MK11,M211,MKA11

11/70 MAIN MEMORY
CEMKAA0

AH-C645A-MC
COPYRIGHT © 1978
FICHE 1 OF 3

APR 1978
digital
MADE IN USA

MK11,M211,MKA11

11/70 MAIN MEMORY
CEMKAA0

AH-C645A-MC
COPYRIGHT © 1978
FICHE 2 OF 3

APR 1978
digital
MADE IN USA

MK11,M211,MKA11

11/70 MAIN MEMORY
CEMKAA0

AH-C645A-MC
COPYRIGHT © 1978
FICHE 3 OF 3

APR 1978
digital
MADE IN USA

PRODUCT CODE: AC-C644A-MC

PRODUCT NAME: CEMKAA0 1170 MAIN MEMORY DIAG

DATE CREATED: 24-Feb-78

MAINTAINER: DIAGNOSTIC ENGINEERING (Memory Test Systems-CC34T)

AUTHOR: KAY R. FISHER

THE INFORMATION IN THIS DOCUMENT IS SUBJECT TO CHANGE WITHOUT NOTICE AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT CORPORATION. DIGITAL EQUIPMENT CORPORATION ASSUMES NO RESPONSIBILITY FOR ANY ERRORS THAT MAY APPEAR IN THIS DOCUMENT.

THE SOFTWARE DESCRIBED IN THIS DOCUMENT IS FURNISHED UNDER A LICENSE AND MAY ONLY BE USED OR COPIED IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE.

DIGITAL EQUIPMENT CORPORATION ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS SOFTWARE ON EQUIPMENT THAT IS NOT SUPPLIED BY DIGITAL.

COPYRIGHT (C) 1978 BY DIGITAL EQUIPMENT CORPORATION

TABLE OF CONTENTS

1.0 GENERAL PROGRAM INFORMATION

1.1 Program Purpose (Abstract)
1.2 System Requirements
1.3 Related Documents And Standards
1.4 Diagnostic Hierarchy Prerequisites
1.5 Assumptions

2.0 OPERATING INSTRUCTIONS

2.1 Loading & Starting Procedures
2.2 Special Environments
2.3 Program Options
2.4 Execution Times

3.0 ERROR INFORMATION

3.1 Error Reporting
3.2 Error Halts

4.0 PROGRESS REPORTS

5.0 DEVICE INFORMATION TABLES (CSR'S)

5.1 Address(s) Of The CSR
5.2 Allocation Of CSR Bits
5.3 CSR Bit Summary

6.0 SUB-TEST SUMMARIES

6.1 Tests
6.2 Patterns

7.0 PROGRAM FEATURES

7.1 Fast Data Access Rates
7.2 Bank Zero Testing
7.3 Differences In Core Testing With 0-2 MEG (DEMJA)
7.4 Memory Configuration Map
7.5 Banks & Boards (MK11)

8.0 FLOW CHARTS

1.0 GENERAL PROGRAM INFORMATION

1.1 Program Purpose (Abstract)

This program will verify correct operation of main memory in PDP-11/70 computers and provide diagnostic error reports.

It has special a maintenance mode (Field Service Mode) to provide specific functional capabilities.

1.2 System Requirements

1. Hardware Requirements

PDP-11/70 CPU with a minimum of 32K of MJ11 Memory or (64K of MK11 memory in a single box).

There must be no UNIBUS response to UNIBUS Address 752100 thru 752136. If a special device or memory (other than MK11) resides there it must be removed in order to run this diagnostic. The lower 16K of physical memory must have no uncorrectable errors (DBE's).

2. Software Requirements

This program is designed to run stand alone or under any of the following monitors:

XXDP
ACT
APT
RT-11

1.3 Related Documents And Standards

1. PDP-11/70 Processor Handbook
2. MJ11 Maintenance Manual
3. MK11 Maintenance Manual
4. Programming Practices (175-003-009-01)
5. System Macro Manual (MAINDEC-11-DXQAC-C-D)
6. SUPER-MAC Reference Guide (130-380-007-00)
7. APT11.MAN (APT/11-317-07-09)
8. ACT11.MAN (AUTOCAT-11-QZAUB-B-D)

1.4 Diagnostic Hierarchy Prerequisites

If the program hangs, dies, goes out to lunch, heads west, hiccups, vomits, eats itself, stutters, bombs, or in any way misbehaves, then:

1. Try it again with Cache off (reference Section 2.3.3.1)
2. Inhibit relocation (reference section 2.3.1)
3. Try CPU diagnostics
4. Try Memory Management Diagnostics
5. Try Cache Diagnostics
6. Try UNIBUS MAP Diagnostics

1.5 Assumptions

This program assumes the correct operation of the CPU, Memory Management, Cache, and the UNIBUS Map. This program occupies (initially) Bank 0 (0-16K). The XXDP loaders are in bank 1.

2.0 OPERATING INSTRUCTIONS

2.1 Loading & Starting Procedures

2.1.1 Quick Starting -

1. Load address 200
2. Set switch register for options (normally 0)
3. Start

2.1.2 Stoping -

1. Set SW8.
2. Type control "C" (Reference section 2.3.4.1).

2.1.3 Restarting (Reserve Configuration Table) -

1. Load address 202
2. Set switch register for options (Normally 0)
3. Start

2.1.4 Switch Register Options -

SWITCH	USE
15	HALT ON ERROR
14	LOOP ON TEST
13	INHIBIT ERROR TYPEOUTS
12	INHIBIT RELOCATION
11	QUICK VERIFY / INHIBIT MARGIN TESTING
10	BELL ON ERROR
9	LOOP ON ERROR
8	HALT PROGRAM (UNRELOCATED & RESTORE LOADERS)
7	DETAILED ERROR REPORTS
6	PRINT CONFIGURATION MAP
5	LIMIT MAX ERRORS PER BANK
4	FAT TERMINAL (132 COLUMNS OR BETTER)
3	TEST MODE - SEE DOCUMENT
2	TEST MODE - SEE DOCUMENT
1	TEST MODE - SEE DOCUMENT
0	DETECT SINGLE BIT ERRORS

2.2 SPECIAL ENVIRONMENTS

2.2.1 XXDP -

The first pass will be a quick verify pass if and only if it is in chain mode.

2.2.2 ACT & APT Automatic Mode -

The program will not create double bit errors (DBE's) after the 1st pass.

The first pass will be a quick verify pass

2.2.3 No SBE Free Banks -

If the program cannot find any SBE (Single Bit Error) free locations (in non-protected MK11 memory) it will print out an error message and continue testing by-passing the CSR tests.

2.2.4 Mixed Core & MOS Configurations -

The program will function normally in mixed environments. The sequence of testing may seem strange due to the recursive test mode algorithm (reference sections 2.3.1.1, 2.3.1.2, & 2.3.1.3).

2.3 Program Options

2.3.1 Switch Register Details -

If a hardware switch register is not available then the software switch register is in location 176. IF under APT if BIT7 is set in the E-TABLE symbolic location "$ENVM" the APT software switch register will be used (location $SWREG).

To change the software switch register contents: Type "control G". This will cause display the current value of the SWR and prompt for the octal input of the new SWR value from the terminal. This routine will ignore you (not respond to control "G") if you have a hardware switch register.

SW15 = HALT ON ERROR
(100000)

Continuing from this halt will first check for a change in the software switch register ("Control G" in the TTY input buffer) then it will continue testing.

SW14 = LOOP ON TEST
(40000)

This will cause looping on the present test or patern (back to last scope trap). If in a pattern then the looping will be for an entire bank of 16K addresses.

SW13 = INHIBIT ERROR TYPEOUTS
(20000)

This will cause returns from the error routine without the typed messages. Other on error functions are not affected.

SW12 = INHIBIT RELOCATION
(10000)

This prevents the program from moving and consequently prevents the program from testing at least 16K of memory - in the case of MK11 memory up to 128K of memory may be bypassed because of PROTECTION POINTER coverage (reference section 5.2.4).

SW11 = QUICK VERIFY/INHIBIT MARGIN TESTING
(4000)

Each pass of the diagnostic normally runs all margins and checks all combinations of SBE's & DBE's. If this switch is selected no margins are run (shortening pattern tests by a factor of 5) and approximately one 64th of the possible combinations of SBE's & DBE's are tested.

Each pass complete typeout will indicate this mode by preceding the pass number with "QV".

SW10 = BELL ON ERROR
(2000)

This causes a bell (or beep or click) on each error trap

SW9 = LOOP ON ERROR
(1000)

This will cause looping from failure point back to the last correctly initialized area of the current test. This loop will frequently be too large for scoping

purposes. Tighter loops are provided for in Field Service Mode reference section 2.3.4.5.10.

SW8 = HALT PROGRAM
(400)

This initiates the following sequence:

1. clears margins in "MAINT" REGISTER
2. if program is relocated it moves back to bank zero
3. flush out all possible DBE's.
4. turns off memory management
5. restore loaders
6. unmap the Unibus Map
7. Halt if under APT or ACT branch sel.

SW7 = DETAILED ERROR REPORTS
(200)

After any normal error report is typed this option causes the contents of the following registers to be typed:
R0, R1, R2, R3, R4, R5, SP, "MEMERR", "MAINT", "CONTROL", "LOADRS", "HIADRS", "CPUERR"

SW6 = PRINT CONFIGURATION MAP
(100)

This prints a map showing the memory configuration - reference section 7.4.

SW5 = LIMIT MAX ERRORS PER BANK
(40)

This will limit the number of error typeouts per bank. The default is 10. DECIMAL, however this can be changed by changing location "ERRMAX" manually or with OTT (T-reference Section 2.3.4.2).

SW4 = FAT TERMINAL
(20)

This informs the program that the console terminal has a width of at least 132 columns (LA36 with wide paper).

SW3-1 = TEST MODE

Test modes determine the recursion algorithm to be used during pattern tests.

	MODE NAME	DESCRIPTION
(0)	0	BAFPAF Banks forward, patterns forward
(2)	1	BAFPAR Banks forward, patterns reverse
(4)	2	BAWPAF Banks worst first, patterns forward.
(6)	3	BAWPAR Banks worst first, patterns reverse.
(10)	4	PAFBAF Patterns forward, banks forward
(12)	5	PAFBAW Patterns forward, banks worst first
(14)	6	PARBAF Patterns reverse, banks forward
(16)	7	PARBAW Patterns reverse, banks worst first.

NOTE: Margins follow patterns
Forward is (0,2,3,4,5)
Reverse is (5,4,3,2,0)

For more details reference section 2.3.1.1 and 2.3.1.2.

SW0 = DETECT SINGLE BIT ERRORS (SBI's)
(1)

For manufacturing purposes this switch should always be on. For field service purposes this switch should always be off.

This switch will allow all MK11 Single Bit errors to be reported by disabling error correction.

Error printouts of SBE's are not distinguishable from DBE's.

2.3.1.1 Test Mode Example -

Example analysis of mode 5 "PAFBAW". Assume Banks 0 & 1 are MJ11 (core) and Banks 2,3,4,& 5 are MK11 (MOS).

Assume also that Bank 3 is known bad by the program (via the sizing routine or previous runs reference section 3.1 and 6.1 test 22). The testing sequence would be as follows:

```
;TEST MK11 MEMORY TYPES FIRST
;TEST KNOWN BAD MEMORY (BANK 3)

PATTERN 17,     BANK 3,         MARGIN 0
PATTERN 17,     BANK 3,         MARGIN 2
PATTERN 17,     BANK 3,         MARGIN 3
PATTERN  7,     BANK 3,         MARGIN 0
PATTERN  7,     BANK 3,         MARGIN 2
PATTERN  7,     BANK 3,         MARGIN 3
PATTERN  1,     BANK 3,         MARGIN 0
PATTERN  1,     BANK 3,         MARGIN 2
PATTERN  1,     BANK 3,         MARGIN 3
PATTERN  2,     BANK 3,         MARGIN 0
PATTERN  2,     BANK 3,         MARGIN 2
PATTERN  2,     BANK 3,         MARGIN 3
PATTERN  4,     BANK 3,         MARGIN 0
PATTERN  4,     BANK 3,         MARGIN 2
PATTERN  4,     BANK 3,         MARGIN 3
PATTERN  5,     BANK 3,         MARGIN 0
PATTERN  5,     BANK 3,         MARGIN 2
PATTERN  5,     BANK 3,         MARGIN 3
PATTERN 21,     BANK 3,         MARGIN 0
PATTERN 21,     BANK 3,         MARGIN 2
PATTERN 21,     BANK 3,         MARGIN 3
PATTERN 20,     BANK 3,         MARGIN 0
PATTERN 20,     BANK 3,         MARGIN 2
PATTERN 20,     BANK 3,         MARGIN 3
PATTERN 22,     BANK 3,         MARGIN 0
PATTERN 22,     BANK 3,         MARGIN 2
PATTERN 22,     BANK 3,         MARGIN 3
PATTERN 26,     BANK 3,         MARGIN 0
PATTERN 26,     BANK 3,         MARGIN 2
PATTERN 26,     BANK 3,         MARGIN 3

;TEST PRESUMED GOOD MEMORY (BANKS 2,4,5)

PATTERN 17,     BANK 2,         MARGIN 0
PATTERN 17,     BANK 2,         MARGIN 2
PATTERN 17,     BANK 2,         MARGIN 3
PATTERN 17,     BANK 4,         MARGIN 0
PATTERN 17,     BANK 4,         MARGIN 2
PATTERN 17,     BANK 4,         MARGIN 3
PATTERN 17,     BANK 5,         MARGIN 0
PATTERN 17,     BANK 5,         MARGIN 2
PATTERN 17,     BANK 5,         MARGIN 3
PATTERN  7,     BANK 2,         MARGIN 0
```

