PDP-15 26
MEMORY

IDENTIFICATION

PRODUCT CODE:	MAINDEC-15-D1BC-D
PRODUCT NAME:	PDP-15 Extended Memory Checkerboard (MXCH 15)
DATE CREATED:	April 1, 197Ø
MAINTAINER:	Diagnostic Group
AUTHOR:	John W. Richardson D. K. Macomber

18

1. ABSTRACT

The PDP-15 Extended Memory Checkerboard test verifies the operational status of core memory by testing for core failure on halt-selected lines under worst case noise conditions. The program tests any memory configuration of from 8K to 128K words, in 4K segments. Seven patterns are used for testing, and these may be selected individually by the operator. The program relocates automatically in order to test all memory fields from each field.

2. REQUIREMENTS

2.1 Equipment

A standard PDP-15 equipped with 8K to 128K words of core memory.

2.2 Storage

The program occupies approximately 35ØØ octal locations of any 4K memory field.

3. LOADING PROCEDURE

The program may be loaded into any low order 4K field (ØØØØ to 7777 octal) of memory block Ø (fields ØØ thru Ø7). Normally load the program into memory field ØØ, as follows:

a. The tape supplied is punched in the ABS mode.

b. Place the tape in the reader.

c. Set the ADDRESS switches to Ø177ØØ; the BANK MODE switch to a 1.

d. Press RESET and then READ-IN.

4. STARTING PROCEDURE

4.1 Starting Address

The starting address is ØØØ2ØØ.

4.2 Restarting Address

Restart from ØØØ215 (or FFØ215) to retain previous program conditions.

Restart from ØØØ2ØØ (or FFØ2ØØ) to setup the test limits and ACS and to reinitialize the program.

FF = Field Number.

4.3 Operator Action

After starting from 2ØØ, the program will print "TEST LIMITS". The operator must ehn specify via the Teletype keyboard the amount of core memory to test, followed by a carriage return. The program assumes 4K fields numbered ØØ thru 37, octal. This is the Field Number. Memory Block Ø contains 4K Fields Numbered ØØ thru Ø7; Memory Block 1, 1Ø thru 17; Memory Block 2, 2Ø thru 27; Memory Block 3, 3Ø thru 37.

4.3.1 Specify the Test Limits

a. Type two field numbers, separating the field numbers with a comma, and then a carriage return.

b. The first number typed signifies the first 4K field to test, and the secon number the last 4K field to test.

c. The program will begin testing with the lowest order 4K field to test and will test all consecutive fields up to an including the highest specified.

d. The 4K field containing the program may be included. It will be tested after program relacation takes place. Program relocation is described in Section 5.1.1.

e. If an error is made during typing press the RUBOUT key. "TEST LIMITS" will be printed again. Previous input is ignored.

f. The highest 4K field to be tested may be typed first. The program will reverse the two numbers so as to make the first number the last field to test.

g. Any single field or any two or more consecutive fields may be specified.

For the following examples assume that the program is in field ØØ (ØØØØØØ to ØØ7777), and the PDP-15 being used is equipped with 128K of core memory.

Example A:

TEST LIMITS

ØØ,Ø7↓ (↓) denotes carriage return

The program will test all 8 fields of memory block Ø.

Example B:

TEST LIMITS

Ø7,ØØ↓

The program will perform exactly as Example A.

Example C:

TEST LIMITS

23,23↓

Field 23 will be tested alone. Field 23 is locations 23ØØØØ to 237777 of memory block 2.

Example D:

TEST LIMITS

16, 21↓

Locations 16ØØØØ through 217777 (fields 16, 17, 2Ø and 21 will be tested.

Example E:

TEST LIMITS

ØØ,ØØ↓ PROGRAM IS FIELD ØØ

ØØ,Ø1↓

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

Operation of the program is unpredicatable if the amount of core memory selected for testing exceeds the actual amount available, i.e., selecting 32K for testing on a PDP-15 equipped with a maximum of 128K, or if the program is loaded into any memory block except Ø.

4.3.2 Setup ACS -

After the test limits are specified, the program will print "SETUP ACS". For normal program operation the ACS must be set to ØØØØØØ octal. Press any key on the Teletype keyboard after setting the ACS to all Ø's. The program will then run until stopped by the operator. Normal program operation is defined as performing all eight checkerboard patterns on all of available memory from every 4K memory field.

5. OPERATING PROCEDURE

5.1 Program and Operator Action

a. Load the program into memory block Ø as described in Section 3.

b. Specify the test limits via keyboard as described in Section 4.3.1.

c. The message "SETUP ACS" will be printed. Set the ACS to ØØØØØØ octal, and press any keyboard key.

d. The program will write one pattern in one 4K field, after which, each bit of address Ø is read and tested.

e. Repeat step d on each location in the 4K field.

f. Setup for the next 4K field and repeat the read and test sequence in steps d through f.

When all fields have been tested in this manner, the next pattern is written, and steps d through f repeated.

When all four tests have been executed, on all of memory, the program then relocates and performs all four tests again.

5.1.1 Program Relocation

Program relocation depends upon the amount of core memory being tested. Relocation is always within the group of 4K fields selected for testing, and under certain conditions the program may not relocate at all, but will remain in the current field to perform the tests (see below). The program normally first relocates to the highest order 4K field under test. From there it relocates to the next lower 4K field, after performing all tests. The program keeps relocating to the next lower 4K field until it reaches the lowest order 4K field under test. The testing and relocation cycle is then

repeated. This procedure is repeated until stopped by the operator with ACS Ø. As an example, if the program is initially in field ØØ, and 128K of memory is selected, the program would relocate from ØØ, to field 37, then to field 36, 35, 34, 33, 32, 31, 3Ø, 27, 26, etc. in that order. The program does not relocate to any field which is not included in the test limits. If fields 14, 15, and 16 were selected, relocation would be from ØØ to 16, then 15 and 14. Fields ØØ through 13 and field 17 through 37 would not contain the program again until included in the test limits.

The program will <u>not</u> relocate if any of the conditions described below exist:

a. A forced relocation has been made (Section 8.2.7).
b. Only one 4K field is selected for testing.
c. An error was detected in all of the available 4K fields under test.
d. ACS 9 is on a 1 to inhibit program relocation (Section 8.2.5).

The location of the program is indicated by the message "PROGRAM IS IN FIELD FF", where FF is the field number. This message occurs immediately after each program relocation. The message printout may be deleted by placing ACS 11 on a 1 at any time. The printout will resume when ACS 11 is placed on a Ø.

The program provides a degree of protection for itself by not relocating to any field which has an error. The field number in error is saved, and is compared to the destination field number before relocation takes place. If equal, the next lower field is set up as the destination providing it has

no error. The first field ound to be error-free is set up as the destination. Relocation will not take place if all fields have shown errors. The program will resume relocating to a field whenever the error condition does not exist. During the relocation process the program tests each data word transferred to the new field by performing the transfer, reading the word back and comparing the word with the correct data in the current field. This is done on a one for one basis until the process is completed. The entire 4K field is moved to enable loaders or any other data to be carried with the program. If an error is found during relocation, the address is error, and the "good" and "bad" data words are printed. The error printout format is described in Section 6.

6. ERRORS

6.1 Error Printouts and Description

Immediately after the first error is detected, the header shown below is printed. The header is not printed again until restarting from 2ØØ or 215.

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

Where:

TEST =The current test which detected the error.

OCTAL ADR =The octal address which contains the data in error.

GOOD =What the data should have been in that address. This will always equal ØØØØØØ or 777777 octal.

BAD =The data as read from that address. This will always contain one or more bits which are the complement of those shown under GOOD.

PAT CONTROL WORD =The control word used by the current test to generate the checkerboard pattern. This will be 463144 or 314633 for test 1; 63146Ø or 146317 for test 2; 52525Ø or 252527 for test 3.

TEST	OCTAL ADR	GOOD	BAD	PAT CONTROL WORD
1	Ø14ØØØ	ØØØØØØ	ØØØØØ1	463144
1	Ø6Ø2ØØ	777777	767777	314633
3	Ø14ØØØ	ØØØØØØ	ØØØØØ1	52525Ø
4	Ø37555	777777	377777	

In the above example, errors were detected by tests 1, 3 and 4 in memory field Ø1, Ø6, and Ø3. Test 1 detected a "picked up" bit at location 4ØØØ, field 1, and a dropped bit at location 2ØØ, field 6. Test 3 detected the same error as test 1 at location 4ØØØ, field 1, and test 4 detected a bit Ø error at location 7555 in field 3.

After each error printout, the program continues testing with the next sequential memory location.

Three AC switches may be used to control the error printouts. Placing ACS Ø on a 1 during the printout will cause a program halt after completion of printing. ACS 1 on a 1 will inhibit the printout and cause a program halt. Press CONTINUE to receive the error printout and to continue testing. ACS 2 on a 1 will inhibit printout and ring the TTY BELL for each error. The use of these switches is described in Section 8.2 in more detail.

6.1.1 Program Relocation Error-

This message will be printed upon detection of a relocation error. The error information will immediately follow as in the example below. After all errors have been printed the message "NO MORE ERRORS" is printed, and the program will then set up to relocate to the next lower field if one is available.

EXAMPLE:

```
TEST    OCTAL ADR     GOOD     BAD     PAT CONTROL WORD
PROGRAM RELOCATION ERROR
        Ø31ØØØ        741ØØØ   74ØØØØ
        Ø31ØØ1        611ØØ5   6Ø1ØØ5
        Ø31ØØ2        76Ø2Ø7   76ØØØ7
NO MORE ERRORS
```

The above example shows three consecutive errors during program relocation to field Ø3. Field Ø2 would be set up for relocation. Location 1ØØØ in field 3 should have contained a SKP instruction, but bit 11 was dropped during the transfer. Bit 5 was dropped in the JMP instruction in 1ØØ1. and bit 1Ø dropped in the LAW instruction in 1ØØ2.

6.1.2 Printouts Inhibited

This message is printed whenever 64 (decimal) consecutive printouts have occurred. Error printouts will be inhibited until after all four tests have been run eight times, after which the error printouts will resume 64 more printouts. This feature is not used with program relocation errors. This feature is included to prevent lengthy error printouts when the program is being run for an extended period of time unattended. Error printouts may be resumed by restarting the program from location 2ØØ.

6.1.3 Program is in Field FF

WHERE "FF" IS A FIELD NUMBER. This message is printed if one of the following conditions exist:

a. The operator has specified a single field for testing and that field contains the program. Select another field, refer to Section 4.3.

b. After every program relocation.

6.1.4 Error in Selected Field

This message is printed when a forced program relocation is attempted and the program has previously detected a data error in that field. Type a new field number, or press carriage return to resume automatic program relocation. See Section 8.2.7 for instructions to force the program to another field.

6.1.5 First/Last ADR is Within Program

The operator has specified the first or last address, as indicated by the printout, which is within the program area. Retype a new address. See Section 8.2.8 for setting up scope loops via keyboard.

7. RESTRICTIONS

7.1 Starting Restrictions

Start from FFØ2ØØ to set up the test limits and ACS and to reinitialize the program.

Start from FFØ215 to retain the present program conditions.

(FF = the field the program is in).

7.2 Operating Restrictions

Don't use the STOP key to halt the program. Place ACS Ø on a 1.

8. MISCELLANEOUS

8.1 Execution Time

The time required to run all four tests on one 4K field is approximately 15 seconds.

The time required to run all four tests in 37 (octal) 4K fields is approximately 4 hours, 2Ø minutes.

The above times are based on a 8ØØ ns cycle time.

8.2 Applications

To give the operator control of the program, the ACS are assigned unique functions. The ACS assignments and their effect on the program are described below. Please note that it is important that the program be halted with ACS Ø rather than the STOP key. Using the STOP key may result in a halt while the program is relocating. The operation may not be initiated immediately since most of the ACS are sensed only after all tests have been performed.

8.2.1 Halt After Test or Error Printout - ACS Ø

Placing ACS Ø on a 1 at any time while the program is running will cuase a halt after the current test is completed on one 4K field. The MO will = 652. The ACS may then be changed if desired. Press CONTINUE to recover. If no ACS changes the program will resume the test which was interrupted. If ACS changes were made the new setting are stored and executed. Raising ACS Ø during an error printout will cause a halt at the same location mentioned above, after the printout.

8.2.2 Delete Error Printout and Halt on Error - ACS 1

Raising ACS 1 at any time causes all data error printouts to be inhibited. A halt will occur with the MO = 654 if an error occurs. Press CONTINUE to receive the error printout and to resume testing. ACS changes may be made.

8.2.3 Bell on Error - ACS 2

ACS 2 on a 1 causes the program to ring the TTY BELL whenevery an error occurs.

This is convenient when attempting to isolate an intermittant error. ACS 1 has no effect if ACS 2 and 1 should both happen to be on a 1. If ACS Ø and 2 are 1, a halt occurs after the bell. Proceed as described in Section 8.2.1.

8.2.4 Test Selection - ACS 3 through 6

Any one, or any combination of tests may be executed by setting any combination of ACS 3 through 6 to a 1. ACS 3 specifies tests 1; ACS 4, test 2; ACS 5, test 3; ACS 6, test 4. The test specified by the most significant ACS will be performed first.

If all four ACS are Ø, all four tests are performed in sequence. The ACS may be changed while the program is running. The new test will be recognized after the last of the current selection is performed.

8.2.5 Inhibit Program Relocation - ACS 9

The program normally relocates utomatically as indicated by the PC or MO indicators. To retain the program in its current 4K field, place ACS 9 on a 1 at any time.

8.2.6 Inhibit "PROGRAM IS IN FIELD" - ACS 11

The program normally prints the field number containing the program immediately after each relocation. The message may be suppressed by placing ACS 11 on a 1 at any time. To resume the printout place ACS 11 on a Ø. This switch does not inhibit the message printout when an operator error is made.

8.2.7 Program Relocation - ACS 12

The operator may relocate the program to any 4K field by specifying a forced relocation with ACS 12 on a 1. Use the following procedure.

a. Halt the program with ACS Ø.

b. Place ACS 12 on a 1 and ACS Ø on a Ø. Press CONTINUE.

c. A printout will occur which instructs the operator to place ACS 13 on a Ø. The program will loop until this is done.

d. With ACS 12 on a Ø the message GO TO FIELD is printed followed by the program waiting for a field number.

e. Type the desired field number (ØØ through 37).

f. Relocation is done immediately, and the program is executed in the new field.

The program will not relocate again until restarted from 2ØØ, or in step d above, press carriage return to resume automatic relocation.

If a data error was previously detected in the new 4K field, the message "ERROR IN SELECTED 4K" is printed, followed by step d repeated. Type another field number, or carriage return to resume normal operation.

Each word transferred to the new field is tested in the same manner as described in Section 5.1.1, Program Relocation. Printouts occur for each relocation error. Step d will be repeated after all error reporting is done. Type another field number, or carriage return to resume normal operation.

At times, the program will automatically restart at 2ØØ and print TEST LIMITS. This will occur whenever a single field has been selected for testing, and the operator relocates

the program to that field. New test limits must be specified since the program cannot run the tests on its own 4K field. Proceed as described in Section 4.3.

