERY AND RESTART
01263   601035          JMP     TSTNO
01264   760000  TSTN    LAW
01265   243142          XOR     CT32
01266   043136          DAC     TNUM            /NEW TEST NUMBER
                        /
                        .EJECT
```

```
                        /XCH15 - TAPE 2
                        /
                        /WAIT FOR FIRST 6 DIGIT ADDRESS TO LOOP ON
                        /
01067   202700  ADR1    LAC     ROTB            /POINTER FOR LAW-XX
01070   042677          DAC     ROTA
01071   222677          LAC*    ROTA
01072   043162          DAC     NROTA           /LEFT SHIFT COUNTER
01073   760000          LAW
01074   043160          DAC     ADRA
01075   143155          DZM     ADRCW           /SAVES PARTIAL ADDRESS
01076   102072          JMS     CRLF            /CR, LF
01077   202654          LAC     ADRX            /POINTER FOR FIRST ADR,
01100   043154          DAC     PRNT
01101   102034          JMS     PNXT            /PRINT FIRST ADR,
01102   202655          LAC     AD1R            /C(AD1R) = ADR1
01103   043157          DAC     OVER
01104   202656          LAC     DON1            /S(DON1) = DFST
01105   043156          DAC     EXIT
01106   101544  FADR    JMS     KEYIN           /WAIT FOR INPUT
01107   101553          JMS     LEGAL           /SEE IF VALID
01110   203142          LAC     CT32            /ASCII INPUT
01111   503206          AND     K7
01112   043142          DAC     CT32
01113   101621          JMS     GETAD           /SHIFT LEFT TO FROM ADDRESS
01114   601106          JMP     FADR            /GET NEXT DIGIT
01115   203155          LAC     ADRCW
01116   043160          DAC     ADRA
01117   202700  DFST    LAC     ROTB            /POINTER FOR LAW TABLE
01120   542677          SAD     ROTA            /NOT EQUAL = <6 CHARACTERS
01121   601124          JMP     .+3             /O,K,
01122   042677          DAC     ROTA
01123   601564          JMP     QUERY           /PRINT QUERY AND RESTART
01124   101373          JMS     PROG            /NOW SEE IF 1ST ADR, IS
                                                /IN SAME 4K AS PROGRAM
01125   601143          JMP     ADR2            /OK
01126   102072          JMS     CRLF            /CR, LF
01127   202664          LAC     ADR1P
01130   043154          DAC     PRNT
01131   102034          JMS     PNXT            /PRINT "FIRST"
01132   202666          LAC     OVRLP
01133   043154          DAC     PRNT
01134   102034          JMS     PNXT            /PRINT "ADR, IS WITHIN PROGRAM"
01135   760000          LAW
01136   041572          DAC     LOCAT+4
01137   101566          JMS     LOCAT           /TELL WHERE PROGRAM IS
01140   201620          LAC     GETAD-1
01141   041572          DAC     LOCAT+4
01142   601067          JMP     ADR1            /START OVER
                        /
                        .EJECT
```

```
                        /
                        /WAIT FOR LAST 6 DIGIT ADDRESS OF BLOCK
                        /
01143   760020  ADR2    LAW
01144   043161          DAC     ADRB
01145   143155          DZM     ADRCW
01146   222677          LAC*    ROTA
01147   043162          DAC     NROTA           /FIRST COUNT FOR LEFT SHIFT
01150   102072          JMS     CRLF            /CR, LF
01151   202657          LAC     ADXR            /POINTER FOR LAST ADR,
01152   043154          DAC     PRNT
01153   102034          JMS     PNXT            /PRINT LAST ADR,
01154   202660          LAC     AD2R            /C(AD2R) = ADR2
01155   043157          DAC     OVER
01156   202661          LAC     DON2            /C(DON2) = DLST
01157   043156          DAC     EXIT
01160   101544  LADR    JMS     KEYIN           /WAIT FOR INPUT
01161   101553          JMS     LEGAL           /SEE IF VALID
01162   203142          LAC     CT32            /ASCII INPUT
01163   503206          AND     K7
01164   043142          DAC     CT32
01165   101621          JMS     GETAD           /SHIFT LEFT TO FORM ADDRESS
01166   601160          JMP     LADR            /GET NEXT DIGIT
01167   203155          LAC     ADRCW
01170   043161          DAC     ADRB
01171   202700  DLST    LAC     ROTB
01172   542677          SAD     ROTA            /NOT EQUAL = <6 CHARACTERS
01173   601176          JMP     .+3             /O.K.
01174   042677          DAC     ROTA
01175   601564          JMP     QUERY           /PRINT QUERY AND RESTART
01176   101373          JMS     PROG            /SEE IF LAST ADDRESS IS IN
                                                /SAME 4K AS PROGRAM
01177   601215          JMP     STLP            /O.K.
01200   102072          JMS     CRLF            /CR, LF
01201   202665          LAC     ADR2P
01202   043154          DAC     PRNT
01203   102034          JMS     PNXT            /PRINT "LAST"
01204   202666          LAC     OVRLP
01205   043154          DAC     PRNT
01206   102034          JMS     PNXT            /PRINT "ADR, IS WITHIN PROGRAM"
01207   760000          LAW
01210   041572          DAC     LOCAT+4
01211   101566          JMS     LOCAT           /TELL WHERE PROGRAM IS
01212   201620          LAC     GETAD-1
01213   041572          DAC     LOCAT+4
01214   601143          JMP     ADR2            /START OVER
                        /
                        .EJECT
```

```
                        /
                        /SETUP ADDRESSES AND PATTERNS BEFORE LOOPING.
                        /
01215   760000  STLP    LAW
01216   543160          SAD     ADRA            /A LAW = NO 1ST ADDRESS
01217   741000          SKP
01220   601227          JMP     CKLST
01221   543161          SAD     ADRB            /A LAW = NO LAST ADDRESS
01222   600216          JMP     RTN1            /RESTART PROGRAM
01223   203161          LAC     ADRB
01224   043160          DAC     ADRA            /ONLY ONE SELECTED
01225   043163          DAC     LTST            /LAST OF BLOCK
01226   601271          JMP     SIMU
                        /
01227   543161  CKLST   SAD     ADRB            /A LAW = NO LAST ADDRESS
01230   741000          SKP
01231   601236          JMP     CBOTH
01232   203160          LAC     ADRA            /ONLY 1 ADDRESS WANTED
01233   043161          DAC     ADRB
01234   043163          DAC     LTST            /LAST OF BLOCK
01235   601271          JMP     SIMU
                        /
01236   203160  CBOTH   LAC     ADRA            /FIRST ADR,
01237   503237          AND     K70K            /MASK BITS 3,4 AND 5
01240   043142          DAC     CT32            /SAVE
01241   203161          LAC     ADRB            /LAST ADR,
01242   503237          AND     K70K            /MASK 3,4 AND 5
01243   543142          SAD     CT32            /BOTH MUST = SAME 4K
01244   601247          JMP     .+3             /OK
01245   102607          JMS     WOTIS           /PRINT QUERY
01246   601057          JMP     ADR1            /START OVER
                        /
01247   203160          LAC     ADRA            /FIRST ADDRESS
01250   740021          CMA
01251   343204          TAD     K1              /2'S COMPLEMENT
01252   343161          TAD     ADRB            /SUBTRACT LAST ADDRESS
01253   740100          SMA                     /FIRST IS > LAST IF NEG,
01254   601265          JMP     SIMU+4          /LEAVE AS IS
01255   203161          LAC     ADRB
01256   042072          DAC     CRLF
01257   203160          LAC     ADRA
01260   043161          DAC     ADRB            /FIRST IS NOW LAST
01261   043163          DAC     LTST
01262   202072          LAC     CRLF
01263   043160          DAC     ADRA            /LAST IS NOW FIRST
01264   601271          JMP     .+5
01265   203161          LAC     ADRB
01266   043163          DAC     LTST
01267   203160          LAC     ADRA
01270   721000          PAX
                        /
                        .EJECT
```

```
                        /
01271   760261  SIMU    LAW     261
01272   543136          SAD     TNUM            /TEST 1 IF EQUAL
01273   601302          JMP     SIM1
01274   760262          LAW     262
01275   543136          SAD     TNUM            /TEST 2 IF EQUAL
01276   601323          JMP     SIM2
01277   760263          LAW     263
01300   543136          SAD     TNUM            /TEST 3 IF EQUAL
01301   601352          JMP     SIM3
                        /
01302   203206  SIM1    LAC     K7
01303   043202          DAC     XMSK
01304   143203          DZM     XCOM
01305   203127          LAC     PCWA
01306   043126  SIMC    DAC     CNTRL
01307   102072          JMS     CRLF
01310   203160          LAC     ADRA
01311   503237          AND     K70K
01312   043200          DAC     WORK
01313   730000          PLA
01314   740051          CMA!IAC
01315   343200          TAD     WORK
01316   721000          PAX
01317   770000          LAW     -10000
01320   043143          DAC     CT4K
01321   100417          JMS     WRITE           /WRITE PATTERN
01322   601343          JMP     STSCP
                        /
01323   203212  SIM2    LAC     K77             /WRITE PATTERN #2
01324   043202          DAC     XMSK
01325   203211          LAC     K40
01326   043203          DAC     XCOM
01327   203130          LAC     PCWB
01330   601306          JMP     SIMC
                        /
01331   707702  SCP1    EEM                     /SYNC
01332   210000          LAC     X               /READ
01333   740001          CMA                     /COMPLEMENT
01334   050000          DAC     X               /WRITE
01335   203200          LAC     WORK
01336   543163          SAD     LTST            /CHECK FOR END OF BLOCK
01337   601343          JMP     .+4
01340   737001          AXR+1                   /ADDRESS+1
01341   443200          ISZ     WORK
01342   601331          JMP     SCP1
01343   203160  STSCP   LAC     ADRA            /STARTING ADDRESS
01344   043200          DAC     WORK
01345   730000          PLA
01346   740051          CMA!IAC
01347   343200          TAD     WORK
01350   721000          PAX
01351   601331          JMP     SCP1
                        /
                        .EJECT
```