SEQ 0012

M01
Page 12

PATTERN 20 BANK 2 MARGIN 2
PATTERN 20 BANK 2 MARGIN 2
PATTERN 20 BANK 2 MARGIN 2

SEQ 0013

```
PATTERN 20,     BANK 2,         MARGIN 3
PATTERN 20,     BANK 4,         MARGIN 0
PATTERN 20,     BANK 4,         MARGIN 2
PATTERN 20,     BANK 4,         MARGIN 3
PATTERN 20,     BANK 5,         MARGIN 0
PATTERN 20,     BANK 5,         MARGIN 2
PATTERN 20,     BANK 5,         MARGIN 3
PATTERN 22,     BANK 2,         MARGIN 0
PATTERN 22,     BANK 2,         MARGIN 2
PATTERN 22,     BANK 2,         MARGIN 3
PATTERN 22,     BANK 4,         MARGIN 0
PATTERN 22,     BANK 4,         MARGIN 2
PATTERN 22,     BANK 4,         MARGIN 3
PATTERN 22,     BANK 5,         MARGIN 0
PATTERN 22,     BANK 5,         MARGIN 2
PATTERN 22,     BANK 5,         MARGIN 3
PATTERN 26,     BANK 2,         MARGIN 0
PATTERN 26,     BANK 2,         MARGIN 2
PATTERN 26,     BANK 2,         MARGIN 3
PATTERN 26,     BANK 4,         MARGIN 0
PATTERN 26,     BANK 4,         MARGIN 2
PATTERN 26,     BANK 4,         MARGIN 3
PATTERN 26,     BANK 5,         MARGIN 0
PATTERN 26,     BANK 5,         MARGIN 2
PATTERN 26,     BANK 5,         MARGIN 3

;RELOCATE & TEST PROGRAM SPACE (BANK 0 &+ 1)

PATTERN  1,     BANK 0,         MARGIN 0
PATTERN  1,     BANK 0,         MARGIN 2
PATTERN  1,     BANK 0,         MARGIN 3
PATTERN  1,     BANK 0,         MARGIN 4
PATTERN  1,     BANK 0,         MARGIN 5
PATTERN  2,     BANK 0,         MARGIN 0
PATTERN  2,     BANK 0,         MARGIN 2
PATTERN  2,     BANK 0,         MARGIN 3
PATTERN  2,     BANK 0,         MARGIN 4
PATTERN  2,     BANK 0,         MARGIN 5
PATTERN  3,     BANK 0,         MARGIN 0
PATTERN  3,     BANK 0,         MARGIN 2
PATTERN  3,     BANK 0,         MARGIN 3
PATTERN  3,     BANK 0,         MARGIN 4
PATTERN  3,     BANK 0,         MARGIN 5
PATTERN  4,     BANK 0,         MARGIN 0
PATTERN  4,     BANK 0,         MARGIN 2
PATTERN  4,     BANK 0,         MARGIN 3
PATTERN  4,     BANK 0,         MARGIN 4
PATTERN  4,     BANK 0,         MARGIN 5
PATTERN  5,     BANK 0,         MARGIN 0
PATTERN  5,     BANK 0,         MARGIN 2
PATTERN  5,     BANK 0,         MARGIN 3
PATTERN  5,     BANK 0,         MARGIN 4
PATTERN  5,     BANK 0,         MARGIN 5
```

```
PATTERN 26,     BANK 0,         MARGIN 0
PATTERN 26,     BANK 0,         MARGIN 2
PATTERN 26,     BANK 0,         MARGIN 3
PATTERN 26,     BANK 0,         MARGIN 4
PATTERN 26,     BANK 0,         MARGIN 5
PATTERN  1,     BANK 1,         MARGIN 0
PATTERN  1,     BANK 1,         MARGIN 2
PATTERN  1,     BANK 1,         MARGIN 3
PATTERN  1,     BANK 1,         MARGIN 4
PATTERN  1,     BANK 1,         MARGIN 5
PATTERN  2,     BANK 1,         MARGIN 0
PATTERN  2,     BANK 1,         MARGIN 2
PATTERN  2,     BANK 1,         MARGIN 3
PATTERN  2,     BANK 1,         MARGIN 4
PATTERN  2,     BANK 1,         MARGIN 5
PATTERN  3,     BANK 1,         MARGIN 0
PATTERN  3,     BANK 1,         MARGIN 2
PATTERN  3,     BANK 1,         MARGIN 3
PATTERN  3,     BANK 1,         MARGIN 4
PATTERN  3,     BANK 1,         MARGIN 5
PATTERN  4,     BANK 1,         MARGIN 0
PATTERN  4,     BANK 1,         MARGIN 2
PATTERN  4,     BANK 1,         MARGIN 3
PATTERN  4,     BANK 1,         MARGIN 4
PATTERN  4,     BANK 1,         MARGIN 5
PATTERN  5,     BANK 1,         MARGIN 0
PATTERN  5,     BANK 1,         MARGIN 2
PATTERN  5,     BANK 1,         MARGIN 3
PATTERN  5,     BANK 1,         MARGIN 4
PATTERN  5,     BANK 1,         MARGIN 5
PATTERN 26,     BANK 1,         MARGIN 0
PATTERN 26,     BANK 1,         MARGIN 2
PATTERN 26,     BANK 1,         MARGIN 3
PATTERN 26,     BANK 1,         MARGIN 4
PATTERN 26,     BANK 1,         MARGIN 5
```

NOTE

This is an example & not an actual sequence.

SEQ 0015

The pattern sequence was forward (the simple patterns first, complex patterns last) sequence of patterns (MK11 = 17, 7, 1, 2, 4, 5, 21, 20, 22, 26)(MJ11 = 1, 2, 3, 4, 5, 26).

Since the patterns were forward so were the margins (0,2,3,4,5).

If the patterns were run in reverse (MK11=26,22,20,21,5,4,2,1,7,17) (MJ11=26,5,4,3,2,1,) then the margins would also have been in reverse (5,4,3,2,0)

If the bank selection is forward the banks will be tested in the following order:

1. MK11 banks that are not protected or program space (from 0 to 167).
2. MJ11 banks that are not program space (from 0 to 167).
3. The program now relocates & tests:
4. MK11 banks that were protected or program space (from 0 to 167).
5. MJ11 banks that were program space (from 0 to 167).

If bank selection is worst first the configuration table will be consulted and banks will be tested in the following order.

1. MK11 banks that are known bad and are not protected or program space (from 0 to 167).
2. MJ11 banks that are known bad and are not program space (from 0 to 167).
3. MK11 banks that are presumed good and are not protected or program space (from 0 to 167).
4. MJ11 banks that are presumed good and are not program space (from 0 to 167).
5. The program now relocates & tests:
6. MK11 banks that are known bad and were protected or program space (from 0 to 167).
7. MJ11 banks that are known bad and were program space (from 0 to 167).
8. MK11 banks that are presumed good and were protected or program space (from 0 to 167).
9. MJ11 banks that are presumed good and were program space (from 0 to 167).

2.3.1.2 Test Mode Details -

MODE 0 = "BAFPAF" banks forward, patterns forward

This is the default and simplest mode.

This mode tests each bank completely from 0 to 167 except those requiring relocation*.

While testing each bank the patterns are run with the simple ones first building to the more complex.

MODE 1 = "BAFPAR" = banks forward, patterns reverse

This mode tests each bank completely from 0 to 167 except those requiring relocation*.

While testing each bank the patterns are run with the most complex ones first, working to the simple ones.

MODE 2 = "BAWPAF" = Banks worst first, patterns forward

This mode first tests each known bad bank completely from 0 to 167 except those requiring relocation*, then presumed good banks are tested from 0 to 167 except those requiring relocation*.

While testing each bank the patterns are run with the simple ones first, building to the more complex.

MODE 3 = "BAWPAR" = Banks worst first, patterns reverse

This mode first tests each known bad bank completely from 0 to 167 except those requiring relocation*, then presumed good banks are tested from 0 to 167 except those requiring relocation*.

While testing each bank the patterns are run with the most complex ones first, working to the simple ones.

MODE 4 = "PAFBAF" = Patterns forward, banks forward

This mode tests each pattern completely with the simple ones first, building to the more complex.

While testing each pattern the banks are run from 0 to 167 except those requiring relocation*.

MODE 5 = "PAFBAW" = Patterns forward, banks worst first

This mode tests each pattern completely with the simple ones first, building to the more complex.

While testing each pattern first each known bad bank from 0 to 167 except those requiring relocation* is

run, then presumed good banks are run from 0 to 167 except those requiring relocation*.

MODE 6 = "PARBAF" = Patterns Reverse, Banks Forward

This mode tests each pattern completely with the most complex ones first, working to the simple ones.

While testing each pattern the banks are run from 0 to 167 except those requiring relocation*.

MODE 7 = "PARBAW" = Patterns Reverse, Banks Worst First

This mode tests each pattern completely with the most complex ones first, working to the simple ones.

While testing each pattern first each known bad bank from 0 to 167 except those that require relocation* is run, then presumed good banks are run from 0 to 167 except those requiring relocation*.

NOTE

* Relocation is required to test the bank(s) in program space and also to test any MKII banks protected by diagnostic checkmode with the inhibit mode pointer off (zero)!

2.3.1.3 Test Mode Applications -

1. To verify correct operation of the memory system use Mode 0 "BAFPAF".

   Advantages: Easy to understand.

   Disadvantages: In case of a failing Bank, it may take a long time to find the failure.

2. To get detailed error information on known bad Banks (found by sizing routine) use Mode 2 "BAWPAF".

   Advantages: Seeks Bad Banks. Easy to understand.

   Disadvantages: Failures other than zeros & ones may take a long time to find.

3. To get good error info on any memory problem fast use Mode 4 "PAFBAF".

   Advantages: Covers all banks fast. Easy to understand.

   Disadvantages: Failures from only complex patterns may take a long time to find.

4. To find any problem fast use Mode 7 "PARBAW".

   Advantages: Covers all Banks fast.

   Disadvantages: Difficult to understand failures reported are not necessarily the most basic failure modes.

2.3.2 Display Register -

A software display register exists in location 174 in addition to any hardware display existence.

Display fields are as follows:

```
       15 ! 14  13  12  11  10   9   8 !  7   6   5 !  4   3   2   1
relocated !          Bank  #           !    Margins  !     Pattern #
          !                            !             !
```

PATTERN # = The number of the pattern presently being run. All patterns are described in section 6.2. Any pattern can be found in the Diagnostic by Looking up the symbolic Tags "MTONN" and "MTPNN" - where "NN" is the Pattern number.
MTONN refers to the routine that sets up for the test Pattern whereas MTPNN is the actual pattern itself.

NOTE

The pattern # is not necessarily an indication of degree of difficulty.

MARGINS = The 3 bit value loaded into the Maintenance register (17777750) Bits (3:1). Margins are decoded as follows:

Margin	MJ11(core)	MK11(MOS)
0	Normal Operation	Normal Operation
2	Early Strobe	Early MDR Load
3	Late Strobe	Late Refresh
4	Low Current	Reserved (Normal)
5	High Current	Reserved (Normal)

Margin 1 = Forces "wrong address parity" error

BANK = The number of the Bank (16K) of memory under test (0-167). these bits directly map to physical address bits (21:15).

RELOCATED = This BIT indicates that the program is relocated and no longer in Bank 0. It will be relocated to the first known good non-protected memory bank indicated on the configuration map (reference section 7.4).

2.3.3 Special Memory Locations -

2.3.3.1 CACHE Constant -

The CACHE constant is located at symbolic location "CACHK" and is used to enable CACHE.

If you desire to run with group 0 disabled change this CACHE constant to 4 or 5, if you desire to run with group 1 disabled change the CACHE constant to 10 or 11, if you desire to disable all of CACHE change the CACHE constant to 14 or 15.

NOTE

Bit 0 in the CACHE constant has no effect since it is unconditionally set by the program whenever it tries to enable CACHE.

2.3.3.2 Configuration Table

The configuration table is located at symbolic location "CONFIG" and has the following format:

```
CONFIG:  First 16K Configuration words (3 each)
         2nd   16K Configuration words (3 each)
         .........................
         167th 16K configuration words (3 each)

Configuration Words:
                Low:    BIT 0-2    Interleave factor
                        BIT 3      External Interleave
                        BIT 4      CPU can access
                        BIT 5-7    Spare
                        BIT 8-10   Box number (0-7)
                        BIT 11     Errors present
                        BIT 12     Protected region of a MK11 box
                        BIT 13-14  Memory type:0=MJ11;1=MK11
                        BIT 15     Protected (Program space)
                Mad:    BIT 0-5    Not used
                        BIT 6-8    MK11 CSR Address
                        BIT 9-14   Not used
                        BIT 15     Loader's home bank
               High:    BIT 0      Indicates Bank is addressed by CSR 172100
                        BIT 1      Indicates Bank is addressed by CSR 172104
                        BIT 2      Indicates Bank is addressed by CSR 172110
                        BIT 3      Indicates Bank is addressed by CSR 172114
```

BIT 4	Indicates Bank is addressed by CSR 172120
BIT 5	Indicates Bank is addressed by CSR 172124
BIT 6	Indicates Bank is addressed by Csr 172130
BIT 7	Indicates Bank is addressed by CSR 172134
BIT 8-15	Not used

This table is used as the source for the configuration Map (reference. section 7.4).

2.3.4 Terminal Commands -

2.3.4.1 Control "C"

This command will:

1. Clear any margin conditions.
2. Unrelocate if program was relocated.
3. Flush out any DBE's.
4. Turn off Memory management.
5. Attempt to Boot RK05 Drive 0.
6. Failing 5, attempt to Boot RK05 Drive 1.
7. Failing 6, go to 5.

This command will only be recognized at the completion of the current test or pattern.

2.3.4.2 Control "D" (DEBUG)

This command will enter a modified version of ODT.

This is a subset of the ODT as documented in the RT11 System Reference Manual with the addition of Control "C", Control "F", & control "E" which function as outlined here.

NOTE

It is advisable not to enter ODT when the program is relocated.

2.3.4.3 Control "E" (procEEd)

This command will allow you to exit ODT. It is the reverse of the control "D" command. If you type control "E" after a Breakpoint - you're on your own!

2.3.4.4 Control "T" (Tell me what's happening)

This command will print out the information encoded in the display register. This is mainly intended for CPU's without a hardware display register.

Example:

RELOCATED BANK= 23 MAR= 3 PAT= 26

2.3.4.5 Control "F" (Field Service Mode)

This command will cause you to enter a mode Looking for sub commands.

When the program is looking for a sub command any number that is not a legal command will cause a mini help message to be typed. Therefore when in doubt type 99 (CR) and you will get help.

NOTE

Typing just carriage return is a default command 0.

2.3.4.5.1 Field Service command 0 (Exit)

This command will exit Field Services Mode and return to whatever task it was in prior to typing control "F". Note typing just carriage return is a default 0.

2.3.4.5.2 Field Service command 1 (Read CSR)

This command will typeout the contents of the MK11 CSR.

If there is more than one CSR on the CPU (or if the program has not yet determined the CSR status yet), it will Ask you "WHICH CSR(0-7)" to which you must respond with an Octal number from 0 to 7. Note typing just carriage return is a default 0.

If the CSR you select causes a trap to 4 the program will type "THIS CSR DOES NOT EXIST".

NOTE

CSR references are done in accordance with section 5.1.

2.3.4.5.3 Field Service command 2 (Load CSR)

This command will enable you to load the MK11 CSR.

If there is more than one CSR on the CPU (or if the program has not yet determined the CSR status yet) it will ask you "WHICH CSR(0-7)" to which you must respond with an Octal number from 0 to 7. Note typing just carriage return is a default 0.

If the CSR you select causes a trap to 4 the program will type "THIS CSR DOES NOT EXIST".

The CSR will be read and displayed as in command 1.

The program will then ask you for the "FIRST CSR WORD" to which you must respond with an Octal number. Note typing just carriage return is a default 0.

The program will then ask you for the "2ND CSR WORD" to which you must respond with an Octal number.

The program will then load the CSR and Read it again displaying its new contents.

2.3.4.5.4 Field Service command 3 (Examine Memory)

This command will allow you to examine any physical address and does the necessary memory management mapping for you.

The program will ask you for the "PHYSICAL ADDRESS (0-16777776)" to which you must respond with an Octal number.

If the address access causes a trap to 4 the program will type "TIMEOUT TRAP". If the address access causes a trap to 114 the program will type "PARITY ABORT".

The contents of your physical address will be typed.

2.3.4.5.5 Field Service command 4 (Modify Memory)

This command allows you to modify any physical address and does the necessary memory management mapping for you.

The program will ask you for the "PHYSICAL ADDRESS (0-16777776)" to which you must respond with an Octal number.