8.2.8 Request Scope Loop - ACS 13

The operator ma- initiate, via keyboard, any single or any group of consecutive locations up to 4K for the program to loop on. Any of the four data patterns may also be requested. No error checking is done in the loop. The loop reads a location, complements the data and rewrites into the same location. Each location within the limits specified is treated likewise. The limits specified are looped until halted with the STOP key. The program must be restarted from 2ØØ to resume normal operation.

Initiate the loop with the following procedure:

a. Halt the program with ACS Ø.

b. Place ACS 13 on a 1 and press CONTINUE.

c. The message "TEST" will be printed. The program now wiats for a selection by the operator.

d. Type the desired test number. Either 1, 2, 3, or 4. An automatic carriage return follows.

e. The message "FIRST ADR" is printed.

f. Type the first address of the group to be looped by the program. This number must be a 6 digit octal number. An automatic carriage return follows.

g. The message "LAST ADR" is printed. Type a 6 digit octal address to indicate the last address of the group. The scope loop is immediately entered after the last address is typed.

The loop may also be initiated by restarting from 2ØØ or 125 and placing ACS 13 on a 1 under "SETUP ACS".

If a typing error is made press the RUBOUT key. A ? will be printed, and the input routine will restart with "TEST".

Example A:

```
TEST 3
FIRST ADR Ø1Ø1ØØ
LAST ADR Ø1Ø1ØØ
```

Address 1ØØ in field 1 will be looped continuously after checkerboard pattern number 3 is written in the entire 4K field.

Example B:

```
TEST 2
FIRST ADR Ø2ØØØØ
LAST ADR Ø27777
```

The entire 4K number 2 will be looped after pattern number 2 is written.

Example C:

```
TEST 2
FIRST ADR Ø27777
LAST ADR Ø2ØØØØ
```

The input routine will reverse the two addresses and perform exactly as example B.

Example D:

```
TEST 4
FIRST ADR ØØØ1ØØ
FIRST ADR IS WITHIN PROGRAM
FIRST ADR Ø1Ø1ØØ
LAST ADR Ø1Ø2ØØ
```

Example D shows the message printed when an address is selected which is in the field containing the program. In this case, field Ø.

Example E:

```
TEST 5
?
TEST 4
FIRST ADR Ø7ØØØØ
LAST ADR Ø74ØØØ
```

In example E the operator typed an invalid test number. The program printed the question mark, and restarted with TEST. The instructions in the body of the scope loop appear below.

```
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
```

Note that one 4K field is the maximum that may be looped by the program. If 4K field boundaries are overlapped, the checkboard pattern will be written in the field specified by the FIRST ADR. The scope, however, will reference the locations specified.

8.2.9 Bit Suppression - ACS 14

Excessive error printouts due to one or more bits in error may be suppressed by specifying the bit positions via keyboard input. The bit positions will still contain the checkboard patter, and will be tested. Error printout will occur for any bit position not selected.

Use the following procedure:

a. Halt the program with ACS Ø.

b. Place ACD 14 on a 1, and press CONTINUE.

c. The message "SUPPRESS" will be printed and the program waits for inputs.

d. Place ACS 14 on a Ø.

e. Type in decimal, the desired bit position to be suppressed (Ø through 17).

f. Press the carriage return key.

Error printouts for that position alone will not occur.

To suppress more than one bit position:

a. Proceed as above, but separate the selected bit positions with a comma in step (e). As an example, to suppress bits Ø, 8 and 17 type Ø, 8, 17).

Press RUBOUT and the CARRIAGE RETURN to continue with error printouts of all bit positions.

Each time the bit suppressing routine is entered with ACS 14, the previously selected positions must be retyped if suppression is wanted.

The program is not effected in any way by the typing of letters, or numbers greater than 17. The resulting positions suppressed would be unpredictable.

9. PROGRAM DESCRIPTION

The program is designed to run worst case checkerboard patterns for 3K memory stacks.

A minimum of 8K of core memory is required, and a maximum of 128K words may be tested. The program automatically relocates from 4K field to 4K field to test all of memory from each field. The patterns are shown below as they would appear in a portion of one bit plane. Pattern number 3 is considered to be the worst case pattern of most PDP-15 memory stacks. The patterns are complemented along the X axis every 4Ø octal addresses. The X axis is addressed by bits 12 through 17, and the Y axis by bits 6 through 11.

Test 1:

	Y axis		Y axis
Ø	1ØØ11ØØ1	Ø	Ø11ØØ11Ø
X axis	1ØØ11ØØ1	X axis	Ø11ØØ11Ø
	.		.
	.		.
	.		.
4Ø	Ø11ØØ11Ø	4Ø	1ØØ11ØØ1
	Ø11ØØ11Ø		1ØØ11ØØ1

Test 2:

	Y axis		Y axis
Ø	11ØØ11ØØ	Ø	ØØ11ØØ11
X axis	11ØØ11ØØ	X axis	ØØ11ØØ11
	.		.
	.		.
	.		.
4Ø	ØØ11ØØ11		11ØØ11ØØ
	ØØ11ØØ11		11ØØ11ØØ

Test 3:

	Y axis		Y axis
Ø	1Ø1Ø1Ø1Ø	0	Ø1Ø1Ø1Ø1
X axis	1Ø1Ø1Ø1Ø	X axis	Ø1Ø1Ø1Ø1
	.		.
	.		.
	.		.
4Ø	Ø1Ø1Ø1Ø1		1Ø1Ø1Ø1Ø
	Ø1Ø1Ø1Ø1		1Ø1Ø1Ø1Ø

Test 4:

This test writes all ones into a memory field. One X axis line is then cleared to Ø (all 18 bit planes), and then read 1Ø24 times. All intersecting Y lines are then addressed and checked to make sure the contents did not change. All 1ØØ (octal) X lines are tested in this manner.

The operator is given a degree of control over the program with the AC switches. The operator may halt the program (Ø); inhibit error printouts and halt on error (1); substitute the Teletype bell for error indication (2); select any one or a combination of four test patterns (3 through 6), inhibit program relocation (9), relocate the program to any 4K field (12); setup a scope loop via keyboard input (13); and inhibit error printouts for one or more bit positions (14).