```
                        /
                        /ROUTINE TO SIMULATE TEST 3, ALL ONES
                        /ARE WRITTEN INTO ONE 4K FIELD, AND
                        /THEN THE ADDRESS IS LOOPED, THE
                        /X LINE SPECIFIED IN THE FIRST ADDR
                        /IS SET TO 000000, AND THEN READ.
                        /THE LAST ADDR, AND ALL BETWEEN, ARE
                        /NOT REFERENCED.
                        /
01352   102072  SIM3    JMS     CRLF
01353   203160          LAC     ADRA
01354   503237          AND     K70K            /MASK FIELD NUMBER
01355   043200          DAC     WORK
01356   730000          PLA
01357   740031          CMAIIAC
01360   343200          TAD     WORK
01361   721000          PAX
01362   770000          LAW     -10000
01363   043143          DAC     CT4K            /4K COUNTER
01364   777777          LAW     -1
01365   250000          DAC     X               /WRITE 1'S
01366   443143          ISZ     CT4K            /DONE 4K WHEN SKIP
01367   601364          JMP     .-3
01370   163160          DZM*    ADRA            /CLEAR X+Y LINE
01371   223160          LAC*    ADRA            /HANG HERE AND READ
01372   601371          JMP     .-1
                        /
                        /CHECK WANTED ADDRESS AND PROGRAM AREA
                        /
01373   000000  PROG    0
01374   100646          JMS     WHERE
01375   042072          DAC     CRLF            /SAVE
01376   760000          LAW
01377   543155          SAD     ADRCW           /NONE IF = LAW
01400   621373          JMP*    PROG
01401   203155          LAC     ADRCW
01402   503237          AND     K70K
01403   542072          SAD     CRLF            /C(CRLF) = CURRENT 4K BANK
01404   741000          SKP                     /EQUAL
01405   621373          JMP*    PROG            /EXIT
01406   441373          ISZ     PROG            /RETURN+1
01407   621373          JMP*    PROG            /EXIT
                        /
                        ,EJECT
```

```
                        /
                        /BIT SUPPRESSION INPUT ROUTINE, TYPE A
                        /CARRIAGE RETURN TO RESUME TESTING ALL BITS
                        /TO SUPPRESS, TYPE THE DECIMAL BIT POSITION(S)
                        /SEPARATING EACH WITH A COMMA, TERMINATE WITH
                        /A C.R, PRESS RUBOUT TO RESTART THE LINE IN
                        /CASE OF TYPING ERROR.
                        /
01410   000000  SUPBIT  0
01411   143167          DZM     SCW             /SUPPRESSION CONTROL WORD
01412   202662          LAC     SUPX            /POINTER FOR SUPPRESSS
01413   043154          DAC     PRNT
01414   202663          LAC     SUPXA           /C(SUPXA) = SUPBIT+1
01415   043157          DAC     OVER
01416   102072          JMS     CRLF            /CR, LF
01417   102034          JMS     PNXT            /PRINT "SUPPRESS"
                        /
01420   101544  AGAIN   JMS     KEYIN           /WAIT FOINPUT
01421   543216          SAD     K215            /CHECK FOR C.R.
01422   601467          JMP     EOT             /DONE SELECTING
01423   543226          SAD     K377            /CHECK FOR RUB-OUT
01424   601411          JMP     SUPBIT+1
01425   543217          SAD     K254            /CHECK FOR COMMA
01426   601420          JMP     AGAIN           /WAIT FOR NEXT BIT POS.
01427   101517          JMS     NUMB            /DETERMINE INPUT NUMBER
01430   601564          JMP     QUERY           /NOT VALID RESTART
                        /
01431   741200          SNA                     /CHECK FOR 0
01432   601513          JMP     ZERO            /POSITION 0
01433   043164          DAC     TTYW            /SAVE DIGIT
01434   101544          JMS     KEYIN           /WAIT FOR SECOND DIGIT
01435   543217          SAD     K254            /CHECK FOR COMMA
01436   601475          JMP     EOM             /2 DIGIT POSITION
01437   543216          SAD     K215            /CHECK FOR C.R.
01440   601501          JMP     EOTA            /DONE
01441   543226          SAD     K377            /RUB-OUT IF NO SKIP
01442   601411          JMP     SUPBIT+1                /START OVER
01443   101517          JMS     NUMB            /DETERMINE NUMBER
01444   601564          JMP     QUERY           /NOT VALID, RESTART
                        /
                        .EJECT
```

```
                        /
01445   043165          DAC     TTYX            /SAVE NUMBER
01446   203164          LAC     TTYW            /PREVIOUS DIGIT
01447   744010          RCL; RCL; RCL
01450   744010
01451   744010
01452   243165          XOR     TTYX            /COMBINE DIGITS
01453   740001          CMA                     /1' COMPLEMENT
01454   043166          DAC     TTYY            /SAVE
01455   777777          LAW     -1
01456   343166          TAD     TTYY            /SUBTRACT 1
01457   043166  ROTOR   DAC     TTYY
01460   203242          LAC     K400K           /400000
01461   744020          RCR
01462   443166          ISZ     TTYY            /SHIFT COUNT
01463   601461          JMP     .-2
01464   243167          XOR     SCW             /INSERT IN CONTROL WORD
01465   043167          DAC     SCW
01466   601420          JMP     AGAIN           /WAIT FOR NEXT BIT POSITION
                        /
01467   203167  EOT     LAC     SCW             /SELECTION COMPLETED
01470   740001          CMA
01471   043170          DAC     BITSUP
01472   143167          DZM     SCW
01473   102072          JMS     CRLF            /CR,LF
01474   621410          JMP*    SUPBIT          /EXIT
                        /
01475   203164  EOM     LAC     TTYW            /SINGLE DIGIT
01476   740001          CMA
01477   343204          TAD     K1
01500   601457          JMP     ROTOR
                        /
01501   203164  EOTA    LAC     TTYW            /INPUT DIGIT
01502   740001          CMA
01503   343204          TAD     K1              /2'S COMPLEMENT
01504   043166          DAC     TTYY
01505   203242          LAC     K400K           /400000
01506   744020          RCR
01507   443166          ISZ     TTYY            /SHIFT COUNTER
01510   601506          JMP     .-2
01511   243167          XOR     SCW
01512   621470          JMP     EOT+1           /EXIT
                        /
01513   203167  ZERO    LAC     SCW
01514   243242          XOR     K400K           /400000
01515   043167          DAC     SCW
01516   601420          JMP     AGAIN           /WAIT FOR NEXT
                        /
                        ,EJECT
```

```
                        /
01517   000000  NUMB    0
01520   203142          LAC     CT32            /ASCII INPUT
01521   503225          AND     K370
01522   543220          SAD     K260
01523   741000          SKP
01524   601531          JMP     .+5             /CHECK FOR A 270 OR 271
01525   441517          ISZ     NUMB            /RETURN+1
01526   203142          LAC     CT32
01527   503206          AND     K7
01530   621517          JMP*    NUMB            /EXIT
01531   543222          SAD     K270            /= 8 OR 9 IF EQUAL
01532   741000          SKP
01533   621517          JMP*    NUMB            /INVALID
01534   203142          LAC     CT32            /ASCII INPUT
01535   503204          AND     K1              /= A 8 IF BIT 17 = 0
01536   740200          SZA
01537   601542          JMP     .+3             /A 9
01540   777770          LAW     -10             /SHIFT COUNT OF 8
01541   601457          JMP     ROTOR
01542   777767          LAW     -11             /SHIFT COUNT OF 9
01543   601457          JMP     ROTOR
                        /
                        /
                        /CHARACTER INPUT ROUTINE
                        /
01544   000000  KEYIN    0
01545   700312          KRB                     /INITIALIZE
01546   700301          KSF                     /WAIT FOR INPUT
01547   601546          JMP     .-1
01550   700312          KRB                     /READ BUFFER
01551   043142          DAC     CT32            /SAVE
01552   621544          JMP*    KEYIN
                        /
                        /
                        /CHECK VALIDITY OF INPUT CHARACTER
                        /
01553   000000  LEGAL   0
01554   203142          LAC     CT32            /ASCII INPUT
01555   543226          SAD     K377            /IS IT A RUBOUT
01556   601034          JMP     KYBRD           /START OVER
01557   543216          SAD     K215            /CHECK FOR C.R.
01560   623156          JMP*    EXIT            /LINE TERMINATED
01561   503225          AND     K370
01562   543220          SAD     K260            /SHOULD EQUAL 260
01563   621553          JMP*    LEGAL           /O.K.
01564   102607  QUERY   JMS     WOTIS           /PRINT QUESTION MARK
01565   623157          JMP*    OVER            /START LINE OVER
                        /
                        .EJECT
```