If the address access causes a trap to 4 the program will type "TIMEOUT TRAP". If the address access causes a trap to 114 the program will type "PARITY ABORT".

The program will type "OLD DATA WAS" and the contents of your physical address.

The program will then type "INPUT NEW DATA" to which you must respond with an Octal number. Note typing just carriage return is a default 0.

The program will attempt to write this new data into your physical address after which it will read it again and type "DATA IS NOW" and the new contents of your physical address.

NOTE

If you can't change the data, that would indicate that you have a double bit error in that double word pair.

2.3.4.5.6 Field Service command 5 (Select Bank, Margin & Pattern)

This command allows you to run any bank at any margin with any pattern forever.

The program will ask you "BANK(0-167)" to which you must respond with an Octal number. If the bank is not accessible. The program will type "BANK NOT ACCESSIBLE" and ask question over.

The program will then ask "PATTERN (0-37)" to which you must respond with an Octal number.

NOTE

Any pattern can be run including those that are not part of the APT E-TABLE defaults (reference section 6.2.1). If you select Pattern 0, the program will ask "PATTERN 0 DATA IS?" to which you must respond with an Octal number.

The program will then ask "MARGIN (0,2,3,4,5)?" to which you must respond with an Octal number of the set (0,2,3,4,5). If SW5 is set in the Switch register the program will type "MARGINS INHIBITED BY SW5" and ask you for the margin again. You must then either turn off SW5 or select a margin of 0.

If the Bank you selected requires relocation the program will type "BANK REQUIRES RELOCATION" and exit this command. Note normally this is true for Bank 0.

The program will then arm the console keyboard for interrupts and type "TO ESCAPE TYPE ANY KEY!".

The test pattern will be entered and run until a console key is depressed to escape this loop.

2.3.4.5.7 Field Service command 6 (Type Configuration Map)

This command types the configuration map.

This is useful after a long run (overnight) to see all the banks that are marked as bad. (Especially if your console is a video terminal).

For a detailed explanation of the map reference section 7.4.

2.3.4.5.8 Field Service Command 7 (Battery Backup Test)

This command will check that memory does not forget while powered down.

It will write and check the address pattern in every non-protected Memory Bank.

It will then prompt with "REMOVE SYSTEM POWER FOR 10 SECONDS MAX!".

If you change your mind and do not wish to power down the system type any character and the program will think that it has gone thru a power fail/auto restart sequence.

The program will then read check each non-protected bank for the address pattern and when done type "TEST COMPLETE" and exit this command.

2.3.4.5.9 Field Service Command 8 (SOB-A-LONG TEST)

This command allows execution of the SOB-A-LONG Test on all non-protected Banks reference Section 6.2.2.26. Operation is identical to command 5 except that no Pattern or Bank is entered and each pass causes a Bell.

2.3.4.5.10 Field Service Command 9 (Super Tight scope loop)

This command allows super tight scope loops on any physical address with any margin setting.

The program will ask for "PHYSICAL ADDRESS" to which you must respond with an octal number.

The program will then ask "MARGIN (0,2,3,4,5)?" to which you must respond with an Octal number of the set (0,2,3,4,5). If SW5 is set in the Switch register the program will type "MARGIN INHIBITED BY SW5" and ask you for the margin again. You must then either turn off SW5 or select a margin of 0.

The program will then type "SWR ---> MEM,MEM ---> DISPLAY" indicating that it will move the switch register into the physical address, move the physical address into the Display Register, and loop.

The the program arms the console keyboard for interrupts and types "TO ESCAPE TYPE ANY KEY!".

The program now enters a loop at symbolic location "CMD9LOOP" which consists of the following code:

```
CMD9LOOP:       MOV     @SWR,(R0)       ;WRITE SWITCH REGISTER INTO MEMORY
                MOV     (R0),@DISPLAY   ;READ MEMORY INTO DISPLAY REGISTER
                BR      CMD9LOOP        ;LOOP TILL KEYBOARD INTERRUPT
```

Typing any key will exit this command.

2.3.4.5.11 Field Service Command 10 (Error Summary)

This command types out the number of passes and the total number of errors. If there were any errors it will type out the Banks and the number of errors per bank up to 255 DECIMAL.

This becomes useful after long runs (all night) on systems with a video console terminal.

2.3.4.5.12 Field Service Command 11 (Refresh TEST)

This command allows execution of the Refresh Test on all non-protected Banks reference Section 6.2.2.19. Operation is identical to command 5 except that no Pattern or Bank is entered and each pass causes a Bell.

2.3.4.5.13 Field Service Command 12 (Set Fill Count)

This command allows setting of the terminal fill count (necessary for LA30's, ASR33's, and VT05's). It is normally set to zero for LA36's, VT52's, VT61's, etc.

2.4 Execution Times

2.4.1 Typical (System) -

Execution time depends on many variables however here are some measured times.

	CEMKAA(70 MEM)	DEMJA (OLD 0-2 MEG)
128K of MJ11 (core) Memory		
Normal Pass	2 Min 45 Sec	11 Min 41 Sec
Quick Verify	33 Sec	2 Min
128K of MK11 (MOS) Memory		
Normal Pass	2 Min 52 Sec	11 Min 41 Sec
Quick Verify	41 Sec	2 Min

2.4.2 Calculations (System)

Normal Pass (all margins)
- Add 14 Sec per MK11 Box (CSR)
- Add 20 Sec per BANK of MOS
- Add 21 Sec per BANK of Core

Quick Verify Pass
- Add 5 Sec per BANK of MOS
- Add 4 Sec per BANK of Core

2.4.3 Typical (Patterns)

Pattern	Time	Description
MT0000	<1 SEC	DATA PATTERN TEST
MT0001	<1 SEC	ADDRESS TEST
MT0002	<1 SEC	COMPLIMENT ADDRESS TEST
MT0003	1 SEC	3 XOR 9 WORST CASE NOISE TEST
MT0004	1 SEC	ROTATING ZEROS TEST
MT0005	1 SEC	ROTATING ONES TEST
MT0006	<1 SEC	INITIAL DATA TEST
MT0007	<1 SEC	ADDRESS BIT TEST
MT0010	<1 SEC	BYTE ADDRESSING TEST
MT0011	<2 SEC	CREATE SINGLE BIT ERROR TEST
MT0012	<1 SEC	WRITE BYTE CLEARS SBE TEST
MT0013	1 SEC	CREATE DOUBLE BIT ERROR TEST
MT0014	1 SEC	WRITE INHIBIT DURING DATIP WITH DBE
MT0015	1 SEC	WRITE INHIBIT OF BYTE WITH DBE
MT0016	<1 SEC	WRITE INHIBIT OF WORD WITH DBE
MT0017	<1 SEC	HOLDING 1'S & 0'S TEST
MT0020	<1 SEC	MARCHING 1'S & 0'S IN CHECK BITS

SEQ 0023

```
MT0021  :  1 SEC        MARCHING 0'S & 1'S TEST
MT0022  :  9 SEC        REFRESH TEST
MT0023  ;69 SEC         SHIFTING DIAGONAL TEST
MT0024  ;58 SEC         FAST GALLOPING PATTERN TEST
MT0025  :<1 SEC         INTERRUPT ENABLE TEST
MT0026  :<1 SEC         RANDOM DATA TEST
MT0027  :  1 SEC        UNIQUE BANK TEST
MT0030  :  1 SEC        FLUSH OUT DBE'S TEST
MT0031  :  3 SEC        SOB-A-LONG TEST
```

3.0 ERROR INFORMATION

3.1 Error Reporting

Most errors are reported using the EMT trap and handler provided by SYSMAC.SML. Most errors will be of the "MEMORY DATA ERROR" type which will be described here. MEMORY DATA ERRORS will also cause the bank to be marked as Bad in the configuration table.

Other errors are best explained by referencing the specific typeout and if necessary the program listing.

Example:

MEMORY DATA ERROR

PC	BANK	VADD	PADD	GOOD	BAD	XOR	MAR	BOX	MTYPE	INT	PAT	ARRAY
022132	37	060006	03700006	000000	000100	000100	0	0	MK11	0	06	15
022132	37	060006	03700006	000000	000100	000100	0	0	MK11	0	06	15
022132	37	060006	03700006	000000	000100	000100	0	0	MK11	0	06	15
022132	37	060006	03700006	000000	000100	000100	0	0	MK11	0	06	15

While testing Bank 37 at virtual address 60006 (virtual addresses are always between 60000 and 157776 for mapping purposes), physical address 3700006 (that's Bank 37 physical 6 within the Bank) with Pattern 6 (Initial Data Test), the good data expected was 0 but the data actually read (BAD) was 100, the exclusive OR at Good & Bad yields 100 which indicates only failing bit(s) (Bit 6). The margins were at 0 (no margin) and the memory resides in Box 0, it is an MK11 (MOS) Memory and its interleave factor is 0 (not interleaved). The CSR identified an error in slot A1 (reference Section 7.5).

NOTE

Subsequent errors of the same test do not type a new heading.

The Box Interleave and slot would be a "?" if the memory type was MJ11 (Core).

If an MJ11 (Core) gives Parity Aborts then the Bad Data & XOR will be "??????".

If no MK11 has no CSR with a SBE or DBE then slot will be a "?".

3.2 Error Halts

There are several Halts in the program.

All unused trap vectors contain a trap catcher (.WORD .+2,HALT).

An undefined TRAP instruction halts at symbolic location "$HALT2".

The APT down load sequence will halt at symbolic location "APTHLT".

Halt on Error option (SW15 Set) at symbolic location "$HALT".

Halt program (SW8 Set) at symbolic location "$EXHALT".

Power Fail will normally halt at the end of the shut down sequence (symbolic location "$DOWN").

Power Fail has a fatal Halt at symbolic location "$ILLUP" which can be caused by power up occurring before power down sequence completed or by power down before a power up sequence is completed.

4.0 PROGRESS REPORTS

Pass complete typeouts as follows:

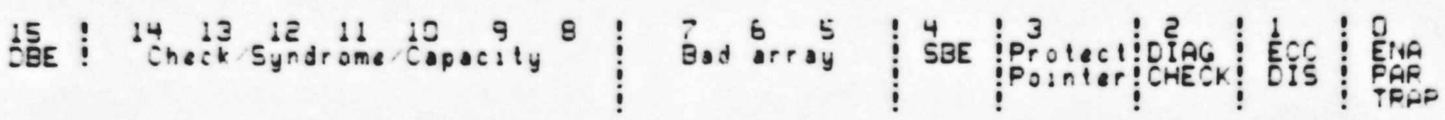
```
END PASS        #       0
END PASS        #       1
END PASS        #QV     2
```

NOTE

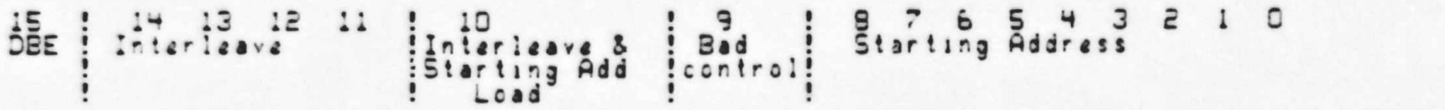
Pass 2 was flagged as a Quick Verify Pass. (Because of a change in SW5)

5.0 DEVICE INFORMATION TABLES (CSR'S)

1st Word



2nd Word



5.1 Address(s) Of The CSR

The CSR is accessed as a 32-bit word on the 11/70 main memory bus. (To the processor, the CSR looks like two consecutive 16 bit words.)

A switch on the M8158 address buffer module selects one of two options for CSR address access, either Direct or Indirect.

5.1.1 Direct Address

(For this option the 11/70 processor requires modification.) Switches on the M8158 address buffer module select one of the following eight CSR (double word) locations (octal addresses):

17777210Q
17777210Q
17777211Q
17777211Q
17777212Q
17777212Q
17777213Q
17777213Q

5.1.2 Indirect Address

(No 11/70 modifications required for this option but its use is quite complex.) Switches on the M8158 address buffer module select one of the following eight CSR (double word) locations (octal addresses):

17772100
17772104
17772110
17772114
17772120
17772124
17772130
17772134

NOTE

These addresses are in the upper 128K word section of the existing 11/70 2M word address space. Normally this is reserved address space never implemented in memory and not directly accessible by the processor.

However, these CSR locations may be accessed for diagnostic purposes through the unibus map with cache turned off. It is not recommended to use this method in an operating system environment since the unibus map is required if other NPR activity is present. One cookbook method of implementing a CSR access (within this diagnostic program) is:

1. Halt all NPR activity on the system.
2. Turn cache off.
3. Par 6 <--- 177400 (other Par's Normal)
4. Enable memory management in 22 bit mode.
5. Map 36 <--- 17760000 (other Maps = 0).
6. Turn on Unibus Map.
7. Access virtual 152100 through 152136 as CSR's.

5.1.2.1 Historical Note

This Indirect path to the CSR's was discovered in early 1977 by an expedition consisting of my three brothers and myself.

Field Service Engineer, Memory Engineer, System Engineer, and myself Software Engineer, departed the CPU on our search for the CSR's. Memory Engineer promised that CSR's would conditionally reside in the shadow of a giant memory depending on whether or not they felt switched.

We began our search for the first CSR believing that if we found one, that the rest of the tribe would surely be nearby. Our task was clear, find CSR 17772100 behind the giant memory 0 through 16777776 (2 Meg words).

Field Service Engineer suggested that since we could arrive in the east by heading west, then surely we could find 17772100 by heading for 152100. This didn't make too much sense to my other brothers and myself, however, since Field Service was willing to pay for the expedition, we all left CPU city in search of 152100.

The first village we came upon was managed by an old experienced relocator named Kay Tee. We told Mr. Tee of our expedition and our high expectations of finding 17772100 by looking for 152100. He expressed doubt that the feat could be accomplished, but he said since our three most significant bits were six that perhaps his most trusted friend Six (on of the Par brothers) could help us. So we proceeded to see Six Par.

We found Six Par at a busy highway control center directing traffic with his brothers. He greeted us warmly and assured us that we had come to the right place. Six said that his older brother Seven was not to be trusted since he preferred devices to people. Also, Six told us how his younger brothers were much too inexperienced to guide us and that they usually only do 1 to 1 relocations. So, at Six Par's highway we were given 177400 which he said to multiply by 64 and add to our 12100 (152100 less 3 MSB's). Thus, we were leaving Kay Tee's village with a total address of 17752100.

At the outskirts of the village there were two toll gates one to Cash and one to Uni. We decided to take the Cash route at System Engineer's suggestion that "Cash makes no enemies". However, the gate guard would not let us pass because he thought we were a device. We tried to explain but he insisted that unless our most significant bits were 16, we could not pass. Reluctantly, we decided to try the Uni gate. Here the guard was more negotiable and he let us pass charging only four bits. Just outside the gate we boarded a Unibus with address 752100.

The bus trip was supposed to be quite long with many stops, however, it turned out the first stop was the last stop for anybody without a device address. Here we departed the Unibus to find a long row of 36 checkout girls at special registers called Maps. Each Map register had a number 0 through 36 and a checkout girl with a matching bust

size. We all agreed to visit Map register 36. The checkout girl was very cordial and went on to say that she preferred customers whose most significant five bits were 36 also. She charged us our five most significant bits and added her favorite number 17760000 to our address of 12100 and we departed with 17772100, which Memory Engineer promised would lead to the CSR.