10. LISTING

```
/
/PDP-15 EXTENDED MEMORY CHECKERBOARD,
/8K MINIMUM CORE REQUIRED, S.A. = 200,
/
/COPYRIGHT 1970, DIGITAL EQUIPMENT CORP.,
/MAYNARD, MASS, 01754
/
/J. RICHARDSON
/D. MACOMBER
/
```

```
                                .TITLE MXCH15
                                .ABS
                        /
00001                           .LOC    1
                        /
00001   600001                 JMP     1
00002   777777                 LAW     -1
00003   777777                 LAW     -1
00004   777777                 LAW     -1
00005   777777                 LAW     -1
                        /
        700406          TLS=700406
        700401          TSF=700401
        700301          KSF=700301
        700312          KRB=700312
        721000          PAX=721000
        707764          EBA=707764
        707762          EPA=707762
        707761          SBA=707761
        707741          EXBA=707741
        735000          CLX=735000
                        /
                                .EJECT
```

```
00200                          .LOC    200
                       /
00200   707762   BEGIN  EPA
00201   143124         DZM     FLAGS           /CLEAR PROGRAM FLAGS
00202   143201         DZM     PINX
00203   100653         JMS     WHERE           /SEE WHERE PROGRAM IS
00204   043125         DAC     INSFLD          /SAVE FIELD NO.
00205   101660         JMS     SLMTS           /SETUP TEST LIMITS
00206   102013         JMS     SETAC           /SETUP ACS
00207   777777         LAW     -1
00210   043173         DAC     BITSUP          /MASK FOR BIT SUPPRESSION
00211   777700         LAW     -100
00212   043250         DAC     MAXERR          /COUNT FOR TOTAL ERRORS
00213   777770         LAW     -10
00214   043122         DAC     SIXT4           /PASS COUNTER
00215   143123         DZM     NOPRNT
00216   201623   RTN1  LAC     GETAD-1
00217   041575         DAC     LOCAT+4
00220   707762         EPA
00221   101571         JMS     LOCAT           /TELL WHERE PROGRAM IS
00222   142621         DZM     PHDR
00223   203153         LAC     LAST1           /FIRST FIELD TO TEST
00224   543152         SAD     FIRST1          /LAST TO TEST
00225   741000         SKP
00226   600231         JMP     .+3
00227   543125         SAD     INSFLD          /INSFLD = FIELD WITH PROGRAM
00230   600202         JMP     BEGIN+2
00231   202677         LAC     ERTBL           /ERROR TABLE POINTER
00232   043251         DAC     ERWRD
00233   777740         LAW     -40
00234   043147         DAC     CT16
00235   760000         LAW
00236   063251         DAC*    ERWRD           /LAW = NO ERROR IN TABLE
00237   443251         ISZ     ERWRD
00240   443147         ISZ     CT16
00241   600236         JMP     .-3
                       /
00242   043126         DAC     LAST            /NO ERROR IN LAST
00243   100653         JMS     WHERE
00244   043125         DAC     INSFLD
00245   202677         LAC     ERTBL           /ERROR TABLE POINTER
00246   043251         DAC     ERWRD
                       /
                               .EJECT
```

```
                        /RETURN TO STOVER AFTER ANY ACS CHANGES
                        /WHILE RUNNING
                        /
00247   750004  STOVER  LAS                     /READ TEST PARAMETERS
00250   503244          AND     K177
00251   043127          DAC     MCWA
00252   503212          AND     K40
00253   744200          SZA!CLL                 /BIT 12 A 1 = FORCE RELOCATE
00254   602312          JMP     FCDMV           /RELOCATE
00255   750004          LAS
00256   503211          AND     K20
00257   740200          SZA                     /BIT 13 A 1 = KEYBOARD INPUT
00260   601041          JMP     KYBRD           /WAIT FOR INPUT
00261   750004          LAS
00262   503210          AND     K10
00263   740200          SZA                     /BIT 14 A 1 = BIT SUPPRESSION
00264   101413          JMS     SUPBIT
00265   203127          LAC     MCWA            /PARAMETERS
00266   503241          AND     K74K            /MASK BITS 3 TO 6
00267   741200          SNA                     /ALL 0 = DO ALL TESTS
00270   600323          JMP     DOALL
                        /
                        .EJECT
```

```
                        /
                        /EXAMINE TEST SWITCHES 3 TO 6
                        /
00271   203127  EXTST   LAC     MCWA
00272   503237          AND     K40K
00273   740200          SZA                     /BIT 3 A 1 = TEST 1
00274   600327          JMP     TST1
00275   203236  EXAM2   LAC     K20K
00276   503127          AND     MCWA
00277   740200          SZA                     /BIT 4 A 1 = TEST 2
00300   600340          JMP     TST2
00301   203234  EXAM3   LAC     K10K
00302   503127          AND     MCWA
00303   740200          SZA                     /BIT 5 A 1 = TEST 3
00304   600351          JMP     TST3
00305   203231  EXAM4   LAC     K4K
00306   503127          AND     MCWA
00307   740200          SZA                     /BIT 6 A 1 = TEST 4
00310   600362          JMP     TST4
00311   443122          ISZ     SIXT4           /64 PASSES IF SKIP
00312   600316          JMP     .+4
00313   143123          DZM     NOPRNT          /CLEAR NO PRINT FLAG
00314   777770          LAW     -10             /RESTORE COUNT
00315   043122          DAC     SIXT4
00316   750004          LAS
00317   503230          AND     K400
00320   740200          SZA                     /BIT 9 A 1 = DON'T MOVE
00321   600216          JMP     RTN1
00322   602130          JMP     CMOVE           /DONE ALL TESTS, SETUP
                                                /FOR RELOCATION
                        /
                        /SETUP TO RUN ALL TESTS
                        /
00323   203127  DOALL   LAC     MCWA
00324   243241          XOR     K74K            /SET ALL TEST BITS
00325   043127          DAC     MCWA            /RESTORE
00326   600327          JMP     TST1            /TEST 1
                        /
                        /TEST 1, WRITE CHECKER PATTERN #1
                        /
00327   203132  TST1    LAC     PCWA            /TEST 1 PAT, CONTROL WORD
00330   043130          DAC     PCW
00331   043131          DAC     CNTRL
00332   760261          LAW     261             /ASCII 1
00333   043141          DAC     TNUM            /TEST NUMBER
00334   100366          JMS     NETWK           /GO WRITE IN ALL FIELDS
00335   100377          JMS     CREAD           /NOW GO READ AND TEST
00336   600275          JMP     EXAM2           /SEE IF TEST 2 WANTED
00337   600334          JMP     .-3             /DO COMPLEMENT
                        /
                        .EJECT
```

```
                        /TEST 2, WRITE CHECKER PATTERN #2
                        /
00340   203133  TST2    LAC     PCWB            /TEST 2 PAT, CONTROL WORD
00341   043130          DAC     PCW
00342   043131          DAC     CNTRL
00343   760262          LAW     262             /ASCII 2
00344   043141          DAC     TNUM            /TEST NUMBER
00345   100366          JMS     NETWK           /WRITE IN ALL FIELDS
00346   100377          JMS     CREAD           /READ AND TEST EACH FIELD
00347   600301          JMP     EXAM3           /SEE IF TEST 3 WANTED
00350   600345          JMP     .+3             /DO COMPLEMENT
                        /
                        /TEST 3, WRITE CHECKER PATTERN #3,
                        /
00351   203134  TST3    LAC     PCWC            /TEST 3 PAT, CONTROL WORD
00352   043130          DAC     PCW
00353   043131          DAC     CNTRL
00354   760263          LAW     263             /ASCII 3
00355   043141          DAC     TNUM            /TEST NUMBER
00356   100366          JMS     NETWK           /WRITE IN ALL FIELDS
00357   100377          JMS     CREAD           /READ AND TEST EACH FIELD
00360   600305          JMP     EXAM4           /SEE IF TEST 4 SELECTED
00361   600356          JMP     .+3             /DO COMPLEMENT
                        /
                        /TEST 4, TEST ALL XY COORDINATES
                        /
00362   760264  TST4    LAW     264             /ASCII 4
00363   043141          DAC     TNUM            /TEST NUMBER
00364   100525          JMS     BURST           /WRITE IN ALL FIELDS
00365   600311          JMP     EXAM4+4         /PREPARE TO RELOCATE
                        /
                        /ROUTINE TO SETUP ADDRESSES FOR WRITE LOOP
                        /
00366   000000  NETWK   0
00367   100610          JMS     SETU1           /SETUP 1ST FIELD TO TEST
00370   100617          JMS     CBANK           /SEE IF IT HAS PROGRAM
00371   741000          SKP                     /NO
00372   620366          JMP*    NETWK           /EXIT
00373   100430          JMS     WRITE           /ACTUALLY WRITE ONE FIELD
00374   100643          JMS     NXTBNK          /SETUP FOR NEXT FIELD
00375   600370          JMP     NETWK+2         /SEE IF IT HAS PROGRAM
00376   620366          JMP*    NETWK           /WROTE ALL, EXIT
                        /
                        /ROUTINE TO SETUP ADDRESSES FOR READ LOOP
                        /
00377   000000  CREAD   0
00400   100435          JMS     READ            /ACTUALLY READ AND TEST 1 FIELD
00401   777774          LAW     -4
00402   243130          XOR     PCW             /AC=COMPLEMENT OF PCW
00403   543131          SAD     CNTRL           /ALL DONE IF EQUAL
00404   620377          JMP*    CREAD           /EXIT
00405   043131          DAC     CNTRL           /CNTRL=COMPLEMENT PATTERN
00406   440377          ISZ     CREAD           /RETURN+1
00407   620377          JMP*    CREAD           /EXIT AND WRITE COMPLEMENT
                        .EJECT
```

```
                        /
                        /PATTERN ROUTINE FOR TESTS 1 THRU 3
                        /
00410   000000  GENPAT  0
00411   203131          LAC     CNTRL           /CURRENT PATTERN CONTROL WORD
00412   043140          DAC     PATN            /SAVE
00413   777700          LAW     -100
00414   043151          DAC     CT04            /COUNTS Y AXIS
00415   203140          LAC     PATN            /CONTROL WORD
00416   043135          DAC     PATR            /SAVE
00417   777760  WCNT    LAW     -20
00420   043147          DAC     CT16            /COUNTS 16 SHIFTS
00421   203135          LAC     PATR
00422   744010          RCL
00423   043135          DAC     PATR
00424   751400          SELICLA                 /NO SKIP SAYS WRITE 777777
00425   740001  COMPL   CMA                     /AC = 7777
00426   043156          DAC     GOOD1           /SAVE
00427   620410          JMP*    GENPAT          /EXIT TO READ OR WRITE
                        /
                        /WRITE ROUTINE FOR TESTS 1 THRU 3
                        /
00430   000000  WRITE   0
00431   100410          JMS     GENPAT          /GET A WORD
00432   050000          DAC     X               /WRITE
00433   100472          JMS     CKXY            /CHECK FOR PATTERN INVERSION
00434   620430          JMP*    WRITE           /DONE 4K
                        .EJECT
```

```
                        /
                        /READ AND TEST ROUTINE FOR TESTS 1 THRU 3
                        /
00435   000000  READ    0
00436   100610          JMS     SETU1           /SETUP FOR FIRST FIELD
00437   100617          JMS     CBANK           /SEE IF IT HAS PROGRAM
00440   741000          SKP                     /NO
00441   620435          JMP*    READ            /NO MORE CORE TO READ
00442   100410          JMS     GENPAT          /GET A WORD
00443   203205          LAC     K1
00444   043142          DAC     BITCON          /USED FOR BIT INVERSION
00445   203142  RCOM    LAC     BITCON
00446   250000          XOR     X               /COMPLEMENT A BIT
00447   050000          DAC     X               /WRITE
00450   203142          LAC     BITCON
00451   250000          XOR     X               /RE-COMPLEMENT
00452   050000          DAC     X               /RE-WRITE
00453   210000          LAC     X               /READ
00454   543156          SAD     GOOD1           /COMPARE
00455   741000          SKP                     /OK
00456   600467          JMP     ERSET           /PRINT INFO
00457   203142          LAC     BITCON
00460   744010          RCL                     /SETUP FOR NEXT BIT
00461   740400          SNL                     /DONE 18 IF LINK = 1
00462   600444          JMP     RCOM-1          /DO NEXT BIT POSITION
00463   100472  CKAL    JMS     CKXY            /CHECK FOR PATTERN INVERSION
00464   100643          JMS     NXTBNK          /SETUP FOR NEXT FIELD
00465   600437          JMP     READ+2          /READ NEXT FIELD
00466   620435          JMP*    READ            /WRITE NEXT PATTERN
00467   043154  ERSET   DAC     BAD1
00470   100673          JMS     ERROR           /PRINT INFO
00471   600463          JMP     CKAL
                        /
                        .EJECT
```

```
                        /
                        /ROUTINE TO CHECK FOR PATTERN INVERSION
                        /
00472   000000  CKXY    0
00473   443146          ISZ     CT4K            /DONE 4K IF SKIP
00474   741000          SKP
00475   620472          JMP*    CKXY            /EXIT TO WRITE OR READ
00476   443151          ISZ     CT04            /DONE WITH Y AXIS IF SKP
00477   741000          SKP
00500   600507          JMP     Y64             /DONE 64 Y LINES
00501   724000          PXA
00502   343214  N64     TAD     K100            /INCREMENT Y ADDRESS BY 1
00503   721000          PAX
00504   443147          ISZ     CT16            /CHECK FOR 16 LOCATIONS
00505   600421          JMP     WCNT+2          /NOT YET
00506   600415          JMP     WCNT-2          /RESTORE COUNT
                        /
00507   737001  Y64     AXR+1                   /INCREMENT X LINE BY 1
00510   777700          LAW     -100
00511   043151          DAC     CT04            /RESTORE Y LINE COUNTER
00512   724000          PXA
00513   503213          AND     K77
00514   543212          SAD     K40             /COMPLEMENT PATTERN IF EQUAL
00515   600521          JMP     .+4
00516   724000          PXA
00517   343312          TAD     (770000
00520   600502          JMP     N64             /START WITH NEW X-Y COMBO
00521   203140          LAC     PATN            /PATTERN CONTROL WORD
00522   740001          CMA
00523   043140          DAC     PATN            /COMPLEMENTED CONTROL WORD
00524   600516          JMP     .-6
                        /
                        .EJECT
```

```
                        /TEST 4 WRITE AND READ ROUTINE
                        /
00525   000000  BURST   0
00526   100610          JMS     SETU1           /SETUP FOR FIRST FIELD
00527   100617          JMS     CBANK           /SEE IF IT HAS PROGRAM
00530   741000          SKP                     /NO
00531   600541          JMP     DOXY            /READ XY COORDINATES
00532   777777  WONS    LAW     -1
00533   073204          DAC*    MEMADR,X        /WRITE 1'S INTO ALL FIELDS
00534   443204          ISZ     MEMADR          /ADDRESS+1
00535   443146          ISZ     CT4K            /DONE 4K WHEN SKIP
00536   600532          JMP     WONS
00537   100643          JMS     NXTBNK          /SETUP FOR NEXT FIELD
00540   600527          JMP     BURST+2
00541   100610  DOXY    JMS     SETU1           /SETUP FOR FIRST FIELD
00542   100617          JMS     CBANK           /SEE IF IT HAS PROGRAM
00543   741000          SKP                     /NO
00544   620525          JMP*    BURST           /EXIT
00545   203204          LAC     MEMADR          /ADDRESS 0 OF FIELD X
00546   343214          TAD     K100            /ADD Y LINE 01 TO IT
00547   043135          DAC     PATR            /SAVE
00550   203135  BRSTA   LAC     PATR
00551   043146          DAC     CT4K            /Y LINE ADDRESS
00552   776000          LAW     -2000           /-1024 DECIMAL
00553   043147          DAC     CT16
00554   173204          DZM*    MEMADR,X        /CLEAR LINE XN
00555   233204          LAC*    MEMADR,X        /READ 000000 512 TIMES TO
                                                /TRY TO SWITCH OTHER LINES
00556   443147          ISZ     CT16
00557   600555          JMP     .-2
00560   777777  BUST    LAW     -1
00561   273146          XOR*    CT4K,X          /LINE Y + X MUST = 777777
00562   741200          SNA                     /SHOULD NOT SKIP
00563   600571          JMP     CEND            /OK