```
                        /PRINT AREA CONTAINING PROGRAM
                        /
01566   000000  LOCAT   0
01567   750004          LAS
01570   503213          AND     K100
01571   740200          SZA
01572   621566          JMP*    LOCAT
01573   102072          JMS     CRLF            /CR, LF
01574   202667          LAC     PISIN
01575   043154          DAC     PRNT
01576   102034          JMS     PNXT            /PRINT "PROGRAM IS IN FIELD"
01577   100646          JMS     WHERE           /WHERE IS IT
01600   043200          DAC     WORK
01601   744010          RCL!    RTL!    RAL
01602   742010
01603   740010
01604   503206          AND     K7
01605   343220          TAD     K260
01606   102027          JMS     PCHAR           /PRINT 1ST HALF FIELD NO.
01607   203200          LAC     WORK
01610   744010          RCL!    RTL!    RTL!    RTL
01611   742010
01612   742010
01613   742010
01614   503206          AND     K7
01615   343220          TAD     K260
01616   102027          JMS     PCHAR           /PRINT 2ND HALF FIELD NO.
01617   102072          JMS     CRLF            /CR, LF
01620   621566          JMP*    LOCAT           /EXIT
                        /
                        .EJECT
```

```
                        /GENERATE 6 DIGIT ADDRESSES FROM KEYBOARD INPUT
                        /
01621   000000  GETAD   0
01622   707704          LEM
01623   222677          LAC*    ROTA            /GET A NEG, LAW FOR COUNT
01624   043162          DAC     NROTA           /SHIFT COUNTER
01625   203142          LAC     CT32            /ASCII INPUT
01626   443162  CNROT   ISZ     NROTA
01627   601645          JMP     GOLEFT          /ROTATE 1 LEFT
01630   243155          XOR     ADRCW           /ADR, CONTROL WORD
01631   043155          DAC     ADRCW
01632   777777          LAW     -1
01633   562677          SAD*    ROTA            /REC'D 6 DIGITS IF EQUAL
01634   601637          JMP     .+3
01635   442677          ISZ     ROTA            /LAW POINTER + 1
01636   621621          JMP*    GETAD           /EXIT AND WAIT FOR NEXT
01637   202700          LAC     ROTB            /ESTORE POINTERS
01640   042677          DAC     ROTA
01641   222677          LAC*    ROTA
01642   043162          DAC     NROTA
01643   441621          ISZ     GETAD           /RETURN+1
01644   621621          JMP*    GETAD           /EXIT
01645   744010  GOLEFT   RCL
01646   601626          JMP     CNROT
                        /
01647   777760  ROTC    LAW     -20             /ROTATE 15 LEFT FOR 1ST DIGIT
01650   777763          LAW     -15             /12 LEFT FOR 2ND
01651   777766          LAW     -12             /9 LEFT FOR 3RD
01652   777771          LAW     -7              /6 LEFT FOR 4TH
01653   777774          LAW     -4              /3 LEFT FOR 5TH
01654   777777          LAW     -1              /NONE FOR 6TH
                        /
                        .EJECT
```

```
                        /
                        /ROUTINE TO ACCEPT TEST LIMITS FROM KEYBOARD INPUT
                        /
21655   000000  SLMTS   0
21656   201620          LAC     GETAD-1
21657   041572          DAC     LOCAT+4         /RESTORE JMP*
21660   102072          JMS     CRLF            /CR, LF
21661   202670          LAC     TLMX            /TEST LIMITS POINTER
21662   043154          DAC     PRNT
21663   102034          JMS     PNXT            /PRINT "TEST LIMITS"
21664   102072          JMS     CRLF            /CR,LF
21665   202671          LAC     SLMX            /C (SLMX)=SLMTS+1
21666   043157          DAC     OVER
21667   202672          LAC     DON3            /RETURN ADDRESS=CREVR
21670   043156          DAC     EXIT
21671   101544          JMS     KEYIN           /WAIT FOR INPUT OF MEM, NO,
21672   543226          SAD     K377
21673   601656          JMP     SLMTS+1
21674   101553          JMS     LEGAL           /SE IF VALID
21675   203142          LAC     CT32            /ASCII INPUT
21676   503206          AND     K7              /MASK 15,16 AND 17
21677   744020          RCR;    RTR;    RAR
21700   742020
21701   740020
21702   043147          DAC     FIRST1
21703   101544          JMS     KEYIN           /WAIT FOR INPUT OF FIELD NO,
21704   543226          SAD     K377
21705   601656          JMP     SLMTS+1
21706   101553          JMS     LEGAL           /SEE IF VALID
21707   203142          LAC     CT32            /ASCII INPUT
21710   503206          AND     K7              /MASK 15,16 AND 17
21711   744020          RCR;    RTR;    RTR;    RTR
21712   742020
21713   742020
21714   742020
21715   243147          XOR     FIRST1
21716   043147          DAC     FIRST1          /FIRST FIELD TO TEST IS STORED
21717   101544          JMS     KEYIN           /WAIT FOR COMMA
21720   543217          SAD     K254
21721   741000          SKP
21722   601564          JMP     QUERY           /PRINT QUERY,  AND RESTART
21723   101544          JMS     KEYIN           /WAIT FOR INPUT OF MEM, NO,
21724   543226          SAD     K377
21725   601656          JMP     SLMTS+1
21726   101553          JMS     LEGAL           /SEE IF VALID
21727   203142          LAC     CT32            /ASCII INPUT
21730   503206          AND     K7
21731   744020          RCR;    RTR;    RAR
21732   742020
21733   740020
21734   043150          DAC     LAST1
21735   101544          JMS     KEYIN           /WAIT FOR INPUT OF FIEL NO,
21736   543226          SAD     K377
21737   601656          JMP     SLMTS+1
21740   101553          JMS     LEGAL           /SEE IF VALID
```

```
01741  203142          LAC     CT32            /ASCII INPUT
01742  503206          AND     K7              /MASK 15,16 AND 17
01743  744020          RCR;    RTR;    RTR;    RTR
01744  742020
01745  742020
01746  742020
01747  243150          XOR     LAST1
01750  043150          DAC     LAST1           /LAST FIELD TO TEST IS STORED
01751  777777          LAW     -1
01752  043143          DAC     CT4K
01753  443143  CREVR   ISZ     CT4K            /NO 2ND DIGIT IF NO SKIP
01754  601564          JMP     QUERY           /PRINT QUERY AND RESTART
01755  203147          LAC     FIRST1          /FIRST FIELD
01756  740001          CMA
01757  343204          TAD     K1              /2'S COMPLEMENT
01760  343150          TAD     LAST1           /FIRST IS >LAST IF NEG,
01761  740100          SMA
01762  601771          JMP     OKAS            /FIRST IS LOWEST ORDER
                       /
                       .EJECT
```

```
                        /
01763   203147          LAC     FIRST1
01764   043146          DAC     CT04            /SAVE
01765   203150          LAC     LAST1
01766   043147          DAC     FIRST1          /LAST IS NOW FIRST
01767   203146          LAC     CT04
01770   043150          DAC     LAST1           /FIRST IS NOW LAST
01771   203150  OKAS    LAC     LAST1
01772   543147          SAD     FIRST1          /SEE IF ONLY 1 SELECTED
01773   741000          SKP                     /YES, SEE IF IT HAS PROGRAM
01774   602004          JMP     ALOK
01775   543122          SAD     INSFLD          /REJECT IF EQUAL,
01776   741000          SKP                     /TELL WHERE IT IS
01777   602004          JMP     ALOK
02000   760000          LAW
02001   041572          DAC     LOCAT+4         /CHANGE JMP* TO NOP
02002   101506          JMS     LOCAT
02003   601656          JMP     SLMTS+1         /RESTART
02004   101544  ALOK    JMS     KEYIN           /WAIT FOR A C,R
02005   543216          SAD     K215
02006   621655          JMP*    SLMTS           /EXIT
02007   601564          JMP     QUERY           /PRINT QUERY AND RESTART
                        /
                        /SETUP ACS, PRESS CARRIAGE RETURN TO EXIT
                        /
02010   000000  SETAC   0
02011   102072          JMS     CRLF            /CR,LF
02012   202673          LAC     SETX            /POINTER
02013   043154          DAC     PRNT
02014   102034          JMS     PNXT            /PRINT "SETUP ACS"
02015   700312          KRB
02016   700301          KSF
02017   602016          JMP     .-1
02020   700312          KRB
02021   543226          SAD     K377            /CHECK FOR RO
02022   601656          JMP     SLMTS+1         /RESTART
02023   750004          LAS
02024   043124          DAC     MQWA
02025   102072          JMS     CRLF            /CR,LF
02026   622010          JMP*    SETAC           /EXIT
                        /
                        .EJECT
```