Next we came to the cash highway, but since the cash register was turned off, we were declared amiss and allowed to pass all the way down the highway toll free where we arrived at the Memory bus. We gave the driver our address of 17772100 and proceeded on our way. The driver warned us that no one ever goes there and he waited for us at our stop as it was not a regular part of his route.

Here waiting at the bus stop with open arms, we found the CSR and the entire tribe. We briefly exchanged data and climbed back on the Memory bus as we had to return to CPU city before the last Unibus timed out.

The journey home was without incident. It is interesting to note that once home none of the engineers including myself, could remember what the checkout girl's face looked like.

5.1.3 Indirect Manual Mode:

1. Turn off Cache 17777746 <--- 14
2. Set up MAP0 17770200 <--- 170000
   17770202 <--- 77
3. Turn on Unibus Map 17772516 <--- 40
4. Turn display switch to CONS PHY
5. Examine or deposit 17002100 through 17002136 as CSR's

5.2 Allocation Of CSR Bits

5.2.1 Enable Parity Traps (Bit0 - of 1st word)

If this bit is set, it allows bad parity to be asserted (for a READ operation) if a double error occurs or a single error occurs with ECC disabled. Section 5.2.5 summarizes the operation of this bit along with other control bits. This bit is set on power up and can be written or read by the main memory bus.

5.2.2 ECC Disable (Bit1 - 1st word)

If this bit is set, the error correction circuits are disabled and no single bit error correction will occur. However, error detection circuitry is enabled and any single or double bit error may still be detected on a READ operation and correct check bits are still written into memory on a write operation.

This is a diagnostic aid to allow writing and reading data from memory without interference from the error correction logic. This is a READ/WRITE bit cleared on power up.

5.2.3 Diagnostic Check (Bit2 - 1st word)

This mode allows a means of writing data stored in CSR Bits (14:8) into the MOS chips used for check bit storage. A READ cycle to memory will read the check bits from MOS storage into CSR bits (14:8).

This allows an integrity check on the check bit storage chips and a check on the error correction logic.

This bit is cleared on power up and can be written or read by the main memory bus.

5.2.4 Protection Pointer (Bit3 - 1st word)

This bit selects the 16K section of memory which will be protected (not be affected by ECC disable and diagnostic check). This 16K section will contain the diagnostic program. When this bit is unasserted, the first 16K associated with each control card is protected. When this bit is asserted the second 16K associated with each control card will be protected. This is a READ/WRITE bit cleared on power up.

5.2.5 Summary of Bits(3:0) of 1st word

CSR	BIT3	BIT2	BIT1	BIT0	FUNCTION
	0	0	0	0	Parity trap, disable (Normal Mode)
	0	0	0	1	Enable traps on DBE's (Initialized Value from power up)
	0	0	1	0	Disable error correction & traps
	0	0	1	1	Enable traps on SBE's
	0	1	0	0	R/W checkbits
	0	1	0	1	Not used by program
	0	1	1	0	Not used by program
	0	1	1	1	Read Box Capacity
	1	0	0	?	Read Box Capacity in Bits (14:8) of 1st word
	1	0	1	0	Disable error correction & traps (Program relocated)
	1	0	1	1	Enable traps on SBE's (Program relocated)
	1	1	0	0	R/W Checkbits (Program relocated)
	1	1	0	1	Not used by program
	1	1	1	0	Not used by program
	1	1	1	1	Not used by program

? = DON'T CARE

5.2.6 Single Bit Error Indication (Bit4 - 1st word)

On a READ cycle to memory, if a single bit error occurs, this bit is set. Simultaneously, the syndrome bits are latched into CSR Bits (14:8) and the array identification and controller selected are latched into CSR (Bits [7:5] 1st word) and Bit9 2nd word.

An examination of the CSR can thus be performed to see if a single bit error had occurred and the array that failed if such error occurred. This is a READ/WRITE bit cleared on power up.

5.2.7 Array Select (Bits (7:5) 1st word)

During a READ cycle to memory, latched address bits on the selected control board will be stored in these locations upon an error. Any DBE will rewrite these bits & SBE's from protected memory will not be latched.

5.2.8 Check Bit/Syndrome Bit Storage/Box Capacity (Bits (14:8) 1st word)

5.2.8.1 Check Bit Storage (DIAG CK = 1)

During a diagnostic check, bits are read from or written into memory as described in Section 5.2.3.

5.2.8.2 Syndrome Bit Storage

If a single bit error occurs on a read cycle, then the syndrome bits indicating what bit failed are stored into those CSR bits.

These CSR bits can be written or read by the main memory bus.
They are neither set nor reset on power up.

5.2.8.3 Hamming Code Matrix

Each check bit (C1-C32) is the result of odd parity generated on the data bits designated by X.

C1 = f(1,3,5,7,9,11,13,15,17,19,21,23,25,27,29,31)
C2 = f(2,3,6,7,10,11,14,15,18,19,22,23,26,27,30,31)
C4 = f(4-7,12-15,20-23,28-31)
C8 = f(0-15,24-31)
C16= f(0-7,16-31)
C32= f(8-31)

Check bit CT (check total) is the result of even parity generated on the data bits.

CT = f(0,3,5,6,P 11,13,14,16,19,21,22,25,26,28,31)

Syndrone bits (S1-ST) are formed by a comparison of the check bits read from memory and new check bits generated on the data read from

memory.

A single error will cause an odd number of syndrome bits to go high. If Bit 4 should fail S16, S8, & S4 would be set.

An examination of the syndrome bits can disclose the failing bit.

If there are more than one syndrome bits set and still an odd number of bits then the following simplification exists. S4-S1 indicate in octal the failing Bit (0-7). S32-S8 indicate the Bad Byte. The cleared bit in S32-S8 indicates which Byte from left to right. S32-S8 = 3 = Byte 0; 5 = Byte 1. 6 = byte 2; and all set = 7 = Byte 3. ST (Syndrome total) should be set or cleared to produce an odd number of syndrome.

If only one syndrome bit is set it directly indicates which check bit is bad.

If an even number of syndrome bits are set something is wrong with the hardware.

SEG 0041

		SYNDROME BITS						
	BIT	ST	S32	S16	S8	S4	S2	S1
BYTE 0	0	X		X	X			
	1			X	X			X
	2			X	X		X	
	3	X		X	X		X	X
	4			X	X	X		
	5	X		X	X	X		X
	6	X		X	X	X	X	
	7			X	X	X	X	X
BYTE 1	8	X	X		X			
	9		X		X			X
	10		X		X		X	
	11	X	X		X		X	X
	12		X		X	X		
	13	X	X		X	X		X
	14	X	X		X	X	X	
	15		X		X	X	X	X
BYTE 2	16	X	X	X				
	17		X	X				X
	18		X	X			X	
	19	X	X	X			X	X
	20		X	X		X		
	21	X	X	X		X		X
	22	X	X	X		X	X	
	23		X	X		X	X	X
BYTE 3	24		X	X	X			
	25	X	X	X	X			X
	26	X	X	X	X		X	
	27		X	X	X		X	X
	28	X	X	X	X	X		
	29		X	X	X	X		X
	30		X	X	X	X	X	
	31	X	X	X	X	X	X	X
CHECK BITS	C1						X	
	C2							X
	C4					X		
	C8				X			
	C16			X				
	C32		X					
	CT	X						

5.2.8.4 Box Capacity

Multiplexed with the locations for reading and/or writing check bits or reading syndrome bits is the ability to read the total number of 16K X 32 sections (double blocks) in memory. In order to read this box capacity, the special code as indicated in Section 5.2.5 (bits (3:1) = 1,0,0) will have to be loaded into the CSR 1st word prior to performing the READ cycle.

(Note that if ECC is manually disabled, writing a Zero into CSR Bit2 1st word is overridden and a One remains. In order to read the box capacity for this case, an internal flip-flop is used whose state is what was written into CSR Bit2 1st word. Hence writing Bits(3:1) = 1,0,0 1st word will still allow box capacity to be read even though CSR BIT2 1st word is still a One.)

5.2.9 Double Error Indication (Bit 15 of 1st or 2nd word)

On a READ cycle, these bits are set if a double error occurs. Simultaneously the array select bits latched on the control board selected are stored into CSR Bits (7:5) 1st word and the latched control is stored in Bit 9 2nd word. It should be noted that Bits 15 of 1st & 2nd word are separate and independent. Clearing one bit via a write to that byte will not affect the other. They are however asserted by the addressed controller simultaneously. These are READ/WRITE bits reset on power up.

5.2.10 Starting Address (A24-A16) (Bits (8:0) 2nd word)

These bits are used to load a starting box address through the CSR on a WRITE cycle or can be used to read the starting address of the memory box if the address is set by switches on the data interface card or remotely from the processor. The starting address can be reconfigured on a 32K word segment. These bits are READ/WRITE. On power up the switch settings on the data interface card are loaded into these CSR locations. In order to load the starting address Bit 10 2nd word must accompany the load.

5.2.11 Control Select (Bit 9 2nd word)

If a single or double bit error occurs during a READ cycle, then the address of the controller selected is stored in this CSR bit location.

This bit is read only and cleared on power up.

5.2.12 Starting Address Selector (Bit 10 2nd word)

This bit controls selection of the starting address & External Interleave loaded into the CSR or the starting address and external interleave dictated by the switches on the data buffer board or control panel. A Zero in this bit indicates selection of the switches, a One indicates selection of the starting address loaded into CSR bits (8:0) 2nd word and External Interleave bits (14-12) 2nd word!

(Note that a READ operation to the CSR will always give the actual box starting address in Bits (8:0) 2nd word regardless of its origin.)

This is a READ/WRITE bit cleared on power up. (However, a switch on the data buffer board can hold this bit cleared, thereby defeating any attempt at loading the starting address and External Interleave through the CSR.)

5.2.13 Internal Interleaving (Bit 11 2nd word)

This bit indicates if there is internal interleaving within a memory box. If this bit is cleared there is no internal interleaving and if it is set, there is internal interleaving.

NOTE

Only if each controller has identical memory capacity is this bit capable of being set. However, writing a Zero to this bit disables all possible internal interleaving. This is a READ/WRITE bit set on power up if each of the two control card sets has identical memory capacity.

5.2.14 External Interleaving (Between Boxes) (Bits (14:11) 2nd word)

These bits indicate how the system is configured for external interleaving as follows:

BIT14	BIT12	BIT13	NUMBER OF WAYS EXTERNALLY INTERLEAVED
0	0	0	No External Interleaving
1	0	0	Not Used
0	1	0	2 Way: Memory Responds to Even Double Words
1	1	0	: Memory Responds to Odd Double Words
0	0	1	4 Way: Memory Selected If: A02=0; A03=0
1	0	1	: Memory Selected If: A02=1; A03=0
0	1	1	: Memory Selected If: A02=0; A03=1
1	1	1	: Memory Selected If: A02=1; A03=1

The bits will be treated in a similar manner to the Start Address as described in Section 5.2; i.e., CSR Bit10 2nd word will determine the origin. Also, on power up the corresponding switch settings will be loaded into these CSR bit locations.

5.2.15 Interleave Summary

BIT14	BIT13	BIT12	BIT11	DESCRIPTION
0	0	0	0	No Interleave
0	0	0	1	Internal 2-1
0	0	1	0	External 2-1 Even Double Words
0	0	1	1	Ext 2-1 & Int Even Double Words
0	1	0	0	Ext 4-1 A3=0 A2=0
0	1	0	1	Ext 4-1 & INT A3=0 A2=0
0	1	1	0	Ext 4-1 A3=1 A2=0
0	1	1	1	Ext 4-1 & Int A3=1 A2=0
1	0	0	0	Impossible (ERROR)
1	0	0	1	Impossible (ERROR)
1	0	1	0	Ext 2-1 Odd Double Words
1	0	1	1	Ext 2-1 & Int Odd Double Words
1	1	0	0	Ext 4-1 A3=0,A2=1
1	1	0	1	Ext 4-1 & Int A3=0,A2=1
1	1	1	0	Ext 4-1 A3=1,A2=1
1	1	1	1	Ext 4-1 & Int A3=1,A2=1

5.3 CSR Bit Summary

CSR BIT ALLOCATION - AS DESIGNED			COMMENTS
1st word	Bit0	Enable Parity Traps	: Set on Power Up (Read/Write)
	Bit1	ECC Disable	: Clear on Power Up (R/W)
	Bit2	Diagnostic Check	: Clear on Power Up (R/W)
	Bit3	Protection Pointer	: Clear on Power Up (R/W)
	Bit4	Single Bit Error Indication	: Clear on Power Up (R/W)
	Bit5	Array Sel A01	: Clear on Power Up (Read Only)
	Bit6	Array Sel A02	: Clear on Power Up (Read Only)
	Bit7	Array Sel A03	: Clear on Power Up (Read Only)
	Bit8	CK Bit1/Syndrome 1/Capacity 1	: (Read/Write) Bits, Neither
	Bit9	CK Bit2/Syndrome 2/Capacity 2	: Set nor Reset on Power
	Bit10	CK Bit4/Syndrome 4/Capacity 4	: Up; Based on CSR Bit2 & bit3 Check
	Bit11	CK Bit8/Syndrome 8/Capacity 8	: Bits Alone are Read from or
	Bit12	CK Bit16/Syndrome 16/Capacity16	: Written into these locations
	Bit13	CK Bit32/Syndrome 32/Capacity32	: Otherwise Syndromes are Written on SBE's
	Bit14	CK Bit T/Syndrome T/Capacity64	
	Bit15	Double Bit Error Indication	: Clear on Power Up (R/W)
2nd word	Bit0	Start Address (A16)	: (Read/Write) Bits Capable of
	Bit1	Start Address (A17)	: Being Loaded via the Main Data
	Bit2	Start Address (A18)	: Bus or Switches on the Data
	Bit3	Start Address (A19)	: Card. On Power Up, the switch
	Bit4	Start Address (A20)	: Settings are loaded into
	Bit5	Start Address (A21)	: These Locations.
	Bit6	Start Address (A22)	
	Bit7	Start Address (A23)	
	Bit8	Start Address (A24)	
	Bit9	Controller Select	: Cleared on Power Up (Read Only)
	Bit10	Start Address Selector	: Loaded on Power Up (Read/Write)
	Bit11	Internal Interleaving	: Loaded on Power Up (Read/Write)
	Bit12	Two Way	: Loaded Same as Bits (8:0) 2nd word
	Bit13	Four Way	: Loaded Same as Bits (8:0) 2nd word
	Bit14	Ext A02	: Loaded Same as Bits (8:0) 2nd word
	Bit15	Double Bit Error Indications	: Clear on Power Up (R/W)

6.0 SUB-TEST SUMMARIES

6.1 Tests

TEST 1

ANALIZE CONFIGURATION
(CSR Access may cause wrong Type of Traps)

TEST 2

TEST BANK 0 ACCESSES
Failures are fatal.

TEST 3

TEST BANKS 1-167 (OCTAL) FOR ZEROS & ONES
Errors are not typed here - only logged in
the configuration table

TEST 4

INHIBIT MODE POINTER TEST

TEST 5

DIAGNOSTIC MODE DISPATCH ROUTINE
This test runs all the patterns in the
mode selected.