00564   740001          CMA
00565   043154          DAC     BAD1            /SAVE BAD DATA
00566   200560          LAC     BUST
00567   043156          DAC     GOOD1           /SAVE GOOD DATA
00570   100673          JMS     ERROR           /PRINT INFO
00571   203214  CEND    LAC     K100
00572   343146          TAD     CT4K            /Y AXIS PLUS 1
00573   043146          DAC     CT4K
00574   503232          AND     K7700           /MASK Y ADDRESS
00575   740200          SZA                     /ALL DONE IF SKIP
00576   600560          JMP     BUST            /READ NEXT Y ON CURRENT X
00577   443204          ISZ     MEMADR          /INCREMENT X ADDRESS
00600   443135          ISZ     PATR            /INCREMENT X+Y ADDRESS
00601   203213          LAC     K77
00602   503204          AND     MEMADR
00603   740200          SZA                     /DONE 64 X LINES IF 0
00604   600550          JMP     BRSTA           /TEST NEW X WITH Y01 TO Y63,
00605   100643          JMS     NXTBNK          /SETUP FOR NEXT FIELD
00606   600542          JMP     DOXY+1
00607   620525          JMP*    BURST           /EXIT
                        .EJECT
```

```
                        /
                        /SETUP FOR FIRST 4K FIELD
                        /
00610   000000  SETU1   0
00611   203152          LAC     FIRST1          /FIRST TO TEST
00612   721000          PAX                     /ADDRESS COUNTER
00613   043143          DAC     SVADR
00614   770000          LAW     -10000
00615   043146          DAC     CT4K            /4K COUNTER
00616   620610          JMP*    SETU1           /EXIT
                        /
                        /ROUTINE TO SEE IF TESTED FIELD HAS PROGRAM
                        /
00617   000000  CBANK   0
00620   100653          JMS     WHERE           /CURRENT PROGRAM FIELD
00621   722000          PAL
00622   543143          SAD     SVADR           /NEXT TO TEST
00623   600644          JMP     NXTBNK+1        /SEE IF CURRENT IS LAST
00624   740031          CMAIIAC
00625   343143          TAD     SVADR
00626   721000          PAX
00627   730000          PLA
00630   043204          DAC     MEMADR
00631   620617          JMP*    CBANK           /EXIT
00632   440617  NOMOR   ISZ     CBANK           /RETURN +1
00633   620617          JMP*    CBANK
00634   203143          LAC     SVADR
00635   343234          TAD     K10K            /CURRENT +4K
00636   043143          DAC     SVADR           /NEW FIELD
00637   721000          PAX
00640   770000          LAW     -10000          /-4K
00641   043146          DAC     CT4K            /4K COUNTER
00642   600620          JMP     CBANK+1         /EXIT AND TEST NEW FIELD
                        /
                        /ROUTINE TO CHECK FOR LAST FIELD
                        /
00643   000000  NXTBNK  0
00644   750004          LAS                     /CHECK ACS0 FOR HALT
00645   741100          SPA
00646   100660          JMS     HALT            /GO HALT
00647   203143          LAC     SVADR
00650   543153          SAD     LAST1           /ALL DONE IF EQUAL
00651   600632          JMP     NOMOR
00652   600634          JMP     NOMOR+2
                        /
                        /ROUTINE TO DETERMINE WHERE PROGRAM IS
                        /
00653   000000  WHERE   0                       /CONTAINS EPC
00654   200653          LAC     .-1
00655   503240          AND     K70K            /CLEAR ALL BUT BITS 3,4,5
00656   243201          XOR     PINX
00657   620653          JMP*    WHERE           /EXIT
                        /
                        .EJECT
```

```
                        /HALT ROUTINE,  PRESS CONTINUE TO RESUME
                        /TESTING, OR IF ACS CHANGES, TO EXECUTE
                        /NEW PARAMETERS,
                        /
00660   000000  HALT    0
00661   740040          HLT                     /PRESS CONTINUE
00662   750004          LAS
00663   740010          RAL
00664   741100          SPA
00665   620660          JMP*    HALT
00666   740020          RAR
00667   503244          AND     K177
00670   543127          SAD     MCWA
00671   620660          JMP*    HALT            /RESUME WHERE LEFT OFF
00672   600247          JMP     STOVER          /EXECUTE NEW PARAMETERS
                        /
                        /ERROR PRINT-OUT ROUTINE.  PLACE ACS0 UP FOR
                        /HALT AFTER PRINT-OUT,  PRESS CONTINUE TO GO ON.
                        /
00673   000000  ERROR   0
00674   724000          PXA                     /SAVE BAD DATA
00675   503233          AND     K7777
00676   243143          XOR     SVADR
00677   043155          DAC     OCADR           /SAVE FAILING ADDRESS
00700   203251          LAC     ERWRD           /ERROR TABLE POINTER
00701   542700          SAD     ENERR           /LAST ADDRESS OF TABLE
00702   741000          SKP
00703   600707          JMP     .+4
00704   202677          LAC     ERTBL           /FIRST ADDRESS OF TABLE
00705   043251          DAC     ERWRD           /PUT POINTER TO TOP OF TABLE
00706   600716          JMP     SW2             /CHECK AC2 FOR BELL
00707   203155          LAC     OCADR           /FAILING ADDRESS
00710   503245          AND     K370K           /MASK 3, 4 AND 5
00711   543126          SAD     LAST            /NEW ERROR FIELD IF SKIP
00712   600716          JMP     .+4             /SAME FIELD AS LAST ERROR
00713   043126          DAC     LAST
00714   063251          DAC*    ERWRD           /STORE FIELD# IN TABLE
00715   443251          ISZ     ERWRD           /INCREMENT POINTER
                        /
00716   760000  SW2     LAW                     /PRINT INHIBIT IF = LAW
00717   543123          SAD     NOPRNT
00720   620673          JMP*    ERROR           /DON'T PRINT
00721   750004          LAS
00722   742010          RTL
00723   740100          SMA                     /BELL IF SKIP
00724   600730          JMP     SW1             /CHECK ACS 1
00725   760207          LAW     207             /ASCII BELL
00726   102032          JMS     PCHAR           /PRINT
00727   600736          JMP     SW0             /CHECK ACS 0 FOR HALT
00730   750004  SW1     LAS
00731   740010          RAL
00732   740100          SMA                     /NO SKIP = PRINT INFO
00733   600742          JMP     DOERR           /PRINT
00734   100660          JMS     HALT            /HALT
00735   600742          JMP     DOERR           /PRINT INFO
```

```
00736   750004      SW0     LAS
00737   741100              SPA                     /NO SKIP = HALT
00740   100660              JMS     HALT
00741   620673              JMP*    ERROR           /RETURN TO READ ROUTINE
                    /
                            ,EJECT
```

```
                    /SETUP TO PRINT ERROR
                    /
00742   203154      DOERR   LAC     BAD1            /BAD DATA
00743   741200              SNA
00744   600757              JMP     STER-6          /FULL WORD ERROR
00745   740001              CMA
00746   741200              SNA
00747   600757              JMP     STER-6          /FULL WORD ERROR
00750   740001              CMA
00751   503173              AND     BITSUP          /MASK SUPPRESSED BITS
00752   740200              SZA
00753   740001              CMA
00754   503173              AND     BITSUP
00755   741200              SNA                     /NEW ERROR IF SKIP
00756   620673              JMP*    ERROR           /ERROR IS SUPPRESSED
00757   202621              LAC     PHDR
00760   741200              SNA                     /PRINT HEADER IF 0
00761   102621              JMS     PHDR
00762   203141              LAC     TNUM            /ASCII TEST NUMBER
00763   102032              JMS     PCHAR           /PRINT TEST NO.
00764   777767              LAW     -11             /-9
00765   043145      STER    DAC     CT32            /USED FOR SPACING COUNT
00766   102104              JMS     SPING           /SPACE 9
00767   203155              LAC     OCADR           /OCTAL ADDRESS
00770   042075              DAC     CRLF            /SAVE TEMPORARILY
00771   102112              JMS     PROCTL          /PRINT FAILING ADDRESS
00772   777772              LAW     -6
00773   043145              DAC     CT32
00774   102104              JMS     SPING           /SPACE 7
00775   203156              LAC     GOOD1           /WHAT DATA SHOULD BE
00776   042075              DAC     CRLF
00777   102112              JMS     PROCTL          /PRINT THE GOOD
01000   777776              LAW     -2
01001   043145              DAC     CT32
01002   102104              JMS     SPING           /SPACE 5
01003   203154              LAC     BAD1            /DATA READ
01004   042075              DAC     CRLF            /SAVE
01005   102112              JMS     PROCTL          /PRINT THE BAD
01006   760264              LAW     264
01007   543141              SAD     TNUM
01010   601017              JMP     INDY+4
                    /
                            .EJECT
```

```
01011   777773          LAW     -5
01012   043145          DAC     CT32
01013   102104  INDY    JMS     SPING           /SPACE 5
01014   203131          LAC     CNTRL           /CURRENT CONTROL WORD
01015   042075          DAC     CRLF            /SAVE
01016   102112          JMS     PROCTL          /PRINT PATTERN CONTROL WORD
01017   102075          JMS     CRLF            /CR, LF
01020   443250          ISZ     MAXERR          /CHECK FOR MAX, PRINT-OUTS
01021   600736          JMP     SW0             /CHECK ACS 0
01022   777700          LAW     -100            /-64 DECIMAL
01023   043250          DAC     MAXERR
01024   760000          LAW
01025   043123          DAC     NOPRNT          /NO MORE ERROR PRINT-OUTS
                                                /UNTIL RESTART FROM 100
01026   202701          LAC     PTO
01027   043157          DAC     PRNT
01030   102037          JMS     PNXT            /PRINT-OUTS INHIBITED
01031   102075          JMS     CRLF            /CR,LF
01032   777766          LAW     -12             /-10 DECIMAL
01033   043145          DAC     CT32
01034   760212          LAW     212             /LF
01035   102032          JMS     PCHAR
01036   443145          ISZ     CT32            /10 LINE FEEDS
01037   601034          JMP     .-3
01040   600736          JMP     SW0
                /
                        .EJECT
```

```
                        /
                        /KEYBOARD INPUT ROUTINES
                        /
01041   735000  KYBRD   CLX
                        /
                        /TYPE TEST# AND WAIT FOR INPUT,  CARRIAGE
                        /RETURN ONLY MEANS USE LAST PATTERN WRITTEN
                        /
01042   202656  TSTNO   LAC     TSNX            /POINTER FOR TEST#
01043   043157          DAC     PRNT
01044   102075          JMS     CRLF            /CR, LF
01045   102037          JMS     PNXT            /PRINT TEST#
01046   101547          JMS     KEYIN           /WAIT FOR INPUT
01047   543227          SAD     K377            /START OVER IF A RUB-OUT
01050   601041          JMP     KYBRD
01051   543217          SAD     K215            /CHECK FOR C.R.
01052   101074          JMS     ADR1            /USE LAST PATTERN WRITTEN
01053   777517          LAW     -261            /-1 ASCII
01054   343145          TAD     CT32
01055   740100          SMA                     /MINUS = INPUT <1
01056   601061          JMP     .+3
01057   102612          JMS     WOTIS           /PRINT QUERY AND RESTART
01060   601042          JMP     TSTNO
01061   203145          LAC     CT32            /TEST# IN ASCII
01062   740001          CMA
01063   343205          TAD     K1
01064   343222          TAD     K264            /SUBTRACT ASCII 4
01065   740100          SMA                     /MINUS = TEST# >4
01066   601071          JMP     .+3
01067   102612          JMS     WOTIS           /PRINT QUERY AND RESTART
01070   601042          JMP     TSTNO
01071   760000  TSTN    LAW
01072   243145          XOR     CT32
01073   043141          DAC     TNUM            /NEW TEST NUMBER
                        /
                        .EJECT
```

```
                        /XCH15 - TAPE 2
                        /
                        /WAIT FOR FIRST 6 DIGIT ADDRESS TO LOOP ON
                        /
01074   202703  ADR1    LAC     ROTB            /POINTER FOR LAW-XX
01075   042702          DAC     ROTA
01076   222702          LAC*    ROTA
01077   043165          DAC     NROTA           /LEFT SHIFT COUNTER
01100   760000          LAW
01101   043163          DAC     ADRA
01102   143160          DZM     ADRCW           /SAVES PARTIAL ADDRESS
01103   102075          JMS     CRLF            /CR, LF
01104   202657          LAC     ADRX            /POINTER FOR FIRST ADR,
01105   043157          DAC     PRNT
01106   102037          JMS     PNXT            /PRINT FIRST ADR,
01107   202660          LAC     AD1R            /C(AD1R) = ADR1
01110   043162          DAC     OVER
01111   202661          LAC     DON1            /S(DON1) = DFST
01112   043161          DAC     EXIT
01113   101547  FADR    JMS     KEYIN           /WAIT FOR INPUT
01114   101556          JMS     LEGAL           /SEE IF VALID
01115   203145          LAC     CT32            /ASCII INPUT
01116   503207          AND     K7
01117   043145          DAC     CT32
01120   101624          JMS     GETAD           /SHIFT LEFT TO FROM ADDRESS
01121   601113          JMP     FADR            /GET NEXT DIGIT
01122   203160          LAC     ADRCW
01123   043163          DAC     ADRA
01124   202703  DFST    LAC     ROTB            /POINTER FOR LAW TABLE
01125   542702          SAD     ROTA            /NOT EQUAL = <6 CHARACTERS
01126   601131          JMP     .+3             /O.K,
01127   042702          DAC     ROTA
01130   601567          JMP     QUERY           /PRINT QUERY AND RESTART
01131   101376          JMS     PROG            /NOW SEE IF 1ST ADR, IS
                                                /IN SAME 4K AS PROGRAM
01132   601150          JMP     ADR2            /OK
01133   102075          JMS     CRLF            /CR, LF
01134   202667          LAC     ADR1P
01135   043157          DAC     PRNT
01136   102037          JMS     PNXT            /PRINT "FIRST"
01137   202671          LAC     OVRLP
01140   043157          DAC     PRNT
01141   102037          JMS     PNXT            /PRINT "ADR, IS WITHIN PROGRAM"
01142   760000          LAW
01143   041575          DAC     LOCAT+4
01144   101571          JMS     LOCAT           /TELL WHERE PROGRAM IS
01145   201623          LAC     GETAD-1
01146   041575          DAC     LOCAT+4
01147   601074          JMP     ADR1            /START OVER
                        /
                                .EJECT
```

```
                        /
                        /WAIT FOR LAST 6 DIGIT ADDRESS OF BLOCK
                        /
01150   760000  ADR2    LAW
01151   043164          DAC     ADRB
01152   143160          DZM     ADRCW
01153   222702          LAC*    ROTA
01154   043165          DAC     NROTA           /FIRST COUNT FOR LEFT SHIFT
01155   102075          JMS     CRLF            /CR, LF
01156   202662          LAC     ADXR            /POINTER FOR LAST ADR,
01157   043157          DAC     PRNT
01160   102037          JMS     PNXT            /PRINT LAST ADR,
01161   202663          LAC     AD2R            /C(AD2R) = ADR2
01162   043162          DAC     OVER
01163   202664          LAC     DON2            /C(DON2) = DLST
01164   043161          DAC     EXIT
01165   101547  LADR    JMS     KEYIN           /WAIT FOR INPUT
01166   101556          JMS     LEGAL           /SEE IF VALID
01167   203145          LAC     CT32            /ASCII INPUT
01170   503207          AND     K7
01171   043145          DAC     CT32
01172   101624          JMS     GETAD           /SHIFT LEFT TO FORM ADDRESS
01173   601165          JMP     LADR            /GET NEXT DIGIT
01174   203160          LAC     ADRCW
01175   043164          DAC     ADRB
01176   202703  DLST    LAC     ROTB
01177   542702          SAD     ROTA            /NOT EQUAL = <6 CHARACTERS
01200   601203          JMP     .+3             /O.K.
01201   042702          DAC     ROTA
01202   601567          JMP     QUERY           /PRINT QUERY AND RESTART
01203   101376          JMS     PROG            /SEE IF LAST ADDRESS IS IN
                                                /SAME 4K AS PROGRAM
01204   601222          JMP     STLP            /O.K.