```
                        /
                        /PRINT ROUTINES FOR MESSAGES
                        /
                        /PRINT ONE CHARACTER AND EXIT
                        /
02027   000000  PCHAR   0
02030   700406          TLS
02031   700401          TSF
02032   602031          JMP     .-1
02033   622027          JMP*    PCHAR
                        /
                        /PRINT A STRING AND EXIT.
                        /
02034   000000  PNXT    0
02035   777775          LAW     -3
02036   043142          DAC     CT32            /CHARACTER COUNTER
02037   443154          ISZ     PRNT            /WORD POINTER+1
02040   223154          LAC*    PRNT
02041   741200          SNA                     /ALL DONE IF 0
02042   622034          JMP*    PNXT            /EXIT
02043   042072  MASK    DAC     CRLF            /SAVE WORD
02044   503212          AND     K77             /MASK 6 BIT CHARACTER
02045   543212          SAD     K77             /CHECK IF RUBOUT
02046   602057          JMP     CK3
02047   043137          DAC     BITCON          /SAVE CHAR
02050   777740          LAW     -40
02051   343137          TAD     BITCON
02052   740100          SMA                     /NEG, = ALPHA
02053   602067          JMP     CRLF-3          /NUMERIC
02054   242050          XOR     .-4
02055   243223          XOR     K300            /MAKE ALPHA
02056   102027          JMS     PCHAR           /PRINT ACS 10-17
02057   443142  CK3     ISZ     CT32            /CHECK FOR 3 CHARACTERS
02060   741000          SKP
02061   602035          JMP     PNXT+1          /GET NEXT 3 CHARACTERS
02062   202072          LAC     CRLF            /POSITION NEXT
02063   742020          RTR; RTR; RTR
02064   742020
02065   742020
02066   602043          JMP     MASK            /PRINT IT
02067   203137          LAC     BITCON
02070   343214          TAD     K200            /MAKE NUMERIC
02071   602056          JMP     CK3+1
                        /
                        .EJECT
```

```
                        /CARRIAGE RETURN, LINE FEED
                        /
02072   000000  CRLF    0
02073   760215          LAW     215             /ASCII CR
02074   102027          JMS     PCHAR
02075   542077          SAD     .+2
02076   622072          JMP*    CRLF            /EXIT
02077   760212          LAW     212             /LF
02100   602074          JMP     CRLF+2
                        /
                        /PRINT SPACES
                        /
02101   000000  SPING   0
02102   760240          LAW     240             /ASCII SPACE
02103   102027          JMS     PCHAR
02104   443142          ISZ     CT32            /COUNTER
02105   602102          JMP     SPING+1
02106   622101          JMP*    SPING           /EXIT
                        /
                        /PRINT SIX DIGIT OCTAL NUMBERS
                        /
02107   000000  PROCTL  0
02110   777772          LAW     -6
02111   043142          DAC     CT32            /DIGIT COUNTER
02112   202072          LAC     CRLF            /OCTAL NUMBER
02113   744010  POSITN  RCL!    RTL
02114   742010
02115   042072          DAC     CRLF
02116   740010          RAL
02117   503206          AND     K7
02120   343220          TAD     K260
02121   102027          JMS     PCHAR           /PRINT
02122   443142          ISZ     CT32
02123   602112          JMP     POSITN-1                /POSITION NEXT DIGIT
02124   622107          JMP*    PROCTL          /EXIT
                        /
                        .EJECT
```

```
                        /XCH15 - TAPE 3
                        /
                        /ROUTINE TO DETERMINE FIELD FOR RELOCATION
                        /
02125   202674  CMOVE   LAC     ERTBL
02126   043250          DAC     ERWRD
02127   203150          LAC     LAST1           /LAST TO TEST
02130   543147          SAD     FIRST1          /DON'T MOVE IF EQUAL
02131   600216          JMP     RTN1            /RETURN
02132   203121          LAC     FLAGS           /PROGRAM FLAGS
02133   741100          SPA                     /FORCED MOVE MADE IF A 1,
02134   600216          JMP     RTN1            /DON'T MOVE
02135   740020          RAR                     /LINK = BIT 17
02136   741400          SZL                     /FIRST MOVE IF SKIP
02137   602224          JMP     NXTMV           /SETUP FOR NEXT MOVE
02140   443121          ISZ     FLAGS           /SET FLAG FOR 1ST MOVE
02141   203150          LAC     LAST1
02142   043122          DAC     INSFLD
02143   770000          LAW     -10000          /-4K
02144   343122          TAD     INSFLD          /SUBTRACT 4K FROM CURRENT
02145   043171          DAC     NXLOC           /NXLOC = DEST'N FOR NEXT TIME,
02146   100646          JMS     WHERE           /WHERE ARE WE NOW
02147   543122          SAD     INSFLD          /ALREADY IN LAST 1 IF EQUAL
02150   602207          JMP     SUB1            /TRY NEXT LOWER
                        /
                        /NOW CHECK FOR ERROR RECORDED IN NEW FIELD
                        /
02151   760000  CKERR   LAW
02152   563250          SAD*    ERWRD           /NO ERRORS IF = LAW
02153   602164          JMP     STMV            /INITIALIZE MOVE
02154   223250          LAC*    ERWRD
02155   543122          SAD     INSFLD          /ERROR IN FIELD IF EQUAL
02156   602176          JMP     EQUAL
02157   443250          ISZ     ERWRD           /POINTER + 1
02160   203250          LAC     ERWRD
02161   542675          SAD     ENERR           /END OF TABLE IF EQUAL
02162   741000          SKP
02163   602154          JMP     CKERR+3
                        /
02164   202674  STMV    LAC     ERTBL
02165   043250          DAC     ERWRD           /RESTORE POINTER
02166   203122          LAC     INSFLD          /NEW FIELD
02167   043173          DAC     DESTN
02170   100646          JMS     WHERE
02171   043172          DAC     SOURCE
02172   543173          SAD     DESTN
02173   600216          JMP     RTN1            /NEW AND CURRENT ARE EQUAL
02174   203173          LAC     DESTN
02175   602343          JMP     MOVE            /MOVE PROGRAM
                        /
                        .EJECT
```

```
                        /ERROR IN NEW FIELD, TRY NEXT LOWER
                        /
02176   543147  EQUAL   SAD     FIRST1          /DON'T TRY NEXT IF EQUAL
02177   602221          JMP     DNMVE
02200   741200          SNA                     /IS IT FIELD 0
02201   602205          JMP     .+4             /YES
02202   770000          LAW     -10000          /-4K
02203   343122          TAD     INSFLD          /SUBTRACT 4K FROM NEW FIELD
02204   043171          DAC     NXLOC           /NEXT NEW FIELD
02205   202674          LAC     ERTBL
02206   043250          DAC     ERWRD           /RESTORE POINTER
                        /
02207   203171  SUB1    LAC     NXLOC           /NEXT NEW FIELD
02210   543122          SAD     INSFLD          /IS IT = CURRENT NEW FIELD
02211   602176          JMP     EQUAL           /TRY NEXT LOWER
02212   043122          DAC     INSFLD          /NEW NEW FIELD
02213   543147          SAD     FIRST1          /DOES IT = LOWEST FIELD
02214   602151          JMP     CKERR           /CHECK FOR ERROR
02215   770000          LAW     -10000
02216   343122          TAD     INSFLD          /SUBTRACT 4K
02217   043171          DAC     NXLOC           /NEW FIELD FOR NEXT PASS
02220   602151          JMP     CKERR
                        /
02221   202674  DNMVE   LAC     ERTBL
02222   043250          DAC     ERWRD           /RESTORE POINTER
02223   600216          JMP     RTN1            /START OVER
                        /
                        .EJECT
```

```
                        /
                        /ROUTINE TO DETERMINE PROGRAM DEST'N AFTER MAKING ONE MOVE
                        /
02224   100646  NXTMV    JMS     WHERE           /WHERE IS PROGRAM NOW
02225   043172          DAC     SOURCE
02226   760000  CKNXT   LAW
02227   563250          SAD*    ERWRD           /NO ERRORS IF 1ST = LAW
02230   602243          JMP     STNXT
02231   202674          LAC     ERTBL
02232   043250          DAC     ERWRD
02233   223250          LAC*    ERWRD           /GET AN ERROR ADDRESS
02234   543171          SAD     NXLOC
02235   602263          JMP     SUB2            /ERROR IN NEXT FIELD TRY NEXT
02236   443250          ISZ     ERWRD
02237   203250          LAC     ERWRD
02240   542675          SAD     ENERR
02241   741000          SKP                     /DONE TABLE AND NO ERRORS
02242   602233          JMP     CKNXT+5
                        /
02243   202674  STNXT   LAC     ERTBL
02244   043250          DAC     ERWRD           /RESTORE POINTER
02245   203171          LAC     NXLOC           /NEW FIELD
02246   543122          SAD     INSFLD          /DOES IT = CURRENT FIELD
02247   602252          JMP     .+3
02250   543147          SAD     FIRST1          /DOES IT = LOWEST FIELD
02251   602300          JMP     MVBK            /YES, CLEAR FLAGS AND MOVE
02252   543147          SAD     FIRST1          /DOES THE CU RENT ALSO=
                                                /THE LOWEST FIELD,
02253   602274          JMP     NXTHI           /YES, SETUP FOR HIGHEST FIELD
02254   043122          DAC     INSFLD          /NEW CURRENT FIELD
02255   770000          LAW     -10000          /-4K
02256   343122          TAD     INSFLD
02257   043171          DAC     NXLOC           /NEW NEXT FIELD
02260   203122          LAC     INSFLD
02261   043173          DAC     DESTN
02262   602343          JMP     MOVE            /MOVE FROM HERE TO C (DESTN)
                        .EJECT
```