TEST 6

UNIQUE BANK TEST
Pattern 27 is run

TEST 7

FORCE ADDRESS PARITY ERRORS
Insures that no bank responds when there is
an address parity error.

6.2 Patterns

6.2.1 General Pattern Information

Actual patterns are identified by symbolic locations "MTPXYY" where X may be any sub program indicator (A,B,C,etc) or 0 and YY will be the number of the pattern.

Setup procedures for each pattern are identified by symbolic locations "MT00YY" where YY will be the number of the pattern.

Patterns reside in 4 scripts that are scanned for execution. Symbolic location "MKCSRT" is a table of patterns that can run once for each MK11 controller (twice per MK11 Box). Symbolic location "MKPAT" is a table of patterns that can run on each Bank of MK11 memory. Symbolic location "MJPAT" is a table of patterns that can run on each Bank of MJ11 memory. Symbolic location "FSPAT" is a table of patterns that can be run in Field Service Mode (command 5).

The 1st 3 scripts are completely controlled by the APT E-table (even if not running under APT). Modifications to this table can be made (1) with APT, (2) manually, or (3) with ODT reference Section 2.3.4.2.

Example E-table Segment:

```
;THE FOLLOWING LOCATIONS SPECIFY WHICH PATTERNS
;ARE TO BE RUN FOR PARTICULAR MEMORIES
;
;REFERENCE THE TABLE LISTED BELOW TO RELATE BITS TO PATTERNS.
;BIT0 SET WILL RUN THE FIRST ENTRY IN THE TABLE, BIT0 SET
;IN THE SECOND WORD WILL RUN THE 17TH ENTRY IN THE TABLE...
;
;NOTE**NULL TESTS DO NOT TAKE ANY TIME
;
$DDW0:   .WORD   177777   ;MK11 CSR TESTS          TABLE = MKCSRT:
$DDW1:   .WORD   177777   ;MK11 CSR TESTS          TABLE = MKCSRT:
$DDW2:   .WORD   161777   ;MK11 PATTERNS           TABLE = MKPAT:
$DDW3:   .WORD   177777   ;MK11 PATTERNS           TABLE = MKPAT:
$DDW4:   .WORD   177677   ;MJ11 PATTERNS           TABLE = MJPAT:
$DDW5:   .WORD   177777   ;MJ11 PATTERNS           TABLE = MJPAT:
```

6.2.2 Specific Patterns

6.2.2.1 Pattern 0 Basic Data Test

Writes & Reads R2 into a 16K Bank.

This is used for Zeros and Ones testing and in Field Service Mode for any console selected pattern.

It executes out of the USER Instruction PAR's.

NOTE

It is frequently modified dynamically such that (1) it returns after writing only (the 1st NOP is replaced with a RETURN) or (2) it only counts Errors (the code PERR02 and NOP are replaced with INC @#PATERR).

6.2.2.2 Pattern 1 Address Test

Writes & Reads an incrementing pattern equivalent to physical addressed into a 16K Bank.

It executes out of the USER Instruction PAR's.

6.2.2.3 Pattern 2 Complement Address Test

Writes the complement of the physical address from high addresses to low (write down) and reads from low addresses to high (read up).

This provides the complement of the coverage of Pattern 1 in both data pattern and addressing sequence.

It executes out of the USER Instruction PAR's.

6.2.2.4 Pattern 3 3 XOR 9

Writes & Reads a pattern that complements as address bits 4 and 10 change. This is because the PDP-11/70 fetches double words therefore a classical 3 XOR 9 in a small PDP-11 must look like a 4 XOR 10 in a PDP-11/70.

This pattern is run 4 times (1) with Zeros & Ones, (2) with Ones & Zeros, (3) with 401 & Ones, and (4) with Ones & 401. The pattern of the 401 is to force a the parity bits to become involved.

It executes out of the USER Data PDR's, the User Instruction PAR's, the Kernel Data PAR's and the Supervisor Data PAR's.

6.2.2.5 Pattern 4 Rotating Zeros Test

Writes a background pattern of Ones., Rotates a Zero Carry Bit left thru each par of bytes (18 times) and then checks that the carry is Zero and the word (2 bytes) is still all Ones.

It executes out of the User Data PAR's and the Kernel Data PAR's.

NOTE

It is not uncommon to observe the good data equal to the bad data. This indicates that the carry was not clear after 18 ROLB's.

6.2.2.6 Pattern 5 Rotating Ones Test

Writes a background pattern of Zeros. Rotates a One carry bit left thru each pair of bytes (18 times) and then checks that the Carry is a One and the Word (2 Bytes) is still all Zeros.

This provides the complement of the coverage of Pattern 4 in data.

It executes out of the User Data PAR's and the Kernel Data PAR's.

NOTE

It is not uncommon to observe the good data equal the bad data. This indicates that the Carry was not set after 18 ROLB's.

SEQ 0051

6.2.2.7 Pattern 6 INITIAL DATA TEST

Writes & Reads a double word first with all bits 0 except 1 (for every bit position), Second with all bits 1 except 1 (for every bit position).

This is a very quick check of the data paths.

6.2.2.8 Pattern 7 Address Bit Test

Writes a background of all Zeros.
Read Address 1 for a 0 Byte.
Complement Address 1.
Read Address 1 for a non 0 Byte.
for each Address Bit position from Bit 1:
virtual (2, 4, 10, 20, 40, 100, 200, 400, 1000, 2000, 4000, 10000, 60000, 20000)
Physical (60002, 60004, 60010, 60020, 60040, 60100, 60200, 60400, 61000, 62000, 64000, 70000, 140000, 100000)
  Read Address for a 0 word.
  Complement Address contents.
  Read Address for a non-zero word.

This is a very quick check of the address bit uniqueness.

6.2.2.9 Pattern 10 Byte Addressing Test

With ECC Disabled.
Writes all ones to a double word.
For each of the 4 Bytes in the Double Word.
  Clears one byte.
  Reads all 4 bytes from double word.
  Checks for only proper byte clear.
  All other bytes set to all ones.

This is only done on one double word address.

SEQ 0052

6.2.2.10 Pattern 11 Single Bit Error Test

1. Create a Single Bit Error.
2. Read data Uncorrected (with ECC Disable).
3. Read First Word of data corrected (with ECC Enabled)
4. Check that the CSR Single Bit Error Flag was set.
5. Clear SBE Flag.
6. Read Second word of data corrected (with ECC Enabled).
7. Check that the CSR Single Bit Error Flag was set.
8. Do (1-7) for a Single Bit Error in each of 32 positions of a double word.
   i.e. (32 TIMES)
9. If not in Quick Verify Mode then Do (1-8) for data consisting of 1 bit set in each of 32 positions of a double word.
   i.e. (32 X 32 = 1024 Times)
10. Do (1-9) for complemented data (1 Bit clear in each of 32 positions of a double word).
    i.e. (1024 X 2 = 2048 Times)
    or (32 X 2 = 64 Times (Quick Verify))
11. Clear any errors out of test locations.

This insures that all Single Bit Errors can be corrected and detected.

6.2.2.11 Pattern 12 Write Byte Clears SBE Test

1. Create a Single Bit Error.
2. Write a Byte of Double Word to Ones.
3. Read a Byte of Double Word.
4. If Byte that had SBE then the SBE Flag should be OFF.
   If Byte that did not have SBE then SBE Flag should be set.
5. Byte should have been equal to Ones.
6. Do (1-5) for each of the 4 Bytes of the Double Word
7. Do (1-6) for a Single Bit Error in each of 32 positions of a Double Word
   i.e. (32 Times)
8. If not in Quick Verify Mode then do (1-7) for data consisting of 1 Bit set in each of 32 positions of a double word.
   i.e. (32 X 32 = 1024 Times)
9. Clear any errors out of test locations.

This insures that single bit errors in the data portion (not in checkbits) can be cleared by writing the corresponding byte and that writing any other byte does not change the existing single bit error.

6.2.2.12 Pattern 13 Create Double Bit Error Test

1. Create a Double Bit Error.
2. Access the Data (TST instruction).
3. Check that the CSR DBE Flags are both set.
4. Initialize CSR to allow parity traps on DBE's.
5. Access the Data (TST Instruction).
6. Check that a parity trap occurred.
7. Do (1-6) for the 2nd Bit of each Double Bit Error in each of 32 positions of a double word less the one position of the 1st Bad Bit.
   i.e. (31 Times)
8. If not in Quick Verify Mode then Do (1-7) for the 1st Bit of each of Double Bit Error in each of 32 positions of a double word.
   i.e. (31 X 32 = 992 Times)
9. Do (1-8) for complemented data (Ones versus Zeros in Double Word)
   i.e. (992 X 2 = 1984 Times)
   or (31 X 2 = 62 Times (Quick Verify))
10. Clear any errors out of test locations.

This insures that all Double Bit Errors can be created and detected and cause traps.

NOTE

This test is only run during the first (QV) PASS when under ACT or APT.

6.2.2.13 Pattern 14 Write Inhibit During DATIP with DBE Test

1. Create a Double Bit Error.
2. Do ASRB on Test Location.
3. Check that Double Word is STILL Bad (Unchanged-with DBE).
4. Do (2-3) on all 4 Bytes of Double Word.
5. Do (1-4) for the 2nd bit of each Double Bit Error in each of 32 positions of a Double Word less the one position of the 1st Bad Bit.
   i.e. (31 Times)
6. If not in Quick Verify Mode then Do (1-5) for the 1st Bit of each Double Bit Error in each of 32 positions of a double word.
   i.e. (32 X 32 = 922 Times)
7. Do (1-6) for complemented data (Ones versus Zeros in Double Word).
   i.e. (922 X 2 = 1984 Times)
   or (31 X 2 = 62 Times (Quick Verify))
8. Clear any errors out of test locations.

This insures that the Double Bit Error can be cleared by a DATIP to any affected Byte.

NOTE

This test is only run during the first (QV) pass when under ACT or APT.

6.2.2.14 Pattern 15 Write Inhibit of Byte with DBE

1. Create a Double Bit Error.
2. Do a MOVB immediate to test byte.
3. Check that Double Word is still Bad (unchanged-with DBE).
4. Do (2-3) on all 4 Bytes of Double Word.
5. Do (1-4) for the 2nd Bit of each Double Bit Error in each of 32 positions of a double word less the one position of the 1st Bad Bit.
   i.e. (31 Times)
6. If not in Quick Verify Mode then Do (1-5) for the 1st Bit of each Double Bit Error in each of 32 positions of a Double Word.
   i.e. (31 X 32 = 922 Times)
7. Do (1-6) for Complemented Data (Ones versus Zeros in Double Word).
   i.e. (992 X 2 = 1984 Times)
   or (31 X 2 = 62 Times (Quick Verify))
8. Clear any errors out of test locations.

This insures that no Double Bit Error can be cleared by a MOVB to any affected By'e.

NOTE

This test is only run during the first (QV) pass when under ACT or APT.

6.2.2.15 Pattern 16 Write Inhibit of Word with DBE Test

1. Create a Double Bit Error.
2. Do MOV immediate on test location.
3. Check that Double Word is STILL Bad (unchanged-with DBE).
4. Do (2-3) on both Double Words.
5. Do (1-4) for the 2nd Bit of each Double Bit Error in each of 32 positions of a Double Word less the one position of the 1st Bad Bit.
   i.e. (31 Times)
6. If not in Quick Verify Mode then Do (1-5) for the 1st Bit of each Double Bit Error in each of 32 positions of a Double Word.
   i.e. (32 X 32 = 992 Times)
7. Do (1-6) for Complemented Data (Ones versus Zeros in Double Word).
   i.e. (992 X 2 = 1984 Times)
   or (31 X 2 = 62 Times (Quick Verify))
8. Clear any errors out of test locations.

This insures that no Double Bit Error can be cleared by a MOV to any affected word.

NOTE

This test is only run during the first (QV) pass when under ACT or APT.

6.2.2.16 Pattern 17 Holding 1's & 0's Test

1. Write a 16K Bank with alternating Bytes of Zeros & Ones writing a Byte at a time.
2. Read each Word for correct pattern.
3. Do (1-2) again for a complement pattern.

This checks the memory for the capability of holding 0's & 1's.

5.2.2.17 Pattern 20 Marching 0's & 1's in Check Bits Test

1. Write Double Words of 000000,,100000 which causes check bits to equal 100 while addressing increments.
   (Write Up/100 --> check bits)
2. If in Quick Verify Mode then Go to Step (5).
3. REad Double Words & check while writing Zeros and addressing decrements.
   (Down / 100 --> 077)
   This flips all the checkbits.
4. Read Double Words & check while writing 000000,,100000 while addressing increments.
   (Up / 077 --> 100)
5. Read Double Words & check while writing Zeros & addressing increments.
   (Up / 100 --> 077)
6. Read Double Words & check while writing 000000,,100000 while addressing decrements.
   (Down / 077 --> 100)
7. Read Double Words & check while Addressing increments.
   (Up / 100)

This checks the integrity of the MOS chips that store the checkbits.

SEC 0059

5.2.2.18 Pattern 21 Marching 0's & 1's Test

1. Write a Background of alternating Bytes of Zeros & Ones
2. For the 16K Bank addressing Down
   (a) Read check a word
   (b) Byte Swap a word
   (c) Read check a word
3. For the 16K Bank addressing Up
   (a) Read check a word
   (b) Byte Swap a word
   (c) Read check a word
4. For the 16K Bank addressing Up
   (a) Read check a word
   (b) Byte Swap a word
   (c) Read check a word
5. For the 16K Bank addressing Down
   (a) Read check a word
   (b) Byte Swap a word
   (c) Read check a word

This checks the integrity of the 32 Bit Double Words.

It executes out of the User Data PAR's.

6.2.2.19 Pattern 22 Refresh Test

1. Write a diagonal pattern of ones on every KDIAG(TH) stripe & write zeros elsewhere.

   This pattern is on addresses not bit positions.

   Example:

   Address

	MSB's							
LSB's	0	0	0	1	0	0	0	1
	0	0	1	0	0	0	1	0
	0	1	0	0	0	1	0	0
	1	0	0	0	1	0	0	0
	0	0	0	1	0	0	0	1
	0	0	1	0	0	0	1	0
	0	1	0	0	0	1	0	0
	1	0	0	0	1	0	0	0

NOTE

Example uses KDIAG of value 4 more typical is a value of 8. Consult the symbolic definition of "KDIAG" in the program listing to be sure.

2. Disturb each row for >2ms
3. Read check diagonal pattern
4. Do (1-3) KDIAG times moving the placement of the diagonal stripe to cover all address positions.
5. Do (1-4) for a complement pattern (zeros in a background of ones)

This test is not in the normal script of execution but may be added via the APT E-TABLE, reference symbolic locations "MKPAT", "$DDW2-5". Field Service Mode command 11 is the normal method of running this pattern.

SEQ 0061

6.2.2.20 Pattern 23 Shifting Diagonal Pattern Test

Similar in overall operation to pattern 22 except

1. uses patterns of 0's & 1's in alternating Bytes.
2. uses code that gives an effect of using a KDIAG of 64 (one stripe every 64th call) or in other words at 4K boundaries only 1 stripe.
3. operates on 4K Boundaries and hence runs 4 times.
4. Takes forever (6.9 Seconds)

NOTE

This pattern is not in the normal script of execution but may be added via the APT E-Table reference symbolic locations "MKPAT," "MJPAT", "$DDW2-5".