01205   102075          JMS     CRLF            /CR, LF
01206   202670          LAC     ADR2P
01207   043157          DAC     PRNT
01210   102037          JMS     PNXT            /PRINT "LAST"
01211   202671          LAC     OVRLP
01212   043157          DAC     PRNT
01213   102037          JMS     PNXT            /PRINT "ADR, IS WITHIN PROGRAM"
01214   760000          LAW
01215   041575          DAC     LOCAT+4
01216   101571          JMS     LOCAT           /TELL WHERE PROGRAM IS
01217   201623          LAC     GETAD-1
01220   041575          DAC     LOCAT+4
01221   601150          JMP     ADR2            /START OVER
                        /
                        .EJECT
```

```
                        /
                        /SETUP ADDRESSES AND PATTERNS BEFORE LOOPING,
                        /
01222   760000  STLP    LAW
01223   543163          SAD     ADRA            /A LAW = NO 1ST ADDRESS
01224   741000          SKP
01225   601234          JMP     CKLST
01226   543164          SAD     ADRB            /A LAW = NO LAST ADDRESS
01227   600216          JMP     RTN1            /RESTART PROGRAM
01230   203164          LAC     ADRB
01231   043163          DAC     ADRA            /ONLY ONE SELECTED
01232   043166          DAC     LTST            /LAST OF BLOCK
01233   601276          JMP     SIMU
                        /
01234   543164  CKLST   SAD     ADRB            /A LAW = NO LAST ADDRESS
01235   741000          SKP
01236   601243          JMP     CBOTH
01237   203163          LAC     ADRA            /ONLY 1 ADDRESS WANTED
01240   043164          DAC     ADRB
01241   043166          DAC     LTST            /LAST OF BLOCK
01242   601276          JMP     SIMU
                        /
01243   203163  CBOTH   LAC     ADRA            /FIRST ADR,
01244   503240          AND     K70K            /MASK BITS 3,4 AND 5
01245   043145          DAC     CT32            /SAVE
01246   203164          LAC     ADRB            /LAST ADR,
01247   503240          AND     K70K            /MASK 3,4 AND 5
01250   543145          SAD     CT32            /BOTH MUST = SAME 4K
01251   601254          JMP     .+3             /OK
01252   102612          JMS     WOTIS           /PRINT QUERY
01253   601074          JMP     ADR1            /START OVER
                        /
01254   203163          LAC     ADRA            /FIRST ADDRESS
01255   740001          CMA
01256   343205          TAD     K1              /2'S COMPLEMENT
01257   343164          TAD     ADRB            /SUBTRACT LAST ADDRESS
01260   740100          SMA                     /FIRST IS > LAST IF NEG,
01261   601272          JMP     SIMU-4          /LEAVE AS IS
01262   203164          LAC     ADRB
01263   042075          DAC     CRLF
01264   203163          LAC     ADRA
01265   043164          DAC     ADRB            /FIRST IS NOW LAST
01266   043166          DAC     LTST
01267   202075          LAC     CRLF
01270   043163          DAC     ADRA            /LAST IS NOW FIRST
01271   601276          JMP     .+5
01272   203164          LAC     ADRB
01273   043166          DAC     LTST
01274   203163          LAC     ADRA
01275   721000          PAX
                        /
                        .EJECT
```

```
                        /
01276   760261  SIMU    LAW     261
01277   543141          SAD     TNUM            /TEST 1 IF EQUAL
01300   601312          JMP     SIM1
01301   760262          LAW     262
01302   543141          SAD     TNUM            /TEST 2 IF EQUAL
01303   601330          JMP     SIM2
01304   760263          LAW     263
01305   543141          SAD     TNUM            /TEST 3 IF EQUAL
01306   601332          JMP     SIM3
01307   760264          LAW     264
01310   543141          SAD     TNUM            /TEST 4 IF EQUAL
01311   601355          JMP     SIM4
                        /
01312   203132  SIM1    LAC     PCWA
01313   043131          DAC     CNTRL
01314   102075          JMS     CRLF
01315   203163          LAC     ADRA
01316   503240          AND     K70K
01317   043203          DAC     WORK
01320   730000          PLA
01321   740031          CMA!IAC
01322   343203          TAD     WORK
01323   721000          PAX
01324   770000          LAW     -10000
01325   043146          DAC     CT4K
01326   100430          JMS     WRITE           /WRITE PATTERN #1
01327   601346          JMP     STSCP
                        /
01330   203133  SIM2    LAC     PCWB            /WRITE PATTERN #2
01331   601313          JMP     SIM1+1
                        /
01332   203134  SIM3    LAC     PCWC            /PATTERN #3
01333   601313          JMP     SIM1+1
                        /
01334   707702  SCP1    EEM                     /SYNC
01335   210000          LAC     X               /READ
01336   740001          CMA                     /COMPLEMENT
01337   050000          DAC     X               /WRITE
01340   203203          LAC     WORK
01341   543166          SAD     LTST            /CHECK FOR END OF BLOCK
01342   601346          JMP     .+4
01343   737001          AXR+1                   /ADDRESS+1
01344   443203          ISZ     WORK
01345   601334          JMP     SCP1
01346   203163  STSCP   LAC     ADRA            /STARTING ADDRESS
01347   043203          DAC     WORK
01350   730000          PLA
01351   740031          CMA!IAC
01352   343203          TAD     WORK
01353   721000          PAX
01354   601334          JMP     SCP1
                        /
                        .EJECT
```

```
                        /
                        /ROUTINE TO SIMULATE TEST 4, 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.
                        /
01355   102075  SIM4    JMS     CRLF
01356   203163          LAC     ADRA
01357   503240          AND     K70K            /MASK FIELD NUMBER
01360   043203          DAC     WORK
01361   730000          PLA
01362   740031          CMA!IAC
01363   343203          TAD     WORK
01364   721000          PAX
01365   770000          LAW     -10000
01366   043146          DAC     CT4K            /4K COUNTER
01367   777777          LAW     -1
01370   050000          DAC     X               /WRITE 1'S
01371   443146          ISZ     CT4K            /DONE 4K WHEN SKIP
01372   601367          JMP     .-3
01373   163163          DZM*    ADRA            /CLEAR X+Y LINE
01374   223163          LAC*    ADRA            /HANG HERE AND READ
01375   601374          JMP     .-1
                        /
                        /CHECK WANTED ADDRESS AND PROGRAM AREA
                        /
01376   000000  PROG    0
01377   100653          JMS     WHERE
01400   042075          DAC     CRLF            /SAVE
01401   760000          LAW
01402   543160          SAD     ADRCW           /NONE IF = LAW
01403   621376          JMP*    PROG
01404   203160          LAC     ADRCW
01405   503240          AND     K70K
01406   542075          SAD     CRLF            /C(CRLF) = CURRENT 4K BANK
01407   741000          SKP                     /EQUAL
01410   621376          JMP*    PROG            /EXIT
01411   441376          ISZ     PROG            /RETURN+1
01412   621376          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,
                        /
01413   000000  SUPBIT  0
01414   143172          DZM     SCW             /SUPPRESSION CONTROL WORD
01415   202665          LAC     SUPX            /POINTER FOR SUPPRESSS
01416   043157          DAC     PRNT
01417   202666          LAC     SUPXA           /C(SUPXA) = SUPBIT+1
01420   043162          DAC     OVER
01421   102075          JMS     CRLF            /CR, LF
01422   102037          JMS     PNXT            /PRINT "SUPPRESS"
                        /
01423   101547  AGAIN   JMS     KEYIN           /WAIT FOINPUT
01424   543217          SAD     K215            /CHECK FOR C.R,
01425   601472          JMP     EOT             /DONE SELECTING
01426   543227          SAD     K377            /CHECK FOR RUB-OUT
01427   601414          JMP     SUPBIT+1
01430   543220          SAD     K254            /CHECK FOR COMMA
01431   601423          JMP     AGAIN           /WAIT FOR NEXT BIT POS,
01432   101522          JMS     NUMB            /DETERMINE INPUT NUMBER
01433   601567          JMP     QUERY           /NOT VALID RESTART
                        /
01434   741200          SNA                     /CHECK FOR 0
01435   601516          JMP     ZERO            /POSITION 0
01436   043167          DAC     TTYW            /SAVE DIGIT
01437   101547          JMS     KEYIN           /WAIT FOR SECOND DIGIT
01440   543220          SAD     K254            /CHECK FOR COMMA
01441   601500          JMP     EOM             /2 DIGIT POSITION
01442   543217          SAD     K215            /CHECK FOR C.R,
01443   601504          JMP     EOTA            /DONE
01444   543227          SAD     K377            /RUB-OUT IF NO SKIP
01445   601414          JMP     SUPBIT+1                /START OVER
01446   101522          JMS     NUMB            /DETERMINE NUMBER
01447   601567          JMP     QUERY           /NOT VALID, RESTART
                        /
                        .EJECT
```

```
                        /
01450   043170          DAC     TTYX            /SAVE NUMBER
01451   203167          LAC     TTYW            /PREVIOUS DIGIT
01452   744010          RCL; RCL; RCL
01453   744010
01454   744010
01455   243170          XOR     TTYX            /COMBINE DIGITS
01456   740001          CMA                     /1' COMPLEMENT
01457   043171          DAC     TTYY            /SAVE
01460   777777          LAW     -1
01461   343171          TAD     TTYY            /SUBTRACT 1
01462   043171  ROTOR   DAC     TTYY
01463   203243          LAC     K400K           /400000
01464   744020          RCR
01465   443171          ISZ     TTYY            /SHIFT COUNT
01466   601464          JMP     .-2
01467   243172          XOR     SCW             /INSERT IN CONTROL WORD
01470   043172          DAC     SCW
01471   601423          JMP     AGAIN           /WAIT FOR NEXT BIT POSITION
                        /
01472   203172  EOT     LAC     SCW             /SELECTION COMPLETED
01473   740001          CMA
01474   043173          DAC     BITSUP
01475   143172          DZM     SCW
01476   102075          JMS     CRLF            /CR,LF
01477   621413          JMP*    SUPBIT          /EXIT
                        /
01500   203167  EOM     LAC     TTYW            /SINGLE DIGIT
01501   740001          CMA
01502   343205          TAD     K1
01503   601462          JMP     ROTOR
                        /
01504   203167  EOTA    LAC     TTYW            /INPUT DIGIT
01505   740001          CMA
01506   343205          TAD     K1              /2'S COMPLEMENT
01507   043171          DAC     TTYY
01510   203243          LAC     K400K           /400000
01511   744020          RCR
01512   443171          ISZ     TTYY            /SHIFT COUNTER
01513   601511          JMP     .-2
01514   243172          XOR     SCW
01515   601473          JMP     EOT+1           /EXIT
                        /
01516   203172  ZERO    LAC     SCW
01517   243243          XOR     K400K           /400000
01520   043172          DAC     SCW
01521   601423          JMP     AGAIN           /WAIT FOR NEXT
                        /
                        .EJECT
```

```
                        /
01522   000000  NUMB    0
01523   203145          LAC     CT32            /ASCII INPUT
01524   503226          AND     K370
01525   543221          SAD     K260
01526   741000          SKP
01527   601534          JMP     .+5             /CHECK FOR A 270 OR 271
01530   441522          ISZ     NUMB            /RETURN+1
01531   203145          LAC     CT32
01532   503207          AND     K7
01533   621522          JMP*    NUMB            /EXIT
01534   543223          SAD     K270            /= 8 OR 9 IF EQUAL
01535   741000          SKP
01536   621522          JMP*    NUMB            /INVALID
01537   203145          LAC     CT32            /ASCII INPUT
01540   503205          AND     K1              /= A 8 IF BIT 17 = 0
01541   740200          SZA
01542   601545          JMP     .+3             /A 9
01543   777770          LAW     -10             /SHIFT COUNT OF 8
01544   601462          JMP     ROTOR
01545   777767          LAW     -11             /SHIFT COUNT OF 9
01546   601462          JMP     ROTOR
                        /
                        /
                        /CHARACTER INPUT ROUTINE
                        /
01547   000000  KEYIN   0
01550   700312          KRB                     /INITIALIZE
01551   700301          KSF                     /WAIT FOR INPUT
01552   601551          JMP     .-1
01553   700312          KRB                     /READ BUFFER
01554   043145          DAC     CT32            /SAVE
01555   621547          JMP*    KEYIN
                        /
                        /
                        /CHECK VALIDITY OF INPUT CHARACTER
                        /
01556   000000  LEGAL   0
01557   203145          LAC     CT32            /ASCII INPUT
01560   543227          SAD     K377            /IS IT A RUBOUT
01561   601041          JMP     KYBRD           /START OVER
01562   543217          SAD     K215            /CHECK FOR C.R.
01563   623161          JMP*    EXIT            /LINE TERMINATED
01564   503226          AND     K370
01565   543221          SAD     K260            /SHOULD EQUAL 260
01566   621556          JMP*    LEGAL           /O.K.
01567   102612  QUERY   JMS     WOTIS           /PRINT QUESTION MARK
01570   623162          JMP*    OVER            /START LINE OVER
                        /
                        .EJECT
```

```
                    /PRINT AREA CONTAINING PROGRAM
                    /
01571   000000      LOCAT   0
01572   750004              LAS
01573   503214              AND     K100
01574   740200              SZA
01575   621571              JMP*    LOCAT
01576   102075              JMS     CRLF            /CR, LF
01577   202672              LAC     PISIN
01600   043157              DAC     PRNT
01601   102037              JMS     PNXT            /PRINT "PROGRAM IS IN FIELD"
01602   100653              JMS     WHERE           /WHERE IS IT
01603   043203              DAC     WORK
01604   744010              RCL)    RTL)    RAL
01605   742010
01606   740010
01607   503207              AND     K7
01610   343221              TAD     K260
01611   102032              JMS     PCHAR           /PRINT 1ST HALF FIELD NO.
01612   203203              LAC     WORK
01613   744010              RCL)    RTL)    RTL)    RTL
01614   742010
01615   742010
01616   742010
01617   503207              AND     K7
01620   343221              TAD     K260
01621   102032              JMS     PCHAR           /PRINT 2ND HALF FIELD NO.
01622   102075              JMS     CRLF            /CR, LF
01623   621571              JMP*    LOCAT           /EXIT
                    /
                    /
                    /GENERATE 6 DIGIT ADDRESSES FROM KEYBOARD INPUT
                    /
01624   000000      GETAD   0
01625   707704              LEM
01626   222702              LAC*    ROTA            /GET A NEG, LAW FOR COUNT
01627   043165              DAC     NROTA           /SHIFT COUNTER
01630   203145              LAC     CT32            /ASCII INPUT
01631   443165      CNROT   ISZ     NROTA
01632   601650              JMP     GOLEFT          /ROTATE 1 LEFT
01633   243160              XOR     ADRCW           /ADR, CONTROL WORD
01634   043160              DAC     ADRCW
01635   777777              LAW     -1
01636   562702              SAD*    ROTA            /REC'D 6 DIGITS IF EQUAL
01637   601642              JMP     .+3
01640   442702              ISZ     ROTA            /LAW POINTER + 1
01641   621624              JMP*    GETAD           /EXIT AND WAIT FOR NEXT
01642   202703              LAC     ROTB            /ESTORE POINTERS
01643   042702              DAC     ROTA
01644   222702              LAC*    ROTA
01645   043165              DAC     NROTA
01646   441624              ISZ     GETAD           /RETURN+1
01647   621624              JMP*    GETAD           /EXIT
01650   744010      GOLEFT  RCL
01651   601631              JMP     CNROT
```

```
                    /
01652   777760      ROTC    LAW     -20         /ROTATE 15 LEFT FOR 1ST DIGIT
01653   777763              LAW     -15         /12 LEFT FOR 2ND
01654   777766              LAW     -12         /9 LEFT FOR 3RD