```
                        /
02263   203171  SUB2    LAC     NXLOC           /IS NEXT = FIELD 02 OR 1ST TO TEST
02264   543147          SAD     FIRST1
02265   602221          JMP     DNMVE           /YES, DON'T MOVE
02266   770000          LAW     -10000          /-4K
02267   343171          TAD     NXLOC           /NEW NEXT FIELD
02270   043171          DAC     NXLOC
02271   543122          SAD     INSFLD          /DOES IT = CURRENT FIELD
02272   602264          JMP     SUB2+1          /YES
02273   602231          JMP     CKNXT+3         /SEE IF ERROR IN NEW FIELD
                        /
02274   203150  NXTHI   LAC     LAST1           /LAST TO TEST
02275   503244          AND     K370K
02276   043171          DAC     NXLOC           /LAST = NEXT FIELD
02277   602231          JMP     CKNXT+3         /CHECK FOR ERROR
                        /
02300   120646  MVBK    JMS     WHERE
02301   043172          DAC     SOURCE
02302   203171          LAC     NXLOC
02303   043122          DAC     INSFLD
02304   043173          DAC     DESTN
02305   143121          DZM     FLAGS
02306   602343          JMP     MOVE
                        /
                        .EJECT
```

```
                        /ROUTINE TO FORCE MOVE THE PROGRAM, DESTINATION
                        /FIELD# MUST BE TYPED IN BY THE OPERATOR (00-37 OCTAL).
                        /
02307   203242  FCDMV   LAC     K400K
02310   740001          CMA
02311   503121          AND     FLAGS
02312   243242          XOR     K400K           /SET BIT 0 FOR FCDMV FLAG
02313   043121          DAC     FLAGS
02314   202674          LAC     ERTBL
02315   043250          DAC     ERWRD           /RESTORE TABLE POINTER
02316   102452          JMS     GOTO            /PRINT GO TO FIELD
                        /
                        /CHECK FOR ERROR IN NEW FIELD
                        /
02317   760000  CKFCD   LAW
02320   563250          SAD*    ERWRD           /NO ERRORS IF 1ST * LAW
02321   602343          JMP     MOVE            /SEE WHERE TO GO
02322   223250          LAC*    ERWRD
02323   543173          SAD     DESTN           /DOES ERROR * NEW FIELD
02324   602332          JMP     XPRT            /YES, PRINT MESSAGE
02325   443250          ISZ     ERWRD           /POINTER+1
02326   203250          LAC     ERWRD
02327   542675          SAD     ENERR           /SEE IF END OF TABLE
02330   602333          JMP     .+3             /DONE AND NO ERRORS
02331   602322          JMP     CKFCD+3
02332   102430  XPRT    JMS     PRSEL           /PRINT ERROR IN SELECTED 4K
                        /
02333   202674          LAC     ERTBL
02334   043250          DAC     ERWRD
02335   203173          LAC     DESTN
02336   543172          SAD     SOURCE
02337   600216          JMP     RTN1
02340   043122          DAC     INSFLD
02341   503241          AND     K300K
02342   043176          DAC     PINX
                        /
                        .EJECT
```

```
                        /
                        /ROUTINE TO RELOCATE THE PROGRAM
                        /
02343   142600  MOVE    DZM     LOCER
02344   203173          LAC     DESTN
02345   503241          AND     K300K
02346   043176          DAC     PINX
02347   770000          LAW     -10000          /-4K
02350   043143          DAC     CT4K            /4K COUNTER
02351   203172          LAC     SOURCE          /CURRENT FIELD
02352   043174          DAC     MOVES
02353   740051          CMAIIAC
02354   343173          TAD     DESTN
02355   721000          PAX                     /NEW FIELD
02356   143201          DZM     MEMADR          /LOC ZERO START.
02357   223174  MOSOM   LAC*    MOVES           /MOVE FROM CURRENT
02360   043136          DAC     TNUM            /SAVE
02361   102457          JMS     RT19L
02362   050000          DAC     X               /PUT IN NEW FIELD
02363   210000          LAC     X               /READ BACK
02364   563174          SAD*    MOVES           /COMPARE
02365   741000          SKP                     /OK
02366   102551          JMS     MVERR           /PRINT ERROR INFO
02367   443174          ISZ     MOVES           /INCREMENT ADDRESSES
02370   737001          AXR*1
02371   443143          ISZ     CT4K
02372   741000          SKP
02373   602421          JMP     DIND
02374   203136          LAC     TNUM
02375   542652          SAD     DLMT            /DELIMITING CHARACTER
02376   741000          SKP                     /ADJUST INDIRECTS
02377   602357          JMP     MOSOM
                        /
                        .EJECT
```

```
02400  223174   AJIN   LAC*   MOVES
02401  542713          SAD    DLMTA          /DONE INDIRECTS IF EQUAL
02402  602360          JMP    MOSOM+1
02403  503232          AND    K7777          /MASK ADDRESS BITS
02404  243173          XOR    DESTN          /PUT FIELD NUMBER ON IT
02405  102437          JMS    RT19L
02406  050000          DAC    X              /PUT IN NEW FIELD
02407  210000          LAC    X              /READ BACK
02410  503232          AND    K7777
02411  243172          XOR    SOURCE
02412  563174          SAD*   MOVES          /COMPARE
02413  741000          SKP                   /OK
02414  102551          JMS    MVERR          /PRINT ERROR INFO
02415  443174          ISZ    MOVES          /INCREMENT ADDRESSES
02416  737001          AXR+1
02417  443143          ISZ    CT4K
02420  602400          JMP    AJIN
                       /
02421  102525   DIND   JMS    ENOT           /WAS TRANSFER MADE OK
02422  203172          LAC    SOURCE
02423  740031          CMAIIAC
02424  343173          TAD    DESTN
02425  721000          PAX
02426  740000          NOP
                                              /EXIT FROM HERE TO LOC
02427  610216          JMP    RTN1,X         /RTN1 IN NEW FIELD
                       /
                       .EJECT
```

```
                        /
                        /PRINT ERROR IN SELECTED 4K
                        /
02430   000000  PRSEL   0
02431   102072          JMS     CRLF            /CR, LF
02432   202712          LAC     ERSEL           /TEXT POINTER
02433   043154          DAC     PRNT
02434   102034          JMS     PNXT            /PRINT
02435   102072          JMS     CRLF
02436   602307          JMP     FCOMV           /WAIT FOR ANOTHER CHOICE
                        /
                        /ROTATE INSTRUCTION 19 LEFT BEFORE MOVING
                        /
02437   000000  RT19L   0
02440   744000          CLL                     /LINK = 0
02441   043136          DAC     TNUM            /SAVE
02442   777767          LAW     -11             /-9 DECIMAL
02443   043146          DAC     CT04            /SHIFT COUNT
02444   203136          LAC     TNUM            /INSTRUCTION
02445   740010          RAL
02446   742010          RTL
02447   443146          ISZ     CT04
02450   602446          JMP     .-2
02451   622437          JMP*    RT19L
                        /
                        /KEYBOARD ROUTINE FOR FORCED RELOCATION
                        /
02452   000000  GOTO    0
02453   750004          LAS                     /READ ACS
02454   503211          AND     K40
02455   741200          SNA                     /CHECK BIT 12
02456   602470          JMP     NOSW            /EQUALS 0
02457   102072          JMS     CRLF            /CR,LF
02460   202701          LAC     PTWLV           /TEXT POINTER
02461   043154          DAC     PRNT
02462   102034          JMS     PNXT            /PRINT PUT ACS 12 ON A 0
02463   750004          LAS
02464   503211          AND     K40
02465   740200          SZA                     /WAIT FOR THE 0
02466   602465          JMP     .-3
02467   102072          JMS     CRLF            /CR,LF X 2
02470   102072  NOSW    JMS     CRLF
02471   202702          LAC     GOFL            /TEXT POINTER
02472   043154          DAC     PRNT
02473   102034          JMS     PNXT            /PRINT GO TO FIELD =
02474   101544          JMS     KEYIN           /WAIT FOR INPUT
02475   543216          SAD     K215            /A CR = NO FORCED MOVE
                                                /AND RESUME AUTO RELOCATE
02476   602544          JMP     CFLG            /CLEAR THE FORCED MOVE FLAG
                        /
                        .EJECT
```