6.2.2.21 Pattern 24 Fast Galloping Pattern Test

This does a classical galloping pattern except that addressing is incremented by 200 Octal (every 64 call)

This test takes 58 seconds to run.

NOTE

This pattern is not in the normal script of execution but may be added via the APT E-Table reference symbolic locations "MKPAT", "MJPAT", "$DDW2-5".

6.2.2.22 Pattern 25 Interrupt Enable Test

1. Set CSR to Allow DBE Traps.
2. Access Test Double Words.
3. Check for no Double Bit Error Trap.
4. Enable CSR for SBE Traps.
5. Access Test Double Words.
6. Check for no SBE Trap.
7. Write a SBE in 1 Byte.
8. Disable CSR for Parity Traps.
9. Access Test Double Words.
10. Check for no Traps.
11. Enable CSR for SBE Traps.
12. Access Test Double Words.
13. Check to Insure Trap Occurred.
14. do (7-13) for the 3 other Bytes in the Double Word.
15. Create a DBE in 1 Byte.
16. Disable CSR for Parity Traps.
17. Access the Test Double Word.
18. Check for no Traps.
19. Enable CSR for DBE Traps.
20. Access the Test Double Word.
21. Check to Insure Trap Occurred.
22. Enable CSR for SBE Traps.
23. Access the Test Double Word.
24. Check to Insure Trap Occurred.
25. Do (15-24) for the 3 other Bytes in the Double Word.

This insures that SBE's & DBE's give the correct type of traps.

6.2.2.23 Pattern 26 Random Data Test

Write Random Data in a 16K Bank while incrementing the Addresses.

Read check Random Data.

This routine regenerates the same random numbers by using the same seed as the write sequence. After the read check the seed is updated so that the next use of this pattern will not invoke the same sequence of random numbers.

If you wish to change the random sequence so that it is different than any other run in the same configuration then there are 2 ways of doing so.

1. Modify symbolic locations "SEEDHI" and "SEEDLO" to any number you like.
2. Enter Field Service Mode and execute this pattern (command 5) on some (any good) bank for a short time (30 sec or so).

This executes out of the User Data PAR's, the Kernel Data PAR's, and the Supervisor Data PAR's.

6.2.2.24 Pattern 27 Unique Bank Test

This pattern uses Pattern 0 to write & read the Bank number in each bank.

It does not test Banks that require relocation to test.

It does not run as part of any script but rather is always run after normal pattern tests are complete.

6.2.2.25 Pattern 30 Flush out DBE's Test

This Reads each location then moves the old value back in. This is done with ECC Disabled and therefore corrects any DBE's or SBE's (if possible).

It does not run as part of any script but rather is always run just prior to the End of Pass Code, as part of a Control "C" (Boot) command, as part of End of Pass shutdown for ACT or XXDP Chain Mode, as part of hanging sequence after an error if under ACT or APT, and as part of a shutdown sequence directed by Switch 8 (Halt Program).

6.2.2.26 Pattern 31 SOB-A-LONG Test

Rationalization

In order to concentrate the memory cycles of a test into a particular address, we must cut the overhead cycles to a minimum. Frequently, the instruction itself may provide adequate data or set up a background in which any complemented bit may find it hard to survive.

The SOB instruction is the only PDP-11 instruction that is (1) a single operand, (2) can be repeatedly executed at the same PC and, (3) can escape this repetitious loop.

Hence, I believe it is possible to SOB a MOS cell to death (or at least brain wash him), and to SOB a core into over-heating (or at least warm discomfort).

The SOB Routine will be loaded and called with R0 set equal to the SOB constant "SOBK", R1 set equal to the complement of a "SOB R0,." Instruction "100776".

Simplified SOB Example:

```
1$:     SOB     R0,1$   ;SOB till R0 underflows
        MOV     R1,1$   ;Write complement of SOB
        CMP     R1,1$   ;Read & check not SOB
        BEQ       2$    ;Skip if OK
        SOBFAIL         ;Trap & report error
2$:     SOBMOV1         ;Code to get self moved
        SOBMOV2         ;Forward 1 word and run again
        SOBMOV3
        SOBMOV4
        SOBMOV...
```

The value of the SOB constant can be found at symbolic location "SOBK" (typical 25 decimal).

This test is not in the normal script of execution but may be added via the APT E-TABLE, reference symbolic locations "MKPAT", "MJPAT", "$DDW2-5". Field Service Mode command B is the normal method of running this pattern.

NOTE

Since this test executes out of the memory under test most errors detected will be fatal to the program. The display register will indicate which bank is at Fault (unless clobbered by crash).

6.2.2.27 Pattern 999 Null Test

This is an instant return added to preserve the software structure.

This pattern replaces any real patterns when the APT E-Table does not specify a pattern to be run.

7.0 PROGRAM FEATURES

7.1 Fast Data Access Rates

One of the main areas of concern in testing memory in systems environments is speed. It is my belief that one of the prime reasons that system programs like RSTS, IAS and MUMPS can crash due to memory failures not detectable by memory diagnostics (0-124K, 0-2 MEG, etc.) is because of multiple NPR devices contending for the bus. After some delay a NPR device becomes bus master with his silo backed up (perhaps only 3-8 words) and does several memory transfers at memory data rates.

On the other hand most diagnostics when writing reading and/or checking patterns spend most of their time fetching instructions and operands out of their program space and proportionally little time accessing the memory under test.

I have attempted to optimize this diagnostic's error detecting abilities around the primary design criteria of speed. To this end I have taken the following steps.

7.1.1 Fast City

Utilization of Memory Management Registers as Non Memory Bus, Non UNIBUS, Bipolar Memory. Since user mode is only used for relocation and data space is never used then subroutines can be executed from the UIPAR's, UDPAR's, KDPAR's, SDPAR's and with some Bit Pattern restrictions the UIPDR's, UDPDR's, KDPDR's, and SDPDR's.

The program runs in Kernel mode and Patterns are executed in Supervisor mode for mapping purposes. All core patterns and some MOS Patterns are subroutines that are moved to this Bipolar region referred to in the program as Fast City.

7.1.2 SOB's

Utilization of the PDP-11/70 Instruction Set to speed pattern algorithms (principally the SOB). No attempt is made to support anything less than a PDP-11/70.

7.1.3 CACHE

CACHE is used between pattern tests to decrease program pass times. CACHE can be defeated or partially disabled by the operator (reference section 2.3.3.1).

7.2 Bank Zero Testing

Bank Zero has been traditionally neglected by memory diagnostics for the following reason.

The vector space exists there and ALL traps must not access test pattern data. If the area is tested the diagnostic must not use any traps and it is against the rules for power to fail.

Systems with memory management can overcome this because all traps are to Kernel Virtual space even power fail (caution must be observed because power up goes to physical address 24 (because the Memory Management Unit comes up off)).

However, Catch 22 is that the diagnostic is not APT compatible in this mode because APT Accesses Physical Memory Locations.

PDP-11/70 Systems can over come this because the UNIBUS MAP can fool APT.

Because of the previous arguments this program does not relocate in the true since of the word (i.e. no position independent code was written (at least not on purpose)), but rather this program moves and remaps (hereafter referred to as relocates). This enables the complete testing of Bank Zero or any other program space or privileged space exactly as all other banks are tested. (The conditional test to see if a bank is protected is complemented when relocated).

7.3 Differences In Core Testing With 0-2 MEG (DEMJA)

7.3.1 Delta noise test only performed with 3 XOR 9 Test

Reason: DEMJA performed the delta noise test with constant data however, it will only have an adverse effect if performed with a worst-case pattern.

7.3.2 8 XOR 13 Test deleted

Reason: Designed for a PDP11-20 Memory not presently in use.

7.3.3 3 XOR 9 Rewritten

Reason: was actually a 4 XOR 9 and should have been a 4 XOR 10 for 11/70's to account for reading Double Words.

7.4 Memory Configuration Map

This map is printed out immediately after sizing the memory if SW6 is set (reference section 2.3.1). It can also be printed at any later time in Field Service Mode (reference section 2,3,4,3,7)

Example:

```
                    MEMORY CONFIGURATION MAP
                           16K BANKS

                1       2       3       4       5       6       7
        01234567012345670123456701234567012345670123456701234567 0123
ERRORS                  xx
ACCESSED11111111111111111111111111111111000000000000000000000000000000
INTRLV  22222222222222222222222222222222????????????????????????????
MEMTYPE KKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKOOOOOOOOOOOOOOOOOOOOOOOOOOOOOO
BOX     00000000000000000000000000000000????????????????????????????
PROTECT PPK
            0       1       2       3       4       5       6
        45670123456701234567012345670123456701234567012345670123456 7
ERRORS
ACCESSED00000000000000000000000000000000000000000000000000000000000000
INTRLV  ????????????????????????????????????????????????????????????
MEMTYPE 00000000000000000000000000000000000000000000000000000000000000
BOX     ????????????????????????????????????????????????????????????
PROTECT
```

Displayed are Banks 0-167 Octal (2 meg words). If the Fat Terminal Switch was set (reference section 2.3.1) then all Banks would fit on one line.

The sizing routine could not write zeros and ones in Banks 20 & 21 hence they are marked as bad with X's.

The CPU was able to access banks 0-37 (512 K words).

The interleaving on Banks 0-37 is 2 way. This may be external with no internal or internal with no external - however since there is only one box of MOS (BOX 0) it must be internal interleave.

Banks 0-37 are Memory Type K (MK11) and Banks 40-167 do not exist. Memory Type J would indicate MJ11 memory.

Banks 0-37 are located in box 0. Boxes are assigned to CSR locations in order and MJ11 Boxes will never have a Box number only a question mark.

Banks 0, 1 and 2 are protected. Bank 0 & 1 are protected because they are program space. Bank 2 is protected because it is in the bottom 16K of an MK11 controller. Bank 0 is also protected for this same reason however, the protection is hierarchical and program space overshadows MK11 protection. Banks 0, 1 & 2 will not be tested until the program relocates.

7.5 Banks & Boards (MK11)

Boards with 4K MOS chips are 32K words each.
Boards with 16K MOS chips are 128K words each.

Imagine that the front view of the memory box is as follows:

BL,15,13,11,9,7,5,3,1,CB,CA,DA,BL,AD,BL,CA,CB,0,2,4,6,8,10,12,14,BL

Where 0 thru 15 are the MOS array cards. CB is Control "B". CA is Control "A". DA is the Data Interface. AD is the Address Interface. BL is a Blank (unwired) slot in the backplane.

SEQ 0070

4K Parts with No Interleave

Array Banks	15	13	11	9	7	5	3	1	0	2	4	6	8	10	12	14
	36	32	26	22	16	12	6	2	0	4	10	14	20	24	30	34
	37	33	27	23	17	13	7	3	1	5	11	15	21	25	31	35

4K Parts with Internal Interleave

Array Banks	15	13	11	9	7	5	3	1	0	2	4	6	8	10	12	14
	34	30	24	20	14	10	4	0	0	4	10	14	20	24	30	34
	35	31	25	21	15	11	5	1	1	5	11	15	21	25	31	35
	36	32	26	22	16	12	6	2	2	6	12	16	22	26	32	36
	37	33	27	23	17	13	7	3	3	7	13	17	23	27	33	37

4K Parts with 2-1 External & No Internal Interleave

Array Banks	15	13	11	9	7	5	3	1	0	2	4	6	8	10	12	14
	74	64	54	44	34	24	14	4	0	10	20	30	40	50	60	70
	75	65	55	45	35	25	15	5	1	11	21	31	41	51	61	71
	76	66	56	46	36	26	16	6	2	12	22	32	42	52	62	72
	77	67	57	47	37	27	17	7	3	13	23	33	43	53	63	73

4K Parts with 2-1 External & Internal Interleave

Array Banks	15	13	11	9	7	5	3	1	0	2	4	6	8	10	12	14
	70	60	50	40	30	20	10	0	0	10	20	30	40	50	60	70
	71	61	51	41	31	21	11	1	1	11	21	31	41	51	61	71
	72	62	52	42	32	22	12	2	2	12	22	32	42	52	62	72
	73	63	53	43	33	23	13	3	3	13	23	33	43	53	63	73
	74	64	54	44	34	24	14	4	4	14	24	34	44	54	64	74
	75	65	55	45	35	25	15	5	5	15	25	35	45	55	65	75
	76	66	56	46	36	26	16	6	6	16	26	36	46	56	66	76
	77	67	57	47	37	27	17	7	7	17	27	37	47	57	67	77

4K Parts with 4-1 External & No Internal Interleave

Array Banks	15	13	11	9	7	5	3	1	0	2	4	6	8	10	12	14
	150	130	110	70	50	30	10	0	20	40	60	100	120	140	160	
	152	132	112	72	52	32	12	2	22	42	62	102	122	142	162	
	152	133	113	73	53	33	13	3	23	43	63	103	123	143	163	
	154	134	114	74	54	34	14	4	24	44	64	104	124	144	164	
	155	135	115	75	55	35	15	5	25	45	65	105	125	145	165	
	156	136	116	76	56	36	16	6	26	46	66	106	126	146	166	
	157	137	117	77	57	37	17	7	27	47	67	107	127	147	167	

4K Parts with 4-1 External & Internal Interleave

Array Banks	15	13	11	9	7	5	3	1	0	2	4	6	8	10	12	14
	160	140	120	100	60	40	20	0	0	20	40	60	100	120	140	160
	161	141	121	101	61	41	21	1	1	21	41	61	101	121	141	161
	162	142	122	102	62	42	22	2	2	22	42	62	102	122	142	162
	163	143	133	103	63	43	23	3	3	23	43	63	103	123	143	163
	164	144	124	104	64	44	24	4	4	24	44	64	104	124	144	164
	165	145	125	105	65	45	25	5	5	25	45	65	105	125	145	165
	166	146	126	106	66	46	26	6	6	26	46	66	106	126	146	166
	167	147	127	107	67	47	27	7	7	27	47	67	107	127	147	167
		150	130	110	70	50	30	10	10	30	50	70	110	130	150	
		151	131	111	71	51	31	11	11	31	51	71	111	131	151	
		152	132	112	72	52	32	12	12	31	52	72	112	132	152	
		153	133	113	73	53	33	13	13	33	53	73	113	133	153	
		154	134	114	74	54	34	14	14	34	54	74	114	134	154	
		155	135	115	75	55	35	15	15	35	55	75	115	135	155	
		156	136	116	76	56	36	16	16	36	56	76	116	136	156	
		157	137	117	77	57	37	17	17	37	57	77	117	137	157	

16K Parts with No Interleave

Array Banks	15	13	11	9	7	5	3	1	0	2	4	6	8	10	12	14
		150	130	110	70	50	30	10	0	20	40	60	100	120	140	160
		151	131	111	71	51	31	11	1	21	41	61	101	121	141	161
		152	132	112	72	52	32	12	2	22	42	62	102	122	142	162
		153	133	113	73	53	33	13	3	23	43	63	103	123	143	163
		154	134	114	74	54	34	14	4	24	44	64	104	124	144	164
		155	135	115	75	55	35	15	5	25	45	65	105	125	145	165
		156	136	116	76	56	36	16	6	26	46	66	106	126	146	166
		157	137	117	77	57	37	17	7	27	47	67	107	127	147	167