01655   777771              LAW     -7          /6 LEFT FOR 4TH
01656   777774              LAW     -4          /3 LEFT FOR 5TH
01657   777777              LAW     -1          /NONE FOR 6TH
                    /
                            .EJECT
```

```
                    /
                    /ROUTINE TO ACCEPT TEST LIMITS FROM KEYBOARD INPUT
                    /
01660   000000      SLMTS   0
01661   201623              LAC     GETAD=1
01662   041575              DAC     LOCAT+4         /RESTORE JMP*
01663   102075              JMS     CRLF            /CR, LF
01664   202673              LAC     TLMX            /TEST LIMITS POINTER
01665   043157              DAC     PRNT
01666   102037              JMS     PNXT            /PRINT "TEST LIMITS"
01667   102075              JMS     CRLF            /CR,LF
01670   202674              LAC     SLMX            /C (SLMX)=SLMTS+1
01671   043162              DAC     OVER
01672   202675              LAC     DON3            /RETURN ADDRESS=CREVR
01673   043161              DAC     EXIT
01674   101547              JMS     KEYIN           /WAIT FOR INPUT OF MEM, NO,
01675   543227              SAD     K377
01676   601661              JMP     SLMTS+1
01677   101556              JMS     LEGAL           /SE IF VALID
01700   203145              LAC     CT32            /ASCII INPUT
01701   503207              AND     K7              /MASK 15,16 AND 17
01702   744020              RCR;    RTR;    RAR
01703   742020
01704   740020
01705   043152              DAC     FIRST1
01706   101547              JMS     KEYIN           /WAIT FOR INPUT OF FIELD NO,
01707   543227              SAD     K377
01710   601661              JMP     SLMTS+1
01711   101556              JMS     LEGAL           /SEE IF VALID
01712   203145              LAC     CT32            /ASCII INPUT
01713   503207              AND     K7              /MASK 15,16 AND 17
01714   744020              RCR;    RTR;    RTR;    RTR
01715   742020
01716   742020
01717   742020
01720   243152              XOR     FIRST1
01721   043152              DAC     FIRST1          /FIRST FIELD TO TEST IS STORED
01722   101547              JMS     KEYIN           /WAIT FOR COMMA
01723   543220              SAD     K254
01724   741000              SKP
01725   601567              JMP     QUERY           /PRINT QUERY,  AND RESTART
01726   101547              JMS     KEYIN           /WAIT FOR INPUT OF MEM, NO,
01727   543227              SAD     K377
01730   601661              JMP     SLMTS+1
01731   101556              JMS     LEGAL           /SEE IF VALID
01732   203145              LAC     CT32            /ASCII INPUT
01733   503207              AND     K7
01734   744020              RCR;    RTR;    RAR
01735   742020
01736   740020
01737   043153              DAC     LAST1
01740   101547              JMS     KEYIN           /WAIT FOR INPUT OF FIEL NO,
01741   543227              SAD     K377
01742   601661              JMP     SLMTS+1
01743   101556              JMS     LEGAL           /SEE IF VALID
```

```
01744   203145          LAC     CT32            /ASCII INPUT
01745   503207          AND     K7              /MASK 15,16 AND 17
01746   744020          RCR;    RTR;    RTR;    RTR
01747   742020
01750   742020
01751   742020
01752   243153          XOR     LAST1
01753   043153          DAC     LAST1           /LAST FIELD TO TEST IS STORED
01754   777777          LAW     -1
01755   043146          DAC     CT4K
01756   443146  CREVR   ISZ     CT4K            /NO 2ND DIGIT IF NO SKIP
01757   601567          JMP     QUERY           /PRINT QUERY AND RESTART
01760   203152          LAC     FIRST1          /FIRST FIELD
01761   740001          CMA
01762   343205          TAD     K1              /2'S COMPLEMENT
01763   343153          TAD     LAST1           /FIRST IS >LAST IF NEG.
01764   740100          SMA
01765   601774          JMP     OKAS            /FIRST IS LOWEST ORDER
                /
                        .EJECT
```

```
                        /
01766   203152          LAC     FIRST1
01767   043151          DAC     CT04            /SAVE
01770   203153          LAC     LAST1
01771   043152          DAC     FIRST1          /LAST IS NOW FIRST
01772   203151          LAC     CT04
01773   043153          DAC     LAST1           /FIRST IS NOW LAST
01774   203153  OKAS    LAC     LAST1
01775   543152          SAD     FIRST1          /SEE IF ONLY 1 SELECTED
01776   741000          SKP                     /YES, SEE IF IT HAS PROGRAM
01777   602007          JMP     ALOK
02000   543125          SAD     INSFLD          /REJECT IF EQUAL,
02001   741000          SKP                     /TELL WHERE IT IS
02002   602007          JMP     ALOK
02003   760000          LAW
02004   041575          DAC     LOCAT*4         /CHANGE JMP* TO NOP
02005   101571          JMS     LOCAT
02006   601661          JMP     SLMTS*1         /RESTART
02007   101547  ALOK    JMS     KEYIN           /WAIT FOR A C,R
02010   543217          SAD     K215
02011   621660          JMP*    SLMTS           /EXIT
02012   601567          JMP     QUERY           /PRINT QUERY AND RESTART
                        /
                        /SETUP ACS, PRESS CARRIAGE RETURN TO EXIT
                        /
02013   000000  SETAC   0
02014   102075          JMS     CRLF            /CR,LF
02015   202676          LAC     SETX            /POINTER
02016   043157          DAC     PRNT
02017   102037          JMS     PNXT            /PRINT "SETUP ACS"
02020   700312          KRB
02021   700301          KSF
02022   602021          JMP     .-1
02023   700312          KRB
02024   543227          SAD     K377            /CHECK FOR RO
02025   601661          JMP     SLMTS*1         /RESTART
02026   750004          LAS
02027   043127          DAC     MCWA
02030   102075          JMS     CRLF            /CR,LF
02031   622013          JMP*    SETAC           /EXIT
                        /
                        .EJECT
```

```
                    /
                    /PRINT ROUTINES FOR MESSAGES
                    /
                    /PRINT ONE CHARACTER AND EXIT
                    /
02032   000000      PCHAR   0
02033   700406              TLS
02034   700401              TSF
02035   602034              JMP     .-1
02036   622032              JMP*    PCHAR
                    /
                    /PRINT A STRING AND EXIT.
                    /
02037   000000      PNXT    0
02040   777775              LAW     -3
02041   043145              DAC     CT32            /CHARACTER COUNTER
02042   443157              ISZ     PRNT            /WORD POINTER+1
02043   223157              LAC*    PRNT
02044   741200              SNA                     /ALL DONE IF 0
02045   622037              JMP*    PNXT            /EXIT
02046   042075      MASK    DAC     CRLF            /SAVE WORD
02047   503213              AND     K77             /MASK 6 BIT CHARACTER
02050   543213              SAD     K77             /CHECK IF RUBOUT
02051   602062              JMP     CK3
02052   043142              DAC     BITCON          /SAVE CHAR
02053   777740              LAW     -40
02054   343142              TAD     BITCON
02055   740100              SMA                     /NEG, = ALPHA
02056   602072              JMP     CRLF-3          /NUMERIC
02057   242053              XOR     .-4
02060   243224              XOR     K300            /MAKE ALPHA
02061   102032              JMS     PCHAR           /PRINT ACS 10-17
02062   443145      CK3     ISZ     CT32            /CHECK FOR 3 CHARACTERS
02063   741000              SKP
02064   602040              JMP     PNXT+1          /GET NEXT 3 CHARACTERS
02065   202075              LAC     CRLF            /POSITION NEXT
02066   742020              RTR; RTR; RTR
02067   742020
02070   742020
02071   602046              JMP     MASK            /PRINT IT
02072   203142              LAC     BITCON
02073   343215              TAD     K200            /MAKE NUMERIC
02074   602061              JMP     CK3-1
                    /
                            .EJECT
```

```
                        /CARRIAGE RETURN, LINE FEED
                        /
02075   000000  CRLF    0
02076   760215          LAW     215             /ASCII CR
02077   102032          JMS     PCHAR
02100   542102          SAD     .+2
02101   622075          JMP*    CRLF            /EXIT
02102   760212          LAW     212             /LF
02103   602077          JMP     CRLF+2
                        /
                        /PRINT SPACES
                        /
02104   000000  SPING   0
02105   760240          LAW     240             /ASCII SPACE
02106   102032          JMS     PCHAR
02107   443145          ISZ     CT32            /COUNTER
02110   602105          JMP     SPING+1
02111   622104          JMP*    SPING           /EXIT
                        /
                        /PRINT SIX DIGIT OCTAL NUMBERS
                        /
02112   000000  PROCTL  0
02113   777772          LAW     -6
02114   043145          DAC     CT32            /DIGIT COUNTER
02115   202075          LAC     CRLF            /OCTAL NUMBER
02116   744010  POSITN  RCL;    RTL
02117   742010
02120   042075          DAC     CRLF
02121   740010          RAL
02122   503207          AND     K7
02123   343221          TAD     K260
02124   102032          JMS     PCHAR           /PRINT
02125   443145          ISZ     CT32
02126   602115          JMP     POSITN-1                /POSITION NEXT DIGIT
02127   622112          JMP*    PROCTL          /EXIT
                        /
                        .EJECT
```

```
                        /XCH15 - TAPE 3
                        /
                        /ROUTINE TO DETERMINE FIELD FOR RELOCATION
                        /
02130   202677  CMOVE   LAC     ERTBL
02131   043251          DAC     ERWRD
02132   203153          LAC     LAST1           /LAST TO TEST
02133   543152          SAD     FIRST1          /DON'T MOVE IF EQUAL
02134   600216          JMP     RTN1            /RETURN
02135   203124          LAC     FLAGS           /PROGRAM FLAGS
02136   741100          SPA                     /FORCED MOVE MADE IF A 1.
02137   600216          JMP     RTN1            /DON'T MOVE
02140   740020          RAR                     /LINK = BIT 17
02141   741400          SZL                     /FIRST MOVE IF SKIP
02142   602227          JMP     NXTMV           /SETUP FOR NEXT MOVE
02143   443124          ISZ     FLAGS           /SET FLAG FOR 1ST MOVE
02144   203153          LAC     LAST1
02145   043125          DAC     INSFLD
02146   770000          LAW     -10000          /-4K
02147   343125          TAD     INSFLD          /SUBTRACT 4K FROM CURRENT
02150   043174          DAC     NXLOC           /NXLOC = DEST'N FOR NEXT TIME.
02151   100653          JMS     WHERE           /WHERE ARE WE NOW
02152   543125          SAD     INSFLD          /ALREADY IN LAST 1 IF EQUAL
02153   602212          JMP     SUB1            /TRY NEXT LOWER
                        /
                        /NOW CHECK FOR ERROR RECORDED IN NEW FIELD
                        /
02154   760000  CKERR   LAW
02155   563251          SAD*    ERWRD           /NO ERRORS IF = LAW
02156   602167          JMP     STMV            /INITIALIZE MOVE
02157   223251          LAC*    ERWRD
02160   543125          SAD     INSFLD          /ERROR IN FIELD IF EQUAL
02161   602201          JMP     EQUAL
02162   443251          ISZ     ERWRD           /POINTER + 1
02163   203251          LAC     ERWRD
02164   542700          SAD     ENERR           /END OF TABLE IF EQUAL
02165   741000          SKP
02166   602157          JMP     CKERR+3
                        /
02167   202677  STMV    LAC     ERTBL
02170   043251          DAC     ERWRD           /RESTORE POINTER
02171   203125          LAC     INSFLD          /NEW FIELD
02172   043176          DAC     DESTN
02173   100653          JMS     WHERE
02174   043175          DAC     SOURCE
02175   543176          SAD     DESTN
02176   600216          JMP     RTN1            /NEW AND CURRENT ARE EQUAL
02177   203176          LAC     DESTN
02200   602346          JMP     MOVE            /MOVE PROGRAM
                        /
                                .EJECT
```

```
                    /ERROR IN NEW FIELD, TRY NEXT LOWER
                    /
02201   543152      EQUAL   SAD     FIRST1          /DON'T TRY NEXT IF EQUAL
02202   602224              JMP     DNMVE
02203   741200              SNA                     /IS IT FIELD 0
02204   602210              JMP     .+4             /YES
02205   770000              LAW     -10000          /-4K
02206   343125              TAD     INSFLD          /SUBTRACT 4K FROM NEW FIELD
02207   043174              DAC     NXLOC           /NEXT NEW FIELD
02210   202677              LAC     ERTBL
02211   043251              DAC      ERWRD          /RESTORE POINTER
                    /
02212   203174      SUB1    LAC     NXLOC           /NEXT NEW FIELD
02213   543125              SAD     INSFLD          /IS IT = CURRENT NEW FIELD
02214   602201              JMP     EQUAL           /TRY NEXT LOWER
02215   043125              DAC     INSFLD          /NEW NEW FIELD
02216   543152              SAD     FIRST1          /DOES IT = LOWEST FIELD
02217   602154              JMP     CKERR           /CHECK FOR ERROR
02220   770000              LAW     -10000
02221   343125              TAD     INSFLD          /SUBTRACT 4K
02222   043174              DAC     NXLOC           /NEW FIELD FOR NEXT PASS
02223   602154              JMP     CKERR
                    /
02224   202677      DNMVE   LAC     ERTBL
02225   043251              DAC     ERWRD           /RESTORE POINTER
02226   600216              JMP     RTN1            /START OVER
                    /
                            .EJECT
```

```
                        /
                        /ROUTINE TO DETERMINE PROGRAM DEST'N AFTER MAKING ONE MOVE
                        /
02227   100653  NXTMV   JMS     WHERE           /WHERE IS PROGRAM NOW
02230   043175          DAC     SOURCE
02231   760000  CKNXT   LAW
02232   563251          SAD*    ERWRD           /NO ERRORS IF 1ST = LAW
02233   602246          JMP     STNXT
02234   202677          LAC     ERTBL
02235   043251          DAC     ERWRD
02236   223251          LAC*    ERWRD           /GET AN ERROR ADDRESS
02237   543174          SAD     NXLOC
02240   602266          JMP     SUB2            /ERROR IN NEXT FIELD TRY NEXT
02241   443251          ISZ     ERWRD
02242   203251          LAC     ERWRD
02243   542700          SAD     ENERR
02244   741000          SKP                     /DONE TABLE AND NO ERRORS
02245   602236          JMP     CKNXT+5
                        /
02246   202677  STNXT   LAC     ERTBL
02247   043251          DAC     ERWRD           /RESTORE POINTER
02250   203174          LAC     NXLOC           /NEW FIELD
02251   543125          SAD     INSFLD          /DOES IT = CURRENT FIELD
02252   602255          JMP     .+3
02253   543152          SAD     FIRST1          /DOES IT = LOWEST FIELD
02254   602303          JMP     MVBK            /YES, CLEAR FLAGS AND MOVE
02255   543152          SAD     FIRST1          /DOES THE CU RENT ALSO=
                                                /THE LOWEST FIELD,
02256   602277          JMP     NXTHI           /YES, SETUP FOR HIGHEST FIELD
02257   043125          DAC     INSFLD          /NEW CURRENT FIELD
02260   770000          LAW     -10000          /-4K
02261   343125          TAD     INSFLD
02262   043174          DAC     NXLOC           /NEW NEXT FIELD
02263   203125          LAC     INSFLD
02264   043176          DAC     DESTN
02265   602346          JMP     MOVE            /MOVE FROM HERE TO C (DESTN)
                        .EJECT
```

```
                        /
02266   203174  SUB2    LAC     NXLOC           /IS NEXT = FIELD 00 OR 1ST TO TEST
02267   543152          SAD     FIRST1
02270   602224          JMP     DNMVE           /YES, DON'T MOVE
02271   770000          LAW     -10000          /-4K
02272   343174          TAD     NXLOC           /NEW NEXT FIELD
02273   043174          DAC     NXLOC
02274   543125          SAD     INSFLD          /DOES IT = CURRENT FIELD
02275   602267          JMP     SUB2+1          /YES
02276   602234          JMP     CKNXT+3         /SEE IF ERROR IN NEW FIELD
                        /
02277   203153  NXTH1   LAC     LAST1           /LAST TO TEST
02300   503245          AND     K370K