```
02477   543226          SAD     K377            /RUBOUT, RE-PRINT; GO TO FIELD =
02520   602470          JMP     NOSW
02521   742020          RTR;    RTR
02522   742020
02523   503241          AND     K300K           /MASK BITS 1,2,
02524   043173          DAC     DESTN           /FIRST CHAR, OF FIELD NO,
02525   101544          JMS     KEYIN           /WAIT FOR INPUT,
02526   543216          SAD     K215            /A CR = NO FORCED MOVE
                                                /AND RESUME AUTO RELOCATE,
02527   602544          JMP     CFLG            /CLEAR THE FORCED MOVE FLAG,
02510   543226          SAD     K377            /RUBOUT, RE-PRINT; GO TO FIELD =
02511   602470          JMP     NOSW
02512   742020          RTR;    RTR;    RTR;    RAR
02513   742020
02514   742020
02515   740020
02516   503237          AND     K70K            /MASK BITS 3,4 & 5,
02517   243173          XOR     DESTN           /OR FIRST & SECOND CHARS,
02520   043173          DAC     DESTN           /NEW FIELD,
02521   100646          JMS     WHERE           /WHERE ARE WE NOW
02522   043172          DAC     SOURCE          /CURRENT FIELD
02523   102072          JMS     CRLF            /CR, LF
02524   622452          JMP*    GOTO            /CHECK FOR ERROR
02525   000000  ENOT    0
02526   202600          LAC     LOCER
02527   741200          SNA                     /NO ERRORS IF 0
02530   622525          JMP*    ENOT            /ENTER NEW FIELD
                /
02531   707704          LEM
02532   142600          DZM     LOCER
02533   102072          JMS     CRLF            /CR,LF
02534   202704          LAC     NERN            /TEXT POINTER
02535   043154          DAC     PRNT
02536   102034          JMS     PNXT            /PRINT NO MORE ERRORS
02537   102072          JMS     CRLF            /CR,LF
02540   203121          LAC     FLAGS
02541   741100          SPA                     /ACS 0 A 1 = FORCED MOVE
02542   602307          JMP     FCOMV           /WAIT FOR ANOTHER CHOICE
02543   602243          JMP     STNXT           /TRY NEXT FIELD LOWER
                /
02544   203242  CFLG    LAC     K400K
02545   740001          CMA
02546   503121          AND     FLAGS           /CLEAR THE FORCED MOVE FLAG
02547   043121          DAC     FLAGS
02550   600216          JMP     RTN1            /START OVER
                        ,EJECT
```

```
                        /
02551   000000  MVERR   0
02552   043151          DAC     BAD1            /SAVE INCORRECT INSTRUCTION
02553   721000          PAX                     /FIELD AND ADDRESS
02554   043152          DAC     OCADR           /SAVE
02555   223174          LAC*    MOVES           /CORRECT INSTRUCTION
02556   043153          DAC     GOOD1           /SAVE
02557   202616          LAC     PHDR
02560   741200          SNA
02561   102616          JMS     PHDR
02562   202600          LAC     LOCER
02563   741200          SNA                     /DON'T PRINT IF 1
02564   102600          JMS     LOCER           /PRINT PROGRAM RELOCATION ERROR
02565   202615          LAC     JMP3            /JMP LOCER-3
02566   041006          DAC     INDY
02567   102072          JMS     CRLF
02570   777766          LAW     -12             /-10 DECIMAL
02571   600760          JMP     STER            /PRINT INFO
                        /
02572   200761          LAC     STER+1          /EQUALS JMS SPING
02573   041006          DAC     INDY
02574   750004          LAS
02575   741100          SPA
02576   100653          JMS     HALT
02577   622551          JMP*    MVERR           /EXIT
                        /
02600   000000  LOCER   0
02601   102072          JMS     CRLF            /CR,LF
02602   202703          LAC     RELOC           /TEXT POINTER
02603   043154          DAC     PRNT
02604   102034          JMS     PNXT            /PRINT PROGRAM RELOCATION ERROR
02605   102072          JMS     CRLF            /CR,LF X 2
02606   622600          JMP*    LOCER           /EXIT AND PRINT THE ERROR
                        /
02607   000000  WOTIS   0
02610   102072          JMS     CRLF            /CR,LF
02611   760277          LAW     277             /QUERY MARK
02612   102027          JMS     PCHAR           /PRINT
02613   102072          JMS     CRLF            /CR,LF
02614   622607          JMP*    WOTIS           /EXIT
                        /
02615   602572  JMP3    JMP     LOCER-6
                        .EJECT
```

```
                        /
                        /HEADER ROUTINE
                        /
02616   000000  PHDR    0
02617   102072          JMS     CRLF            /CR,LF
02620   202705          LAC     TSTX            /POINTER FOR "TEST"
02621   043154          DAC     PRNT
02622   102034          JMS     PNXT            /PRINT TEST
02623   102645          JMS     CLMN            /SPACE 5
02624   202706          LAC     ADRXA           /"OCTAL ADR,"
02625   043154          DAC     PRNT
02626   102034          JMS     PNXT
02627   102645          JMS     CLMN            /SPACE 5
02630   202707          LAC     GDATX           /"GOOD"
02631   043154          DAC     PRNT
02632   102034          JMS     PNXT
02633   102645          JMS     CLMN            /SPACE 5
02634   202710          LAC     BDATX           /"BAD"
02635   043154          DAC     PRNT
02636   102034          JMS     PNXT
02637   102645          JMS     CLMN            /SPACE 5
02640   202711          LAC     PCWX            /"PAT,CONTROL WORD"
02641   043154          DAC     PRNT
02642   102034          JMS     PNXT
02643   102072          JMS     CRLF            /CR,LF
02644   622616          JMP*    PHDR            /DONE
                        /
02645   000000  CLMN    0
02646   777773          LAW     -5
02647   043142          DAC     CT32
02650   102101          JMS     SPING           /SPACE
02651   622645          JMP*    CLMN
                        /
                        .EJECT
```

```
                        /
                        /RETURN ADDRESSES (INDIRECTS)
                        /
02652   752521  DLMT    752521
02653   002756  TSNX    TSTNR                   /TEST#=
02654   002762  ADRX    FADR1                   /FIRST ADR,=
02655   001067  AD1R    ADR1
02656   001117  DON1    DFST
02657   002770  ADXR    LADR1                   /LAST ADR,=
02660   001143  AD2R    ADR2
02661   001171  DON2    DLST
02662   002775  SUPX    SUPR                    /SUPPRESS=
02663   001411  SUPXA   SUPBIT+1
02664   003060  ADR1P   FRST                    /FIRST
02665   003064  ADR2P   LSTA                    /LAST
02666   003002  OVRLP   OVRLAP                  /ADR, IS WITHIN PROGRAM
02667   003034  PISIN   PROIS                   /PROGRAM IS IN FIELD
02670   003045  TLMX    TSLM                    /TEST LIMITS
02671   001656  SLMX    SLMTS+1
02672   001753  DON3    CREVR
02673   003053  SETX    STACS                   /SETUP ACS
02674   003251  ERTBL   ERWRD+1
02675   003311  ENERR   ERWRD+41
02676   003070  PTO     PTOI
02677   001647  ROTA    ROTC
02700   001647  ROTB    ROTC
02701   003107  PTWLV   PUT12
02702   003101  GOFL    GOFLD
02703   003013  RELOC   PROR
02704   003025  NERN    NOMO
02705   002714  TSTX    TST
02706   002720  ADRXA   ADR
02707   002725  GDATX   GDAT
02710   002731  BDATX   BDAT
02711   002734  PCWX    PCWR
02712   002744  ERSEL   SLTER
02713   752522  DLMTA   752522
                        ,EJECT
```

```
                        /
                        /CONSTANTS FOR PRINT ROUTINE TEXTS, PACKED
                        /3 CHARACTERS PER WORD,
                        /
                        /"TEST"
02714   002714  TST     ,
02715   230524          230524; 777724; 0
02716   777724
02717   000000
                        /
02720   002720  ADR     ,
02721   240317          240317; 401401; 220401; 0
02722   401401
02723   220401
02724   000000
                        /
02725   002725  GDAT    ,
02726   171707          171707; 777704; 0
02727   777704
02730   000000
                        /
02731   002731  BDAT    ,
02732   040102          040102; 0
02733   000000
                        /
02734   002734  PCWR    ,
02735   240120          240120; 170340; 222416; 401417
02736   170340
02737   222416
02740   401417
02741   221727          221727; 777704; 0
02742   777704
02743   000000
                        /
02744   002744  SLTER   ,
02745   222205          222205; 402217; 401611; 140523
02746   402217
02747   401611
02750   140523
02751   240305          240305; 400405; 051106; 770414
02752   400405
02753   051106
02754   770414
02755   000000          0
                        /
02756   002756  TSTNR   ,
02757   230524          230524; 774024; 0
02760   774024
02761   000000
                        /
02762   002762  FADR1   ,
02763   221106          221106; 402423; 220401; 777740
02764   402423
02765   220401
02766   777740
```