16K Parts with Internal Interleave

Array Banks	15	13	11	9	7	5	3	1	0	2	4	6	8	10	12	14
	160	140	120	100	60	40	20	0	0	20	40	60	100	120	140	160
	161	141	121	101	61	41	21	1	1	21	41	61	101	121	141	161
	162	142	122	102	62	42	22	2	2	22	42	62	102	122	142	162
	163	143	123	103	63	43	23	3	3	23	43	63	103	123	143	163
	164	144	124	104	64	44	24	4	4	24	44	64	104	124	144	164
	165	145	125	105	65	45	25	5	5	25	45	65	105	125	145	165
	166	146	126	106	66	46	26	6	6	26	46	66	106	126	146	166
	167	147	127	107	67	47	27	7	7	27	47	67	107	127	147	167
		150	130	110	70	50	30	10	10	30	50	70	110	130	150	
		151	131	111	71	51	31	11	11	31	51	71	111	131	151	
		152	132	112	72	52	32	12	12	32	52	72	112	132	152	
		153	133	113	73	53	33	13	13	33	53	73	113	133	153	
		154	134	114	74	54	34	14	14	34	54	74	114	134	154	
		155	135	115	75	55	35	15	15	35	55	75	115	135	155	
		156	136	116	76	56	36	16	16	36	56	76	116	136	156	
		157	137	117	77	57	37	17	17	37	57	77	117	137	157	

16K Parts with 2-1 External & No Internal Interleave

Array Banks	15.	13.	11.	9.	7.	5.	3.	1.	0.	2.	4.	6.	8.	10.	12.	14
					160	120	60	20	0	40	100	140				
					161	121	61	21	1	41	101	141				
					162	122	62	22	2	42	102	142				
					163	123	63	23	3	43	103	143				
					164	124	64	24	4	44	104	144				
					165	125	65	25	5	45	105	145				
					166	126	66	26	6	46	106	146				
					167	127	67	27	7	47	107	147				
						130	70	30	10	50	110	150				
						131	71	31	11	51	111	151				
						132	72	32	12	52	112	152				
						133	73	33	13	53	113	153				
						134	74	34	14	54	114	154				
						135	75	35	15	55	115	155				
						136	76	36	16	56	116	156				
						137	77	37	17	57	117	157				

16K Parts with 2-1 External & Internal Interleave

Array Banks	15.	13.	11.	9.	7.	5.	3.	1.	0.	2.	4.	6.	8.	10.	12.	14
					140	100	40	0	0	40	100	140				
					141	101	41	1	1	41	101	141				
					142	102	42	2	2	42	102	142				
					143	103	43	3	3	43	103	143				
					144	104	44	4	4	44	104	144				
					145	105	45	5	5	45	105	145				
					146	106	46	6	6	46	106	146				
					147	107	47	7	7	47	107	147				
					150	110	50	10	10	50	110	150				
					151	111	51	11	11	51	111	151				
					152	112	52	12	12	52	112	152				
					153	113	53	13	13	53	113	153				
					154	114	54	14	14	54	114	154				
					155	115	55	15	15	55	115	155				
					156	116	56	16	16	56	116	156				
					157	117	57	17	17	57	117	157				
					160	120	60	20	20	60	120	160				
					161	121	61	21	21	61	121	161				
					162	122	62	22	22	62	122	162				
					163	123	63	23	23	63	123	163				
					164	124	64	24	24	64	124	164				
					165	125	65	25	25	65	125	165				
					166	126	66	26	26	66	126	166				
					167	127	67	27	27	67	127	167				
						130	70	30	30	70	130					
						131	71	31	31	71	131					
						132	72	32	32	72	132					
						133	73	33	33	73	133					
						134	74	34	34	74	134					
						135	75	35	35	75	135					
						136	76	36	36	76	136					
						137	77	37	37	77	137					

16k Parts with 4-1 External & No Internal Interleave

Array Banks	15.	13.	11.	9.	7.	5.	3.	1.	0.	2.	4.	6.	8.	10.	12.	14
							140	40	0	100						
							141	41	1	101						
							142	42	2	102						
							143	43	3	103						
							144	44	4	104						
							145	45	5	105						
							146	46	6	106						
							147	47	7	107						
							150	50	10	110						
							151	51	11	111						
							152	52	12	112						
							153	53	13	113						
							154	54	14	114						
							155	55	15	115						
							156	56	16	116						
							157	57	17	117						
							160	60	20	120						
							161	61	21	121						
							162	62	22	122						
							163	63	23	123						
							164	64	24	124						
							165	65	25	125						
							166	66	26	126						
							167	67	27	127						
								70	30	130						
								71	31	131						
								72	32	132						
								73	33	133						
								74	34	134						
								75	35	135						
								76	36	136						
								77	37	137						

16K Parts with 4-1 External & Internal Interleaving

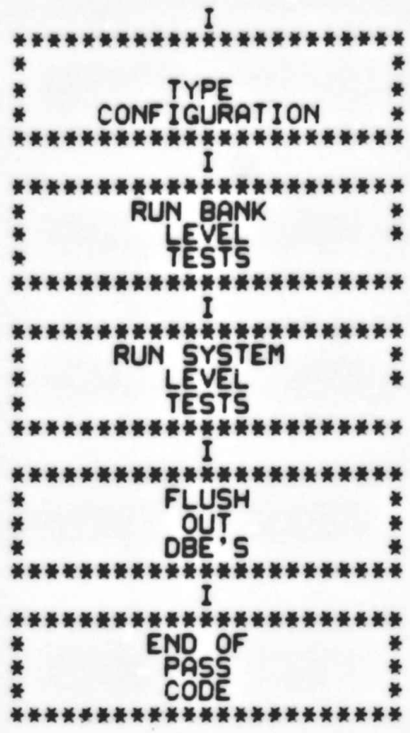
Array Banks	15,	13,	11,	9,	7,	5,	3,	1,	0,	2,	4,	6,	8,	10,	12,	14
							100	0	0	100						
							101	1	1	101						
							102	2	2	102						
							103	3	3	103						
							104	4	4	104						
							105	5	5	105						
							106	6	6	106						
							107	7	7	107						
							110	10	10	110						
							111	11	11	111						
							112	12	12	112						
							113	13	13	113						
							114	14	14	114						
							115	15	15	115						
							116	16	16	116						
							117	17	17	117						
							120	20	20	120						
							121	21	21	112						
							122	22	22	122						
							123	23	23	123						
							124	24	24	124						
							125	25	25	125						
							126	26	26	126						
							127	27	27	127						
							130	30	30	130						
							131	31	31	131						
							132	32	32	132						
							133	33	33	133						
							134	34	34	134						
							135	35	35	135						
							136	36	36	136						
							137	37	37	137						
							140	40	40	140						
							141	41	41	141						
							142	42	42	142						
							143	43	43	143						
							144	44	44	144						
							145	45	45	145						
							146	46	46	146						
							147	47	47	147						
							150	50	50	150						
							151	51	51	151						
							152	52	52	152						
							153	53	53	153						
							154	54	54	154						
							155	55	55	155						
							156	56	56	156						
							157	57	57	157						
							160	60	60	160						
							161	61	61	161						
							162	62	62	162						
							163	63	63	163						

164 64 64 164
165 65 65 165
166 66 66 166
167 67 67 167
70 70
71 71
72 72
73 73
74 74
75 75
76 76
77 77