02301   043174          DAC     NXLOC           /LAST = NEXT FIELD
02302   602234          JMP     CKNXT+3         /CHECK FOR ERROR
                        /
02303   100653  MVBK    JMS     WHERE
02304   043175          DAC     SOURCE
02305   203174          LAC     NXLOC
02306   043125          DAC     INSFLD
02307   043176          DAC     DESTN
02310   143124          DZM     FLAGS
02311   602346          JMP     MOVE
                        /
                        .EJECT
```

```
                        /ROUTINE TO FORCE MOVE THE PROGRAM; DESTINATION
                        /FIELD# MUST BE TYPED IN BY THE OPERATOR (00-37 OCTAL).
                        /
02312   203243  FCDMV   LAC     K400K
02313   740001          CMA
02314   503124          AND     FLAGS
02315   243243          XOR     K400K           /SET BIT 0 FOR FCDMV FLAG
02316   043124          DAC     FLAGS
02317   202677          LAC     ERTBL
02320   043251          DAC     ERWRD           /RESTORE TABLE POINTER
02321   102455          JMS     GOTO            /PRINT GO TO FIELD
                        /
                        /CHECK FOR ERROR IN NEW FIELD
                        /
02322   760000  CKFCD   LAW
02323   563251          SAD*    ERWRD           /NO ERRORS IF 1ST = LAW
02324   602346          JMP     MOVE            /SEE WHERE TO GO
02325   223251          LAC*    ERWRD
02326   543176          SAD     DESTN           /DOES ERROR = NEW FIELD
02327   602335          JMP     XPRT            /YES, PRINT MESSAGE
02330   443251          ISZ     ERWRD           /POINTER+1
02331   203251          LAC     ERWRD
02332   542700          SAD     ENERR           /SEE IF END OF TABLE
02333   602336          JMP     .+3             /DONE AND NO ERRORS
02334   602325          JMP     CKFCD+3
02335   102433  XPRT    JMS     PRSEL           /PRINT ERROR IN SELECTED 4K
                        /
02336   202677          LAC     ERTBL
02337   043251          DAC     ERWRD
02340   203176          LAC     DESTN
02341   543175          SAD     SOURCE
02342   600216          JMP     RTN1
02343   043125          DAC     INSFLD
02344   503242          AND     K300K
02345   043201          DAC     PINX
                        /
                        .EJECT
```

```
                        /
                        /ROUTINE TO RELOCATE THE PROGRAM
                        /
02346   142603  MOVE    DZM     LOCER
02347   203176          LAC     DESTN
02350   503242          AND     K300K
02351   043201          DAC     PINX
02352   770000          LAW     -10000          /-4K
02353   043146          DAC     CT4K            /4K COUNTER
02354   203175          LAC     SOURCE          /CURRENT FIELD
02355   043177          DAC     MOVES
02356   740031          CMA!IAC
02357   343176          TAD     DESTN
02360   721000          PAX                     /NEW FIELD
02361   143204          DZM     MEMADR          /LOC ZERO START
02362   223177  MOSOM   LAC*    MOVES           /MOVE FROM CURRENT
02363   043141          DAC     TNUM            /SAVE
02364   102442          JMS     RT1YL
02365   050000          DAC     X               /PUT IN NEW FIELD
02366   210000          LAC     X               /READ BACK
02367   563177          SAD*    MOVES           /COMPARE
02370   741000          SKP                     /OK
02371   102554          JMS     MVERR           /PRINT ERROR INFO
02372   443177          ISZ     MOVES           /INCREMENT ADDRESSES
02373   737001          AXR+1
02374   443146          ISZ     CT4K
02375   741000          SKP
02376   602424          JMP     DIND
02377   203141          LAC     TNUM
02400   542655          SAD     DLMT            /DELIMITING CHARACTER
02401   741000          SKP                     /ADJUST INDIRECTS
02402   602362          JMP     MOSOM
02403   223177  AJIN    LAC*    MOVES
02404   542716          SAD     DLMTA           /DONE INDIRECTS IF EQUAL
02405   602363          JMP     MOSOM+1
02406   503233          AND     K7777           /MASK ADDRESS BITS
02407   243176          XOR     DESTN           /PUT FIELD NUMBER ON IT
02410   102442          JMS     RT1YL
02411   050000          DAC     X               /PUT IN NEW FIELD
02412   210000          LAC     X               /READ BACK
02413   503233          AND     K7777
02414   243175          XOR     SOURCE
02415   563177          SAD*    MOVES           /COMPARE
02416   741000          SKP                     /OK
02417   102554          JMS     MVERR           /PRINT ERROR INFO
02420   443177          ISZ     MOVES           /INCREMENT ADDRESSES
02421   737001          AXR+1
02422   443146          ISZ     CT4K
02423   602403          JMP     AJIN
                        /
02424   102530  DIND    JMS     ENOT            /WAS TRANSFER MADE OK
02425   203175          LAC     SOURCE
02426   740031          CMA!IAC
02427   343176          TAD     DESTN
02430   721000          PAX
```

```
02431   740000          NOP
                                        /EXIT FROM HERE TO LOC
02432   610216          JMP     RTN1,X  /RTN1 IN NEW FIELD
                        /
                        .EJECT
```

```
                        /
                        /PRINT ERROR IN SELECTED 4K
                        /
02433   000000  PRSEL   0
02434   102075          JMS     CRLF            /CR, LF
02435   202715          LAC     ERSEL           /TEXT POINTER
02436   043157          DAC     PRNT
02437   102037          JMS     PNXT            /PRINT
02440   102075          JMS     CRLF
02441   602312          JMP     FCDMV           /WAIT FOR ANOTHER CHOICE
                        /
                        /ROTATE INSTRUCTION 19 LEFT BEFORE MOVING
                        /
02442   000000  RT19L   0
02443   744000          CLL                     /LINK = 0
02444   043141          DAC     TNUM            /SAVE
02445   777767          LAW     -11             /-9 DECIMAL
02446   043151          DAC     CT04            /SHIFT COUNT
02447   203141          LAC     TNUM            /INSTRUCTION
02450   740010          RAL
02451   742010          RTL
02452   443151          ISZ     CT04
02453   602451          JMP     .-2
02454   622442          JMP*    RT19L
                        /
                        /KEYBOARD ROUTINE FOR FORCED RELOCATION
                        /
02455   000000  GOTO    0
02456   750004          LAS                     /READ ACS
02457   503212          AND     K40
02460   741200          SNA                     /CHECK BIT 12
02461   602473          JMP     NOSW            /EQUALS 0
02462   102075          JMS     CRLF            /CR,LF
02463   202704          LAC     PTWLV           /TEXT POINTER
02464   043157          DAC     PRNT
02465   102037          JMS     PNXT            /PRINT PUT ACS 12 ON A 0
02466   750004          LAS
02467   503212          AND     K40
02470   740200          SZA                     /WAIT FOR THE 0
02471   602466          JMP     .-3
02472   102075          JMS     CRLF            /CR,LF X 2
02473   102075  NOSW    JMS     CRLF
02474   202705          LAC     GOFL            /TEXT POINTER
02475   043157          DAC     PRNT
02476   102037          JMS     PNXT            /PRINT GO TO FIELD =
02477   101547          JMS     KEYIN           /WAIT FOR INPUT
02500   543217          SAD     K215            /A CR = NO FORCED MOVE
                                                /AND RESUME AUTO RELOCATE
02501   602547          JMP     CFLG            /CLEAR THE FORCED MOVE FLAG
                        /
                        .EJECT
```

```
02502  543227            SAD     K377            /RUBOUT, RE-PRINT; GO TO FIELD -
02503  602473            JMP     NOSW
02504  742020            RTR;    RTR
02505  742020
02506  503242            AND     K300K           /MASK BITS 1,2,
02507  043176            DAC     DESTN           /FIRST CHAR, OF FIELD NO,
02510  101547            JMS     KEYIN           /WAIT FOR INPUT,
02511  543217            SAD     K215            /A CR = NO FORCED MOVE
                                                  /AND RESUME AUTO RELOCATE,
02512  602547            JMP     CFLG            /CLEAR THE FORCED MOVE FLAG,
02513  543227            SAD     K377            /RUBOUT, RE-PRINT; GO TO FIELD -
02514  602473            JMP     NOSW
02515  742020            RTR;    RTR;    RTR;    RAR
02516  742020
02517  742020
02520  740020
02521  503240            AND     K70K            /MASK BITS 3,4 & 5,
02522  243176            XOR     DESTN           /OR FIRST & SECOND CHARS,
02523  043176            DAC     DESTN           /NEW FIELD,
02524  100653            JMS     WHERE           /WHERE ARE WE NOW
02525  043175            DAC     SOURCE          /CURRENT FIELD
02526  102075            JMS     CRLF            /CR, LF
02527  622455            JMP*    GOTO            /CHECK FOR ERROR
02530  000000  ENOT      0
02531  202603            LAC     LOCER
02532  741200            SNA                     /NO ERRORS IF 0
02533  622530            JMP*    ENOT            /ENTER NEW FIELD
                         /
02534  707704            LEM
02535  142603            DZM     LOCER
02536  102075            JMS     CRLF            /CR,LF
02537  202707            LAC     NERN            /TEXT POINTER
02540  043157            DAC     PRNT
02541  102037            JMS     PNXT            /PRINT NO MORE ERRORS
02542  102075            JMS     CRLF            /CR,LF
02543  203124            LAC     FLAGS
02544  741100            SPA                     /ACS 0 A 1 = FORCED MOVE
02545  602312            JMP     FCDMV           /WAIT FOR ANOTHER CHOICE
02546  602246            JMP     STNXT           /TRY NEXT FIELD LOWER
                         /
02547  203243  CFLG      LAC     K400K
02550  740001            CMA
02551  503124            AND     FLAGS           /CLEAR THE FORCED MOVE FLAG
02552  043124            DAC     FLAGS
02553  600216            JMP     RTN1            /START OVER
                         .EJECT
```

```
                        /
02554   000000  MVERR   0
02555   043154          DAC     BAD1            /SAVE INCORRECT INSTRUCTION
02556   721000          PAX                     /FIELD AND ADDRESS
02557   043155          DAC     OCADR           /SAVE
02560   223177          LAC*    MOVES           /CORRECT INSTRUCTION
02561   043156          DAC     GOOD1           /SAVE
02562   202621          LAC     PHDR
02563   741200          SNA
02564   102621          JMS     PHDR
02565   202603          LAC     LOCER
02566   741200          SNA                     /DON'T PRINT IF 1
02567   102603          JMS     LOCER           /PRINT PROGRAM RELOCATION ERROR
02570   202620          LAC     JMP3            /JMP LOCER+3
02571   041013          DAC     INDY
02572   102075          JMS     CRLF
02573   777766          LAW     -12             /-10 DECIMAL
02574   600765          JMP     STER            /PRINT INFO
                        /
02575   200766          LAC     STER+1          /EQUALS JMS SPING
02576   041013          DAC     INDY
02577   750004          LAS
02600   741100          SPA
02601   100660          JMS     HALT
02602   622554          JMP*    MVERR           /EXIT
                        /
02603   000000  LOCER   0
02604   102075          JMS     CRLF            /CR,LF
02605   202706          LAC     RELOC           /TEXT POINTER
02606   043157          DAC     PRNT
02607   102037          JMS     PNXT            /PRINT PROGRAM RELOCATION ERROR
02610   102075          JMS     CRLF            /CR,LF X 2
02611   622603          JMP*    LOCER           /EXIT AND PRINT THE ERROR
                        /
02612   000000  WOTIS   0
02613   102075          JMS     CRLF            /CR,LF
02614   760277          LAW     277             /QUERY MARK
02615   102032          JMS     PCHAR           /PRINT
02616   102075          JMS     CRLF            /CR,LF
02617   622612          JMP*    WOTIS           /EXIT
                        /
02620   602575  JMP3    JMP     LOCER+6
                        .EJECT
```

```
                    /
                    /HEADER ROUTINE
                    /
02621   000000      PHDR    0
02622   102075              JMS     CRLF            /CR,LF
02623   202710              LAC     TSTX            /POINTER FOR "TEST"
02624   043157              DAC     PRNT
02625   102037              JMS     PNXT            /PRINT TEST
02626   102650              JMS     CLMN            /SPACE 5
02627   202711              LAC     ADRXA           /"OCTAL ADR,"
02630   043157              DAC     PRNT
02631   102037              JMS     PNXT
02632   102650              JMS     CLMN            /SPACE 5
02633   202712              LAC     GDATX           /"GOOD"
02634   043157              DAC     PRNT
02635   102037              JMS     PNXT
02636   102650              JMS     CLMN            /SPACE 5
02637   202713              LAC     BDATX           /"BAD"
02640   043157              DAC     PRNT
02641   102037              JMS     PNXT
02642   102650              JMS     CLMN            /SPACE 5
02643   202714              LAC     PCWX            /"PAT,CONTROL WORD"
02644   043157              DAC     PRNT
02645   102037              JMS     PNXT
02646   102075              JMS     CRLF            /CR,LF
02647   622621              JMP*    PHDR            /DONE
                    /
02650   000000      CLMN    0
02651   777773              LAW     -5
02652   043145              DAC     CT32
02653   102104              JMS     SPING           /SPACE
02654   622650              JMP*    CLMN
                    /
                            .EJECT
```

```
                        /
                        /RETURN ADDRESSES (INDIRECTS)
                        /
02655   752521  DLMT    752521
02656   002761  TSNX    TSTNR                   /TEST#=
02657   002765  ADRX    FADR1                   /FIRST ADR,=
02660   001074  AD1R    ADR1
02661   001124  DON1    DFST
02662   002773  ADXR    LADR1                   /LAST ADR,=
02663   001150  AD2R    ADR2
02664   001176  DON2    DLST
02665   003000  SUPX    SUPR                    /SUPPRESS=
02666   001414  SUPXA   SUPBIT+1
02667   003063  ADR1P   FRST                    /FIRST
02670   003067  ADR2P   LSTA                    /LAST
02671   003005  OVRLP   OVRLAP                  /ADR, IS WITHIN PROGRAM
02672   003037  PISIN   PROIS                   /PROGRAM IS IN FIELD
02673   003050  TLMX    TSLM                    /TEST LIMITS
02674   001661  SLMX    SLMTS+1
02675   001756  DON3    CREVR
02676   003056  SETX    STACS                   /SETUP ACS
02677   003252  ERTBL   ERWRD+1
02700   003312  ENERR   ERWRD+41
02701   003073  PTO     PTOI
02702   001652  ROTA    ROTC
02703   001652  ROTB    ROTC
02704   003112  PTWLV   PUT12
02705   003104  GOFL    GOFLD
02706   003016  RELOC   PROR
02707   003030  NERN    NOMO
02710   002717  TSTX    TST
02711   002723  ADRXA   ADR
02712   002730  GDATX   GDAT
02713   002734  BDATX   BDAT
02714   002737  PCWX    PCWR
02715   002747  ERSEL   SLTER
02716   752522  DLMTA   752522
                        .EJECT
```

```
                        /
                        /CONSTANTS FOR PRINT ROUTINE TEXTS. PACKED
                        /3 CHARACTERS PER WORD.
                        /
                        /"TEST"
02717   002717  TST     ,
02720   230524          230524; 777724; 0
02721   777724
02722   000000
                        /
02723   002723  ADR     ,
02724   240317          240317; 401401; 220401; 0
02725   401401
02726   220401
02727   000000
                        /
02730   002730  GDAT    ,
02731   171707          171707; 777704; 0
02732   777704
02733   000000
                        /
02734   002734  BDAT    ,
02735   040102          040102; 0
02736   000000
                        /
02737   002737  PCWR    ,
02740   240120          240120; 170340; 222416; 401417
02741   170340
02742   222416
02743   401417
02744   221727          221727; 777704; 0
02745   777704
02746   000000
                        /
02747   002747  SLTER   ,
02750   222205          222205; 402217; 401611; 140523
02751   402217
02752   401611
02753   140523
02754   240305          240305; 400405; 051106; 770414
02755   400405
02756   051106
02757   770414
02760   000000          0
                        /
02761   002761  TSTNR   ,
02762   230524          230524; 774024; 0
02763   774024
02764   000000
                        /