```
02767   000000                  0

                        /
02770   002770          LADR1   ,
02771   230114                  2301141 0140241 402204
02772   014024
02773   402204
02774   000000                  0

                        /
02775   002775          SUPR    ,
02776   202523                  2025231 0522201 402323; 0
02777   052220
03000   402323
03001   000000

                                .EJECT
```

```
                        /
03002   003002  OVRLAP  ,
03003   220401          220401; 231140; 112740; 111024
03004   231140
03005   112740
03006   111024
03007   204016          204016; 071722; 150122; 0
03010   071722
03011   150122
03012   000000
                        /
03013   003013  PROR    ,
03014   172220          172220; 012207; 224015; 171405
03015   012207
03016   224015
03017   171405
03020   240103          240103; 161711; 220540; 221722
03021   161711
03022   220540
03023   221722
03024   000000          0
                        /
03025   003025  NOMO    ,
03026   401716          401716; 221715; 054005; 172222
03027   221715
03030   054005
03031   172222
03032   772322          772322; 0
03033   000000
                        /
03034   003034  PROIS   ,
03035   172220          172220; 012207; 114015; 114023
03036   012207
03037   114015
03040   114023
03041   064016          064016; 140511; 774004; 0
03042   140511
03043   774004
03044   000000
                        /
03045   003045  TSLM    ,
03046   230524          230524; 144024; 111511; 772324
03047   144024
03050   111511
03051   772324
03052   000000          0
                        /
03053   003053  STACS   ,
03054   240523          240523; 402025; 230301; 0
03055   402025
03056   230301
03057   000000
                        /
03060   003060  FRST    ,
03061   221106          221106; 402423; 0
```

```
03062   402423
03063   000000
                        /
03064   003064  LSTA    .
03065   230114          230114; 774024; 0
03066   774024
03067   000000
                        /
03070   003070  PTOI    .
03071   112220          112220; 402416; 242517; 114023
03072   402416
03073   242517
03074   114023
03075   111016          111016; 241102; 770405; 0
03076   241102
03077   770405
03100   000000
                        /
03101   003101  GOFLD   .
03102   401707          401707; 401724; 051106; 400414
03103   401724
03104   051106
03105   400414
03106   000000          0
                        /
03107   003107  PUT12   .
03110   242520          242520; 030140; 614023; 174062
03111   030140
03112   614023
03113   174062
03114   014016          014016; 776040; 0
03115   776040
03116   000000
                        .EJECT
```

```
                        /STORAGE AND CONSTANT REGISTERS
                        /
03117   777770  SIXT4   LAW     -10             /COUNTS 64 PASSES BETWEEN
                                                /ERROR PRINT SUPPRESSION,
03120   000000  NOPRNT  0                       /INDICATES END OF ERROR PRINT-OUTS
03121   000000  FLAGS   0                       /SAVES SUBROUTINE FLAGS
03122   000000  INSFLD  0                       /CURRENT FIELD WITH PROGRAM
03123   000000  LAST    0                       /LAST FIELD WITH DATA ERROR
03124   000000  MCWA    0                       /SAVES ACS SETTINGS
03125   000000  PCW     0                       /CURRENT PAT.CONTROL WORDS
03126   000000  CNTRL   0                       /SAME AS PCW
03127   776000  PCWA    776000                  /CONTROL WORD FOR TEST 1
03130   525250  PCWB    525250                  /FOR TEST 2
03131   777777  PCWC    777777                  /FOR TEST 3
03132   000000  PATR    0                       /ROTATES CONTROL WORD
03133   000000  PATG    0                       /SAVES GOOD DATA DURING READ
03134   000000  PATWD   0                       /SAME AS PATG BUT HAS SUPPRESSES BITS
03135   000000  PATN    0                       /HAS CONTROL WORD TO ROTATE
03136   000004  TNUM    4                       /ASCII TEST NUMBER
03137   000000  BITCON  0
03140   000000  SVADR   0                       /FIELD COUNTER
03141   000000  CT02    0
03142   000000  CT32    0
03143   000000  CT4K    0                       /4K COUNTER
03144   000000  CT16    0                       /COUNTS 16 ROTATES
03145   000000  CT128   0                       /COUNTS 128 LOCATIONS
03146   000000  CT04    0                       /UTILITY COUNTER
03147   000000  FIRST1  0                       /FIRST FIELD TO TEST
03150   000000  LAST1   0                       /LAST FIELD TO TEST
03151   000000  BAD1    0                       /SAVES BAD DATA
03152   000004  OCADR   4                       /SAVES FAILING OCTAL A DRESS
03153   000000  GOOD1   0                       /GOOD DATA
03154   000000  PRNT    0                       /POINTER FOR PRINT ROUTINES
03155   000000  ADRCW   0                       /PARTIAL ADDRESS WORD
03156   000004  EXIT    4                       /TO DISMISS
03157   000000  OVER    0                       /POINTER TO START OF SUBROUTINES
03160   000000  ADRA    0                       /1ST ADR, FROM KEYBOARD INPUT
03161   000000  ADRB    0                       /LAST ADR, FROM KEYBOARD INPUT
03162   000004  NROTA   4                       /ROTATE COUNTER
03163   000000  LTST    0                       /LAST ADR, FOR SCOPE LOOPS
03164   000000  TTYW    0                       /TTYW THRU YY USED FOR KEYBOARD
03165   000000  TTYX    0                       /INPUT BIT SUPPRESSION
03166   000000  TTYY    0
03167   000000  SCW     0                       /TEMP STORAGE OF SUPPRESSED BITS
03170   777777  BITSUP  LAW     -1              /EACH SUPPRESSED BIT = 0
03171   000000  NXLOC   0                       /NEXT FIELD TO MOVE INTO
03172   000000  SOURCE  0                       /FIELD TO MOVE FROM
03173   000000  DESTN   0                       /FIELD TO MOVE TO
03174   000000  MOVES   0                       /SAVE AS MOVED
03175   000216  BGNLO   RTN1                    /EXIT ADR, TO A NEW FIELD
03176   000000  PINX    0                       /STORAGE OF MEM, BLOCK THAT PROG, IS IN
03177   000000  SXR     0                       /STORAGE LOG, FOR XR,
03200   000000  WORK    0
03201   000000  MEMADR  0
                        .EJECT
```

```
                        /
03202   000000  XMSK    0               /X-ADDRESS MASK
03203   000000  XCOM    0               /COUNTS X-LINES FOR PATTERN COMP,
03204   000001  K1      1
03205   000002  K2      2
03206   000007  K7      7
03207   000010  K10     10
03210   000020  K20     20
03211   000040  K40     40
03212   000077  K77     77
03213   000100  K100    100
03214   000200  K200    200
03215   000212  K212    212
03216   000215  K215    215
03217   000254  K254    254
03220   000260  K260    260
03221   000263  K263    263
03222   000270  K270    270
03223   000300  K300    300
03224   000331  K331    331
03225   000370  K370    370
03226   000377  K377    377
03227   000400  K400    400
03230   004000  K4K     4000
03231   007700  K7700   7700
03232   007777  K7777   7777
03233   010000  K10K    10000
03234   017777  K17S    17777
03235   020000  K20K    20000
03236   040000  K40K    40000
03237   070000  K70K    70000
03240   074000  K74K    74000
03241   300000  K300K   300000
03242   400000  K400K   400000
03243   177777  K177    177777
03244   370000  K370K   370000
03245   700000  K700K   700000
03246   770000  K770K   770000
                        /
                        .EJECT
```

```
                        /ERROR FLAG TABLE
                        /
03247   777700          MAXERR  LAW     -100
03250   003251          ERWRD   .+1
03251   760000                  LAW
03252   760000                  LAW
03253   760000                  LAW
03254   760000                  LAW
03255   760000                  LAW
03256   760000                  LAW
03257   760000                  LAW
03260   760000                  LAW
03261   760000                  LAW
03262   760000                  LAW
03263   760000                  LAW
03264   760000                  LAW
03265   760000                  LAW
03266   760000                  LAW
03267   760000                  LAW
03270   760000                  LAW
03271   760000                  LAW
03272   760000                  LAW
03273   760000                  LAW
03274   760000                  LAW
03275   760000                  LAW
03276   760000                  LAW
03277   760000                  LAW
03300   760000                  LAW
03301   760000                  LAW
03302   760000                  LAW
03303   760000                  LAW
03304   760000                  LAW
03305   760000                  LAW
03306   760000                  LAW
03307   760000                  LAW
03310   760000                  LAW
        000000                  .END
03311   770000  *L
            SIZE=03312
```

```
ADR      02720
ADRA     03160
ADRB     03161
ADRCW    03155
ADRX     02654
ADRXA    02706
ADR1     01067
ADR1P    02664
ADR2     01143
ADR2P    02665
ADXR     02657
AD1R     02655
AD2R     02660
AGAIN    01420
AJIN     02400
ALOK     02004
BAD1     03151
BDAT     02731
BDATX    02710
BEGIN    00200
BGNLO    03175
BITCON   03137
BITSUP   03170
BRSTA    00546
BURST    00524
BUST     00556
CBANK    00613
CBOTH    01236
CEND     00567
CFLG     02544
CKAL     00464
CKERR    02151
CKFCD    02317
CKLST    01227
CKNXT    02226
CKXY     00471
CK3      02057
CLMN     02645
CLX      735200
CMOVE    02125
CNROT    01626
CNTRL    03126
COMPL    00414
CREAD    00366
CREVR    01753
CRLF     02072
CT2      03141
CT24     03146
CT128    03145
CT16     03144
CT32     03142
CT4K     03143
DESTN    03173
DFST     01117
DIND     02421
```