02765   002765  FADR1   ,
02766   221106          221106; 402423; 220401; 777740
02767   402423
02770   220401
02771   777740
```

```
02772   000000                  0
                        /
02773   002773          LADR1   .
02774   230114                  230114) 014024) 402204
02775   014024
02776   402204
02777   000000                  0
                        /
03000   003000          SUPR    .
03001   202523                  202523) 052220) 402323) 0
03002   052220
03003   402323
03004   000000
                                .EJECT
```

```
                        /
03005   003005  OVRLAP  .
03006   220401          220401; 231140; 112740; 111024
03007   231140
03010   112740
03011   111024
03012   204016          204016; 071722; 150122; 0
03013   071722
03014   150122
03015   000000
                        /
03016   003016  PROR    .
03017   172220          172220; 012207; 224015; 171405
03020   012207
03021   224015
03022   171405
03023   240103          240103; 161711; 220540; 221722
03024   161711
03025   220540
03026   221722
03027   000000          0
                        /
03030   003030  NOMO    .
03031   401716          401716; 221715; 054005; 172222
03032   221715
03033   054005
03034   172222
03035   772322          772322; 0
03036   000000
                        /
03037   003037  PROIS   .
03040   172220          172220; 012207; 114015; 114023
03041   012207
03042   114015
03043   114023
03044   064016          064016; 140511; 774004; 0
03045   140511
03046   774004
03047   000000
                        /
03050   003050  TSLM    .
03051   230524          230524; 144024; 111511; 772324
03052   144024
03053   111511
03054   772324
03055   000000          0
                        /
03056   003056  STACS   .
03057   240523          240523; 402025; 230301; 0
03060   402025
03061   230301
03062   000000
                        /
03063   003063  FRST    .
03064   221106          2211[redacted] 402423; 0
```

```
03065   402423
03066   000000
                        /
03067   003067  LSTA    .
03070   230114          230114; 774024; 0
03071   774024
03072   000000
                        /
03073   003073  PTOI    .
03074   112220          112220; 402416; 242517; 114023
03075   402416
03076   242517
03077   114023
03100   111016          111016; 241102; 770405; 0
03101   241102
03102   770405
03103   000000
                        /
03104   003104  GOFLD   .
03105   401707          401707; 401724; 051106; 400414
03106   401724
03107   051106
03110   400414
03111   000000          0
                        /
03112   003112  PUT12   .
03113   242520          242520; 030140; 614023; 174062
03114   030140
03115   614023
03116   174062
03117   014016          014016; 776040; 0
03120   776040
03121   000000
                        .EJECT
```

```
                    /STORAGE AND CONSTANT REGISTERS
                    /
03122   777770      SIXT4   LAW     -10             /COUNTS 64 PASSES BETWEEN
                                                    /ERROR PRINT SUPPRESSION.
03123   000000      NOPRNT  0                       /INDICATES END OF ERROR PRINT-OUTS
03124   000000      FLAGS   0                       /SAVES SUBROUTINE FLAGS
03125   000000      INSFLD  0                       /CURRENT FIELD WITH PROGRAM
03126   000000      LAST    0                       /LAST FIELD WITH DATA ERROR
03127   000000      MCWA    0                       /SAVES ACS SETTINGS
03130   000000      PCW     0                       /CURRENT PAT.CONTROL WORDS
03131   000000      CNTRL   0                       /SAME AS PCW
03132   463144      PCWA    463144                  /CONTROL WORD FOR TEST 1
03133   631460      PCWB    631460                  /FOR TEST 2
03134   525250      PCWC    525250                  /FOR TEST 3
03135   000000      PATR    0                       /ROTATES CONTROL WORD
03136   000000      PATG    0                       /SAVES GOOD DATA DURING READ
03137   000000      PATWD   0                       /SAME AS PATG BUT HAS SUPPRESSES BITS
03140   000000      PATN    0                       /HAS CONTROL WORD TO ROTATE
03141   000004      TNUM    4                       /ASCII TEST NUMBER
03142   000000      BITCON  0
03143   000000      SVADR   0                       /FIELD COUNTER
03144   000000      CT02    0
03145   000000      CT32    0
03146   000000      CT4K    0                       /4K COUNTER
03147   000000      CT16    0                       /COUNTS 16 ROTATES
03150   000000      CT128   0                       /COUNTS 128 LOCATIONS
03151   000000      CT04    0                       /UTILITY COUNTER
03152   000000      FIRST1  0                       /FIRST FIELD TO TEST
03153   000000      LAST1   0                       /LAST FIELD TO TEST
03154   000000      BAD1    0                       /SAVES BAD DATA
03155   000004      OCADR   4                       /SAVES FAILING OCTAL A DRESS
03156   000000      GOOD1   0                       /GOOD DATA
03157   000000      PRNT    0                       /POINTER FOR PRINT ROUTINES
03160   000000      ADRCW   0                       /PARTIAL ADDRESS WORD
03161   000004      EXIT    4                       /TO DISMISS
03162   000000      OVER    0                       /POINTER TO START OF SUBROUTINES
03163   000000      ADRA    0                       /1ST ADR. FROM KEYBOARD INPUT
03164   000000      ADRB    0                       /LAST ADR. FROM KEYBOARD INPUT
03165   000004      NROTA   4                       /ROTATE COUNTER
03166   000000      LTST    0                       /LAST ADR. FOR SCOPE LOOPS
03167   000000      TTYW    0                       /TTYW THRU YY USED FOR KEYBOARD
03170   000000      TTYX    0                       /INPUT BIT SUPPRESSION
03171   000000      TTYY    0
03172   000000      SCW     0                       /TEMP STORAGE OF SUPPRESSED BITS
03173   777777      BITSUP  LAW     -1              /EACH SUPPRESSED BIT = 0
03174   000000      NXLOC   0                       /NEXT FIELD TO MOVE INTO
03175   000000      SOURCE  0                       /FIELD TO MOVE FROM
03176   000000      DESTN   0                       /FIELD TO MOVE TO
03177   000000      MOVES   0                       /SAVE AS MOVED
03200   000216      BGNLO   RTN1                    /EXIT ADR. TO A NEW FIELD
03201   000000      PINX    0                       /STORAGE OF MEM. BLOCK THAT PROG. IS IN
03202   000000      SXR     0                       /STORAGE LOC. FOR XR.
03203   000000      WORK    0
03204   000000      MEMADR  0
                    .EJECT
```

```
                        /
03205   000001  K1      1
03206   000002  K2      2
03207   000007  K7      7
03210   000010  K10     10
03211   000020  K20     20
03212   000040  K40     40
03213   000077  K77     77
03214   000100  K100    100
03215   000200  K200    200
03216   000212  K212    212
03217   000215  K215    215
03220   000254  K254    254
03221   000260  K260    260
03222   000264  K264    264
03223   000270  K270    270
03224   000300  K300    300
03225   000331  K331    331
03226   000370  K370    370
03227   000377  K377    377
03230   000400  K400    400
03231   004000  K4K     4000
03232   007700  K7700   7700
03233   007777  K7777   7777
03234   010000  K10K    10000
03235   017777  K17S    17777
03236   020000  K20K    20000
03237   040000  K40K    40000
03240   070000  K70K    70000
03241   074000  K74K    74000
03242   300000  K300K   300000
03243   400000  K400K   400000
03244   177777  K177    177777
03245   370000  K370K   370000
03246   700000  K700K   700000
03247   770000  K770K   770000
03250   777700  MAXERR  LAW     -100
03251   003252  ERWRD   .+1
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
03311   760000              LAW
        000000              .END
03312   770000    #L
            SIZE=03313      NO ERROR LINES
```

```
ADR      02723
ADRA     03163
ADRB     03164
ADRCW    03160
ADRX     02657
ADRXA    02711
ADR1     01074
ADR1P    02667
ADR2     01150
ADR2P    02670
ADXR     02662
AD1R     02660
AD2R     02663
AGAIN    01423
AJIN     02403
ALOK     02007
BAD1     03154
BDAT     02734
BDATX    02713
BEGIN    00200
BGNLO    03200
BITCON   03142
BITSUP   03173
BRSTA    00550
BURST    00525
BUST     00560
CBANK    00617
CBOTH    01243
CEND     00571
CFLG     02547
CKAL     00463
CKERR    02154
CKFCD    02322
CKLST    01234
CKNXT    02231
CKXY     00472
CK3      02062
CLMN     02650
CLOF    700004
CLON    700044
CLSF    700001
CLX     735000
CMOVE    02130
CNROT    01631
CNTRL    03131
COMPL    00425
CREAD    00377
CREVR    01756
CRLF     02075
CT02     03144
CT04     03151
CT128    03150
CT16     03147
CT32     03145
CT4K     03146
```

```
DESTN   03176
DFST    01124
DIND    02424
DLMT    02655
DLMTA   02716
DLST    01176
DNMVE   02224
DOALL   00323
DOERR   00742
DON1    02661
DON2    02664
DON3    02675
DOXY    00541
EBA    707764
EEM    707702
ENERR   02700
ENOT    02530
EOM     01500
EOT     01472
EOTA    01504
EPA    707762
EQUAL   02201
ERROR   00673
ERSEL   02715
ERSET   00467
ERTBL   02677
ERWRD   03251
EXAM2   00275
EXAM3   00301
EXAM4   00305
EXBA   707741
EXIT    03161
EXTST   00271
FADR    01113
FADR1   02765
FODMV   02312
FIRST1  03152
FLAGS   03124
FRST    03063
GDAT    02730
GDATX   02712
GENPAT  00410
GETAD   01624
GOFL    02705
GOFLD   03104
GOLEFT  01650
GOOD1   03156
GOTO    02455
HALT    00660
INDY    01013
INSFLD  03125
JMP3    02620
KEYIN   01547
KRB    700312
KSP    700301
```

```
KYBRD   01041
K1      03205
K10     03210
K10K    03234
K100    03214
K17S    03235
K177    03244
K2      03206
K20     03211
K20K    03236
K200    03215
K212    03216
K215    03217
K254    03220
K260    03221
K264    03222
K270    03223
K300    03224
K300K   03242
K331    03225
K370    03226
K370K   03245
K377    03227
K4K     03231
K40     03212
K40K    03237
K400    03230
K400K   03243
K7      03207
K70K    03240
K700K   03246
K74K    03241
K77     03213
K770K   03247
K7700   03232
K7777   03233
LADR    01165
LADR1   02773
LAST    03126
LAST1   03153
LEGAL   01556
LEM     707704
LOCAT   01571
LOCER   02603
LSTA    03067
LTST    03166
MASK    02046
MAXERR  03250
MCWA    03127
MEMADR  03204
MOSOM   02362
MOVE    02346
MOVES   03177
MVBK    02303
MVERR   02554
```

```
NERN     02707
NETWK    00366
NOMO     03030
NOMOR    00632
NOPRNT   03123
NOSW     02473
NROTA    03165
NUMB     01522
NXLOC    03174
NXTBNK   00643
NXTHI    02277
NXTMV    02227
N64      00502
OCADR    03155
OKAS     01774
OVER     03162
OVRLAP   03005
OVRLP    02671
PATG     03136
PATN     03140
PATR     03135
PATWD    03137
PAX     721000
PCF     700202
PCHAR    02032
PCW      03130
PCWA     03132
PCWB     03133
PCWC     03134
PCWR     02737
PCWX     02714
PHDR     02621
PINX     03201
PISIN    02672
PNXT     02037
POSITN   02116
PRNT     03157
PROCTL   02112
PROG     01376
PROIS    03037
PROR     03016
PRSEL    02433
PSA     700204
PSB     700244
PSF     700201
PTO      02701
PTOI     03073
PTWLV    02704
PUT12    03112
QUERY    01567
RCF     700102
RCOM     00445
READ     00435
RELOC    02706
ROTA     02702
```

```
ROTB     02703
ROTC     01652
ROTOR    01462
RRB     700112
RSA     700104
RSB     700144
RSF     700101
RTN1     00216
RT19L    02442
SBA     707761
SCP1     01334
SCW      03172
SETAC    02013
SETU1    00610
SETX     02676
SIMU     01276
SIM1     01312
SIM2     01330
SIM3     01332
SIM4     01355
SIXT4    03122
SLMTS    01660
SLMX     02674
SLTER    02747
SOURCE   03175
SPING    02104
STACS    03056
STER     00765
STLP     01222
STMV     02167
STNXT    02246
STOVER   00247
STSCP    01346
SUB1     02212
SUB2     02266
SUPBIT   01413
SUPR     03000
SUPX     02665
SUPXA    02666
SVADR    03143
SW0      00736
SW1      00730
SW2      00716
SXR      03202
TCF     700402
TLMX     02673
TLS     700406
TNUM     03141
TSF     700401
TSLM     03050
TSNX     02656
TST      02717
TSTN     01071
TSTNO    01042
TSTNR    02761
```

```
TSTX    02710
TST1    00327
TST2    00340
TST3    00351
TST4    00362
TTYW    03167
TTYX    03170
TTYY    03171
WCNT    00417
WHERE   00653
WONS    00532
WORK    03203
WOTIS   02612
WRITE   00430
XPRT    02335
Y64     00507
ZERO    01516
.EOT    00000
```

```
,EOT    00000
BEGIN   00200
RTN1    00216
STOVER  00247
EXTST   00271
EXAM2   00275
EXAM3   00301
EXAM4   00305
DOALL   00323
TST1    00327
TST2    00340
TST3    00351
TST4    00362
NETWK   00366
CREAD   00377
GENPAT  00410
WCNT    00417
COMPL   00425
WRITE   00430
READ    00435
RCOM    00445
CKAL    00463
ERSET   00467
CKXY    00472
N64     00502
Y64     00507
BURST   00525
WONS    00532
DOXY    00541
BRSTA   00550
BUST    00560
CEND    00571
SETU1   00610
CBANK   00617
NOMOR   00632
NXTBNK  00643
WHERE   00653
HALT    00660
ERROR   00673
SW2     00716
SW1     00730
SW0     00736
DOERR   00742
STER    00765
INDY    01013
KYBRD   01041
TSTNO   01042
TSTN    01071
ADR1    01074
FADR    01113
DFST    01124
ADR2    01150
LADR    01165
DLST    01176
STLP    01222
```

```
CKLST   01234
CBOTH   01243
SIMU    01276
SIM1    01312
SIM2    01330
SIM3    01332
SCP1    01334
STSCP   01346
SIM4    01355
PROG    01376
SUPBIT  01413
AGAIN   01423
ROTOR   01462
EOT     01472
EOM     01500
EOTA    01504
ZERO    01516
NUMB    01522
KEYIN   01547
LEGAL   01556
QUERY   01567
LOCAT   01571
GETAD   01624
CNROT   01631
GOLEFT  01650
ROTC    01652
SLMTS   01660
CREVR   01756
OKAS    01774
ALOK    02007
SETAC   02013
PCHAR   02032
PNXT    02037
MASK    02046
CK3     02062
CRLF    02075
SPING   02104
PROCTL  02112
POSITN  02116
CMOVE   02130
CKERR   02154
STMV    02167
EQUAL   02201
SUB1    02212
DNMVE   02224
NXTMV   02227
CKNXT   02231
STNXT   02246
SUB2    02266
NXTHI   02277
MVBK    02303
FCDMV   02312
CKFCD   02322
XPRT    02335
M       02346
```

```
MOSOM   02362
AJIN    02403
DIND    02424
PRSEL   02433
RT19L   02442
GOTO    02455
NOSW    02473
ENOT    02530
CFLG    02547
MVERR   02554
LOCER   02603
WOTIS   02612
JMP3    02620
PHDR    02621
CLMN    02650
DLMT    02655
TSNX    02656
ADRX    02657
AD1R    02660
DON1    02661
ADXR    02662
AD2R    02663
DON2    02664
SUPX    02665
SUPXA   02666
ADR1P   02667
ADR2P   02670
OVRLP   02671
PISIN   02672
TLMX    02673
SLMX    02674
DON3    02675
SETX    02676
ERTBL   02677
ENERR   02700
PTO     02701
ROTA    02702
ROTB    02703
PTWLV   02704
GOFL    02705
RELOC   02706
NERN    02707
TSTX    02710
ADRXA   02711
GDATX   02712
BDATX   02713
PCWX    02714
ERSEL   02715
DLMTA   02716
TST     02717
ADR     02723
GDAT    02730
BDAT    02734
PCWR    02737
SLTER   02747
```

```
TSTNR   02761
FADR1   02765
LADR1   02773
SUPR    03000
OVRLAP  03005
PROR    03016
NOMO    03030
PROIS   03037
TSLM    03050
STACS   03056
FRST    03063
LSTA    03067
PTOI    03073
GOFLD   03104
PUT12   03112
SIXT4   03122
NOPRNT  03123
FLAGS   03124
INSFLD  03125
LAST    03126
MCWA    03127
PCW     03130
CNTRL   03131
PCWA    03132
PCWB    03133
PCWC    03134
PATR    03135
PATG    03136
PATWD   03137
PATN    03140
TNUM    03141
BITCON  03142
SVADR   03143
CT02    03144
CT32    03145
CT4K    03146
CT16    03147
CT128   03150
CT04    03151
FIRST1  03152
LAST1   03153
BAD1    03154
OCADR   03155
GOOD1   03156
PRNT    03157
ADRCW   03160
EXIT    03161
OVER    03162
ADRA    03163
ADRB    03164
NROTA   03165
LTST    03166
TTYW    03167
TTYX    03170
T       03171
```

```
SCW      03172
BITSUP   03173
NXLOC    03174
SOURCE   03175
DESTN    03176
MOVES    03177
BGNLO    03200
PINX     03201
SXR      03202
WORK     03203
MEMADR   03204
K1       03205
K2       03206
K7       03207
K10      03210
K20      03211
K40      03212
K77      03213
K100     03214
K200     03215
K212     03216
K215     03217
K254     03220
K260     03221
K264     03222
K270     03223
K300     03224
K331     03225
K370     03226
K377     03227
K400     03230
K4K      03231
K7700    03232
K7777    03233
K10K     03234
K17S     03235
K20K     03236
K40K     03237
K70K     03240
K74K     03241
K300K    03242
K400K    03243
K177     03244
K370K    03245
K700K    03246
K770K    03247
MAXERR   03250
ERWRD    03251
CLSF    700001
CLOF    700004
CLON    700044
RSF     700101
RCF     700102
RSA     700104
RRB     700112
```

```
RSB     700144
PSF     700201
PCF     700202
PSA     700204
PSB     700244
KSF     700301
KRB     700312
TSF     700401
TCF     700402
TLS     700406
EEM     707702
LEM     707704
EXBA    707741
SBA     707761
EPA     707762
EBA     707764
PAX     721000
CLX     735000
```