```
DLMT     02652
DLMTA    02713
DLST     01171
DNMVE    02221
DOALL    00317
DOERR    00735
DON1     02656
DON2     02661
DON3     02672
DOXY     00537
EBA     707764
EEM     707702
ENERR    02675
ENOT     02525
EOM      01475
EOT      01467
EOTA     01501
EPA     707762
EQUAL    02176
ERROR    00666
ERSEL    02712
ERSET    00466
ERTBL    02674
ERWRD    03250
EXAM2    00275
EXAM3    00301
EXBA    707741
EXIT     03156
EXREL    00305
EXTST    00271
FADR     01106
FADR1    02762
FCOMV    02307
FIRST1   03147
FLAGS    03121
FRST     03060
GDAT     02725
GDATX    02707
GENPAT   00377
GETAD    01621
GOFL     02702
GOFLD    03101
GOLEFT   01645
GOOD1    03153
GOTO     02452
HALT     00653
INDY     01006
INSFLD   03122
JMP3     02615
KEYIN    01544
KRB     700312
KSF     700301
KYBRD    01034
K1       03204
K1P      03207
```

K10K	03233
K100	03213
K175	03234
K177	03243
K2	03205
K20	03210
K20K	03235
K200	03214
K212	03215
K215	03216
K254	03217
K260	03220
K263	03221
K270	03222
K300	03223
K300K	03241
K331	03224
K370	03225
K370K	03244
K377	03226
K4K	03230
K40	03211
K40K	03236
K400	03227
K400K	03242
K7	03206
K70K	03237
K700K	03245
K74K	03240
K77	03212
K770K	03246
K7700	03231
K7777	03232
LADR	01160
LADR1	02770
LAST	03123
LAST1	03150
LEGAL	01553
LEM	777704
LOCAT	01566
LOCER	02600
LSTA	03064
LTST	03163
MASK	02043
MAXERR	03247
MCAA	03124
MEMADR	03201
MOSCM	02357
MOVE	02343
MOVES	03174
MVRK	02304
MVERR	02551
NERR	02724
NETAK	02357
NOMO	03325

```
NOMOR   02626
NOPRNT  03120
NOSW    02470
NROTA   03162
NUMB    01517
NXLOC   03171
NXTBNK  00637
NXTHI   02274
NXTMV   02224
N64     00501
OCADR   03152
OKAS    01771
OVER    03157
OVRLAP  03002
OVRLP   02666
PATG    03133
PATN    03135
PATR    03132
PATWD   03134
PAX     721000
PCHAR   02027
PCW     03125
PCWA    03127
PCWB    03130
PCWC    03131
PCWR    02734
PCWX    02711
PHDR    02616
PINX    03176
PISIN   02667
PNXT    02034
POSITN  02113
PRNT    03154
PROCTL  02107
PROG    01373
PROIS   03034
PROR    03013
PRSEL   02430
PTO     02676
PTOI    03070
PTWLV   02701
PUT12   03107
QUERY   01564
RCOM    00446
READ    02424
RELOC   02703
ROTA    02677
ROTB    02700
ROTC    01647
ROTOR   01457
RTN1    02216
RTNL    02437
SBA     737761
SCP1    01331
SCA     03167
```

```
SETAC    02210
SETU1    00604
SETX     02673
SIMC     01306
SIMU     01271
SIM1     01302
SIM2     01323
SIM3     01352
SIXT4    03117
SLMTS    01655
SLMX     02671
SLTER    02744
SOURCE   03172
SPING    02101
STACS    03053
STER     00762
STLP     01215
STMV     02164
STNXT    02243
STOVER   00247
STSCP    01343
SUB1     02207
SUB2     02263
SUPBIT   01410
SUPR     02775
SUPX     02662
SUPXA    02663
SVADR    03140
SW0      00731
SW1      00723
SW2      00711
SXR      03177
TLMX     02672
TLS     700406
TNUM     03136
TSF     700401
TSLM     03045
TSNX     02653
TST      02714
TSTN     01364
TSTNC    01335
TSTNR    02756
TSTX     02725
TST1     00322
TST2     00336
TST3     00353
TTYW     03164
TTYX     03165
TTYY     03166
WCNT     00426
WHERE    00646
WONS     02531
WORK     03200
WOTIS    02607
WRITE    00417
```

```
XCOM    03203
XMSK    03202
XPRT    02332
Y64     00506
ZERO    01513
```

BEGIN	00202
RTN1	00216
STOVER	00247
EXTST	00271
EXAM2	00275
EXAM3	00301
EXREL	00305
DOALL	00317
TST1	00322
TST2	00336
TST3	00353
NETWK	00357
CREAD	00366
GENPAT	00377
WCNT	00406
COMPL	00414
WRITE	00417
READ	00424
RCOM	00446
CKAL	00464
ERSET	00466
CKXY	00471
N64	00501
Y64	00506
BURST	00524
WONS	00531
DOXY	00537
BRSTA	00546
BUST	00556
CEND	00567
SETU1	00604
CBANK	00613
NOMOR	00626
NXTBNK	00637
WHERE	00646
HALT	00653
ERROR	00666
SW2	00711
SW1	00723
SW0	00731
DOERR	00735
STER	00760
INDY	01006
KYBRD	01034
TSTNO	01035
TSTN	01064
ADR1	01067
FADR	01106
DFST	01117
ADR2	01143
LADR	01160
DLST	01171
STLP	01215
CKLST	01227
CRPTR	01236

```
DIND    02421
PRSEL   02430
RT1VL   02437
GOTO    02452
NOSW    02470
ENOT    02525
CFLG    02544
MVERR   02551
LOCER   02600
WOTIS   02607
JMP3    02615
PHDR    02616
CLMN    02645
DLMT    02652
TSNX    02653
ADRX    02654
AD1R    02655
DON1    02656
ADXR    02657
AD2R    02660
DON2    02661
SUPX    02662
SUPXA   02663
ADR1P   02664
ADR2P   02665
OVRLP   02666
PISIN   02667
TLMX    02670
SLMX    02671
DON3    02672
SETX    02673
ERTBL   02674
ENERR   02675
PTO     02676
ROTA    02677
ROTB    02700
PTWLV   02701
GOFL    02702
RELOC   02703
NERR    02704
TSTX    02705
ADRXA   02706
GDATX   02707
BDATX   02710
PCWX    02711
ERSEL   02712
DLMTA   02713
TST     02714
ADR     02720
GDAT    02725
BDAT    02731
PCWR    02734
SLTER   02744
TSTNR   02756
FADR1   02762
```

```
LADR1   02770
SUPR    02775
OVRLAP  03002
PROR    03013
NOMO    03025
PROIS   03034
TSLM    03045
STACS   03053
FRST    03060
LSTA    03064
PTOI    03070
GOFLD   03101
PUT12   03107
SIXT4   03117
NOPRNT  03120
FLAGS   03121
INSFLD  03122
LAST    03123
MCWA    03124
PCW     03125
CNTRL   03126
PCWA    03127
PCWB    03130
PCWC    03131
PATR    03132
PATG    03133
PATWD   03134
PATN    03135
TNUM    03136
BITCON  03137
SVADR   03140
CT02    03141
CT32    03142
CT4K    03143
CT16    03144
CT128   03145
CT24    03146
FIRST1  03147
LAST1   03150
BAD1    03151
OCADR   03152
GOOD1   03153
PRNT    03154
ADRCW   03155
EXIT    03156
OVER    03157
ADRA    03160
ADRB    03161
XROTA   03162
LTST    03163
TTYW    03164
TTYX    03165
TTYY    03166
SCW     03167
BITSUP  03170
```

```
MXLOC    03171
SOURCE   03172
DESTN    03173
MOVES    03174
BGNLO    03175
PINX     03176
SXR      03177
WORK     03200
MEMADR   03201
XMSK     03202
XCOM     03203
K1       03204
K2       03205
K7       03206
K10      03207
K20      03210
K40      03211
K77      03212
K100     03213
K200     03214
K212     03215
K215     03216
K254     03217
K260     03220
K263     03221
K270     03222
K302     03223
K331     03224
K370     03225
K377     03226
K400     03227
K4K      03230
K7700    03231
K7777    03232
K10K     03233
K17S     03234
K20K     03235
K40K     03236
K72K     03237
K74K     03240
K300K    03241
K400K    03242
K177     03243
K37EK    03244
K722K    03245
K774K    03246
MAXERR   03247
ERARG    03250
KSF      700301
KRB      700312
TSF      700401
TLS      700406
EEM      707702
LEM      707704
EXRA     707741
```

```
SBA     707761
EPA     707762
EBA     707764
PAX     721000
CLX     735000
```