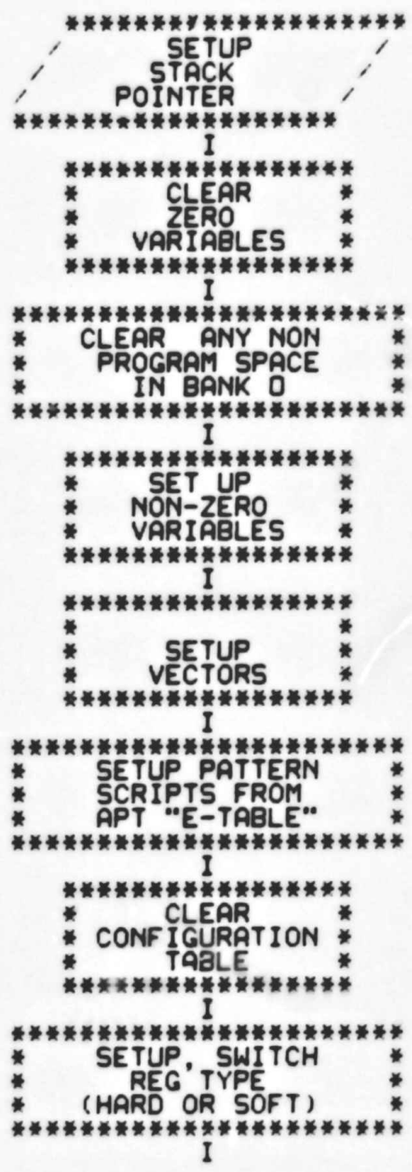
8.0 FLOW CHARTS

8.1 Main Program

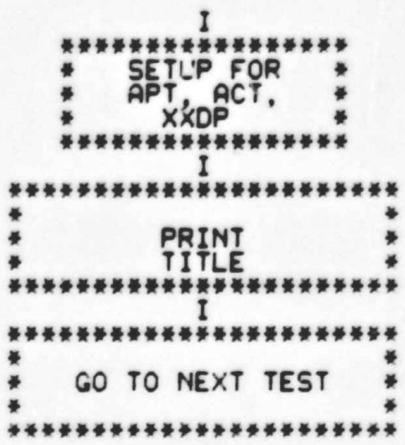
SEG 0077


TYPE CONFIGURATION
RUN BANK LEVEL TESTS
RUN SYSTEM LEVEL TESTS
FLUSH OUT DBE'S
END OF PASS CODE

8.1.1 Initialize -



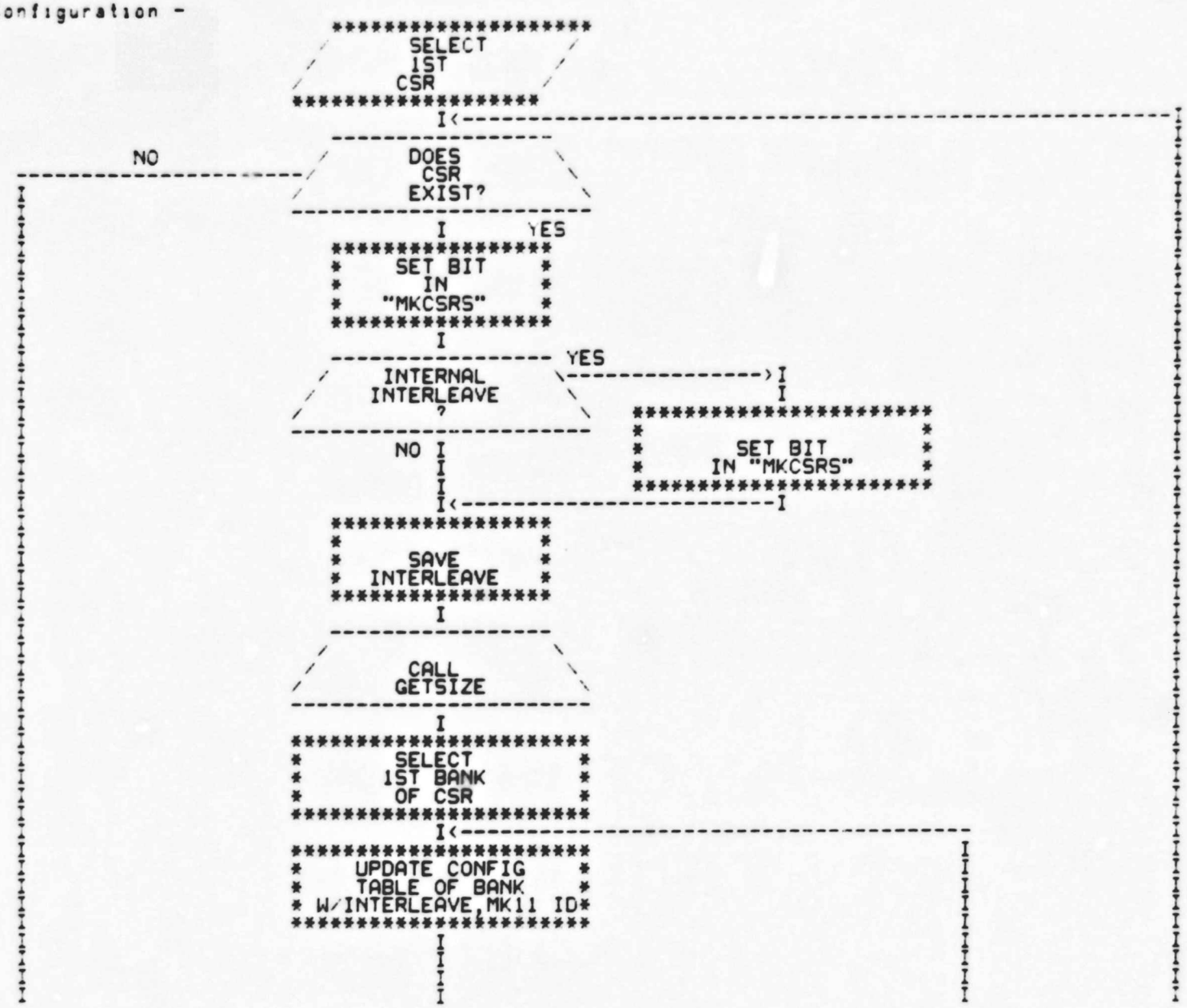
SEQ 0079



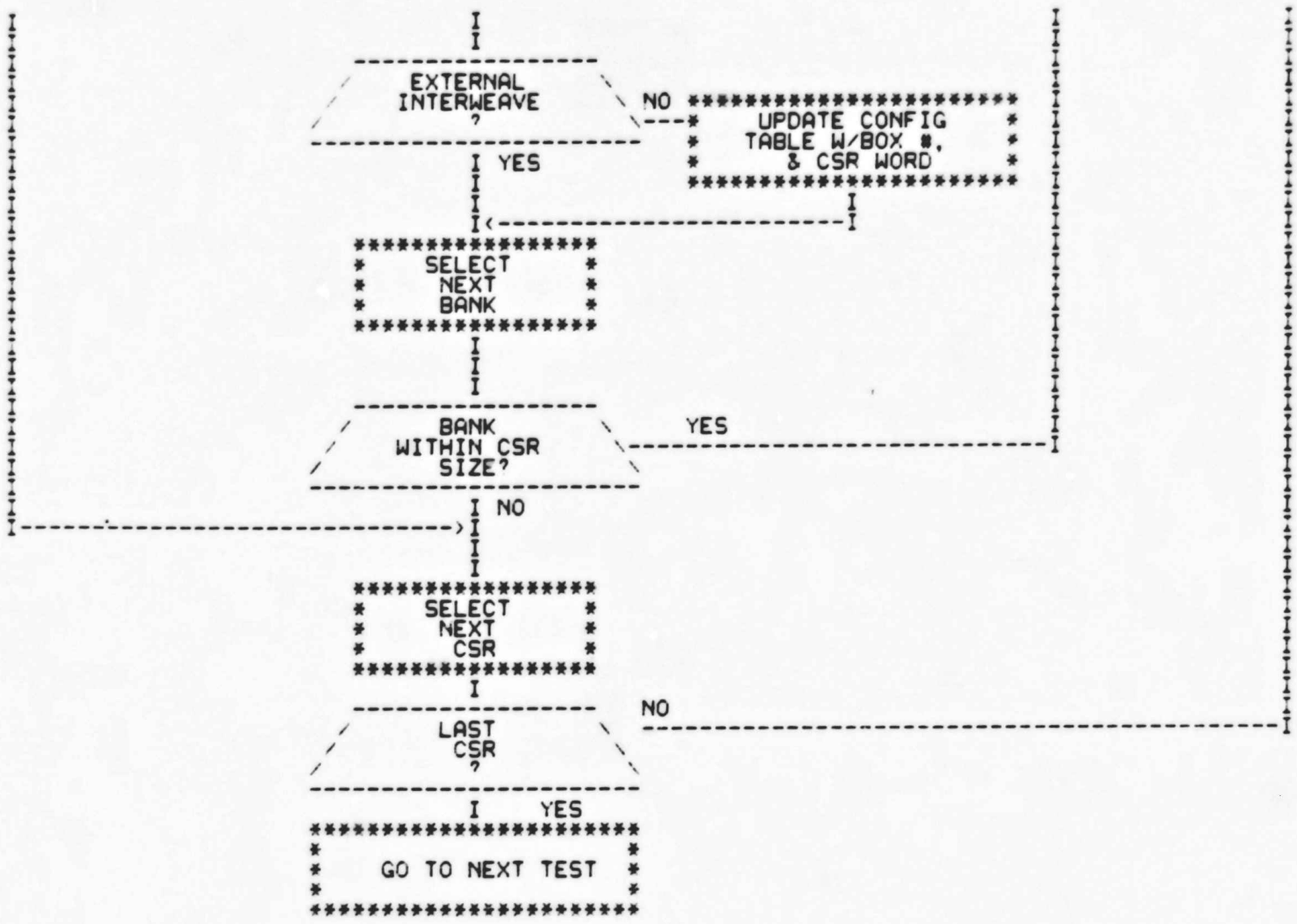
SEQ 0080

8.1.2 Test Hardcore Hardware -

8.1.3 Analize Configuration -

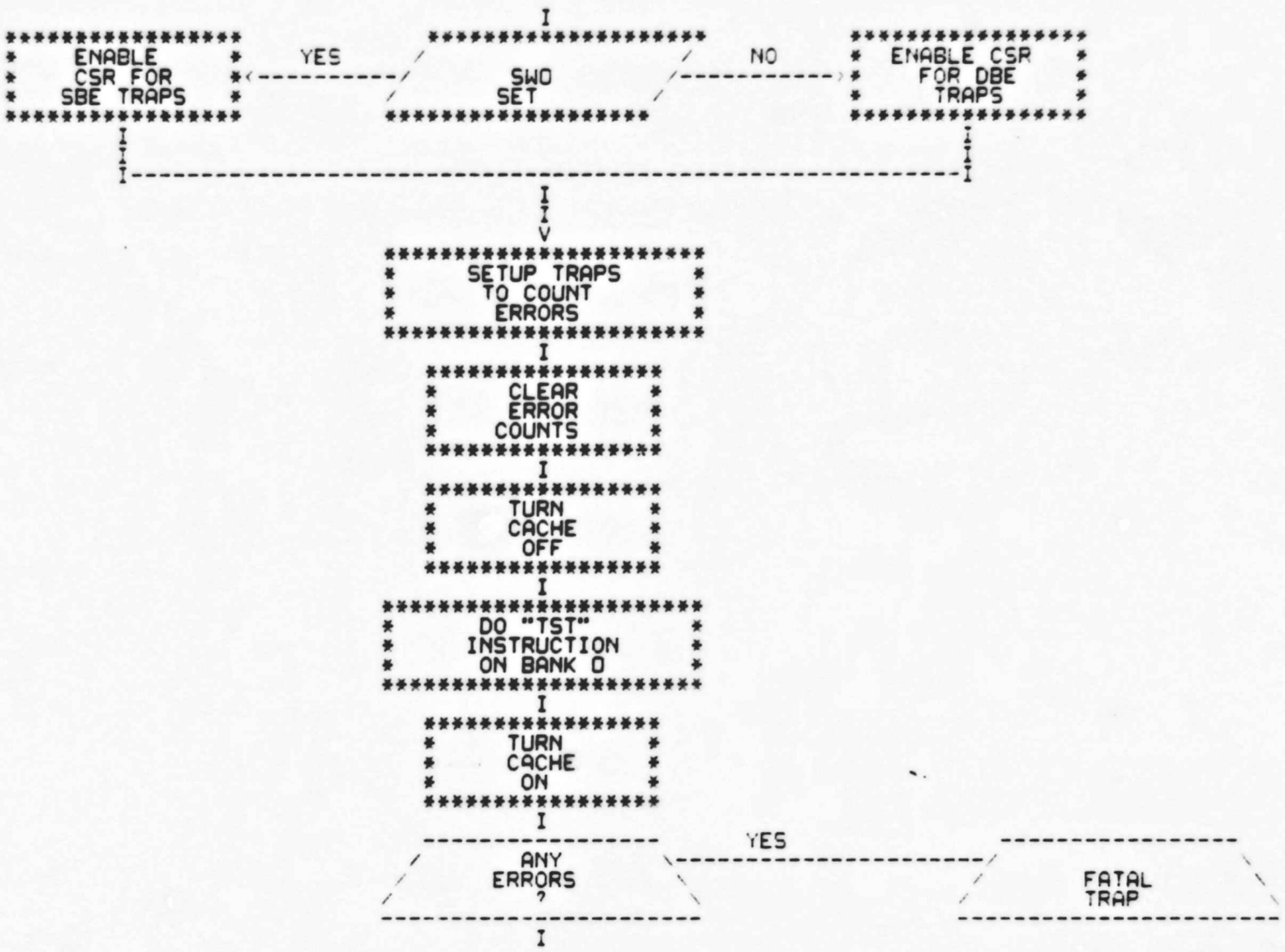


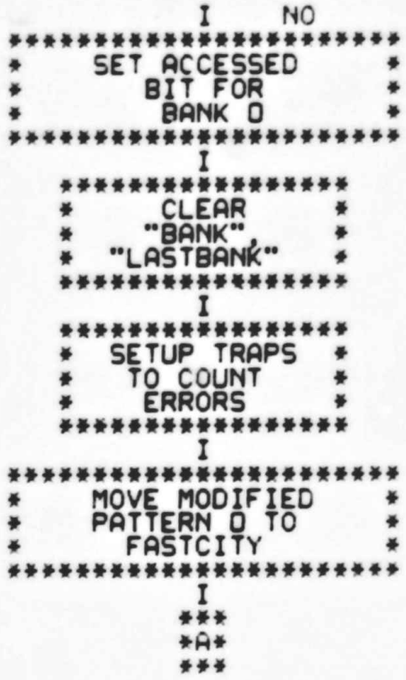
SEQ 0082


EXTERNAL INTERWEAVE ?
NO
UPDATE CONFIG TABLE W/BOX #, & CSR WORD
YES
SELECT NEXT BANK
BANK WITHIN CSR SIZE?
YES
NO
SELECT NEXT CSR
LAST CSR ?
NO
YES
GO TO NEXT TEST

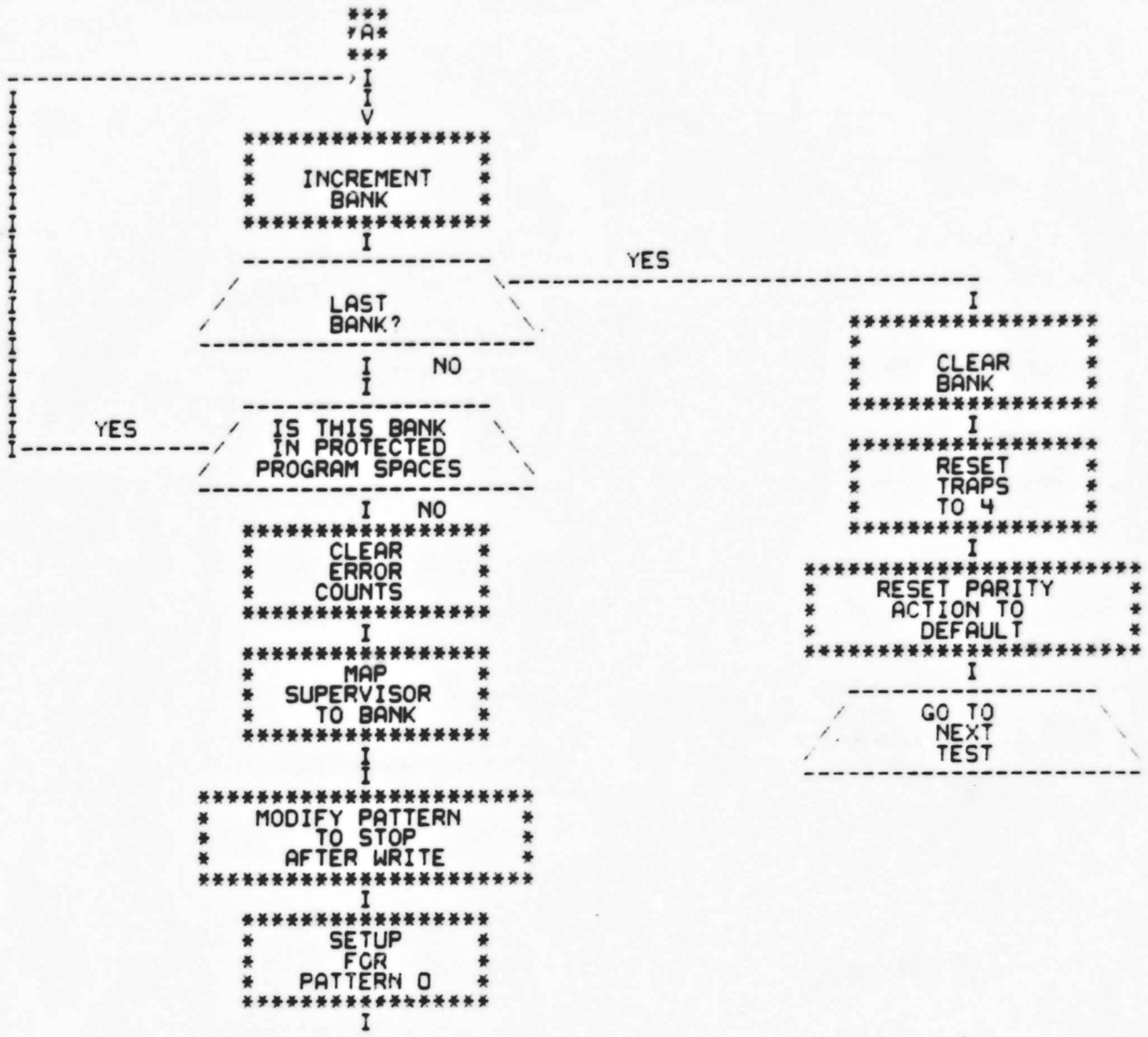
SEQ 0083

8.1.4 Size & Test 0's & 1's On Each Bank -

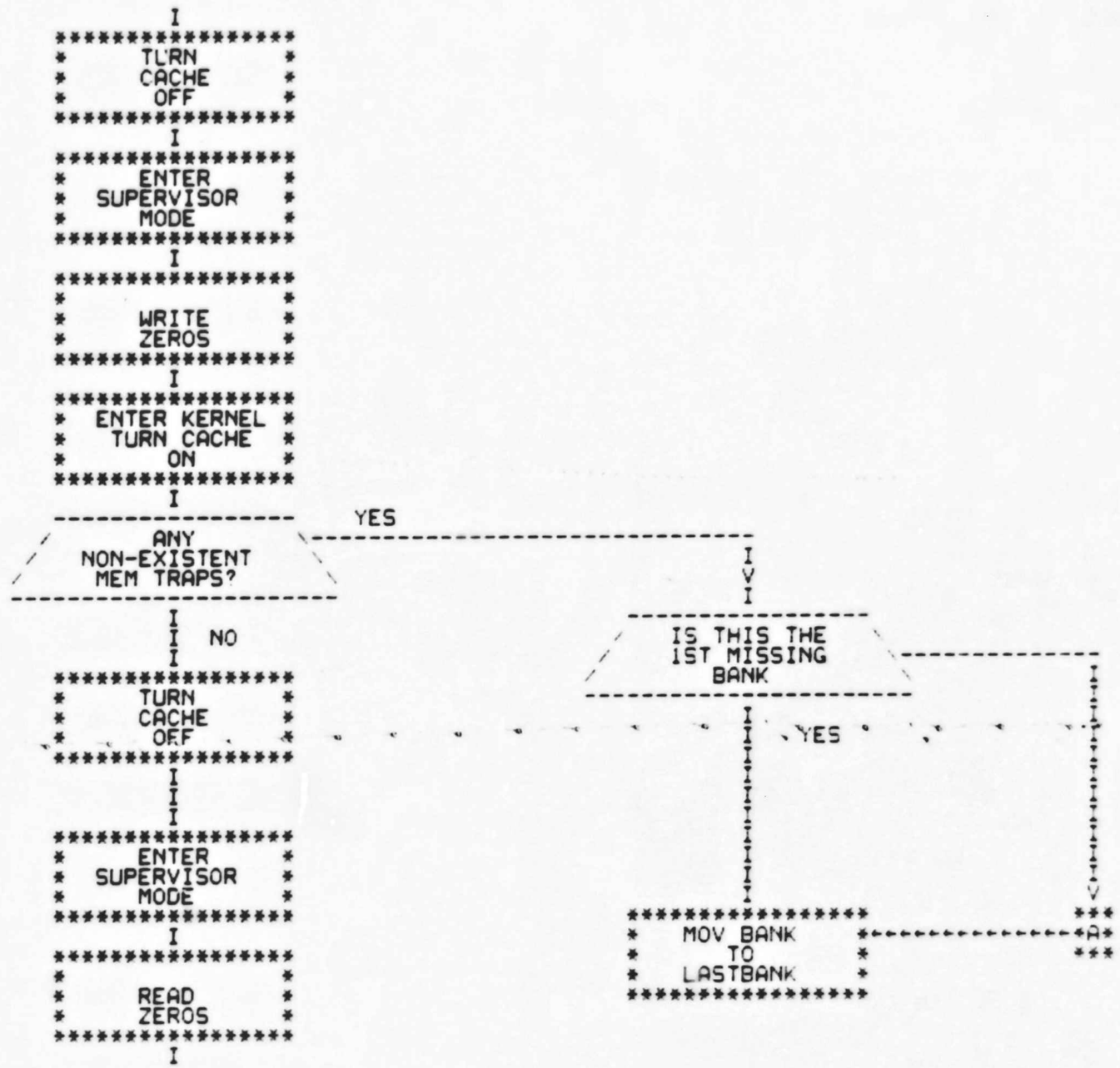


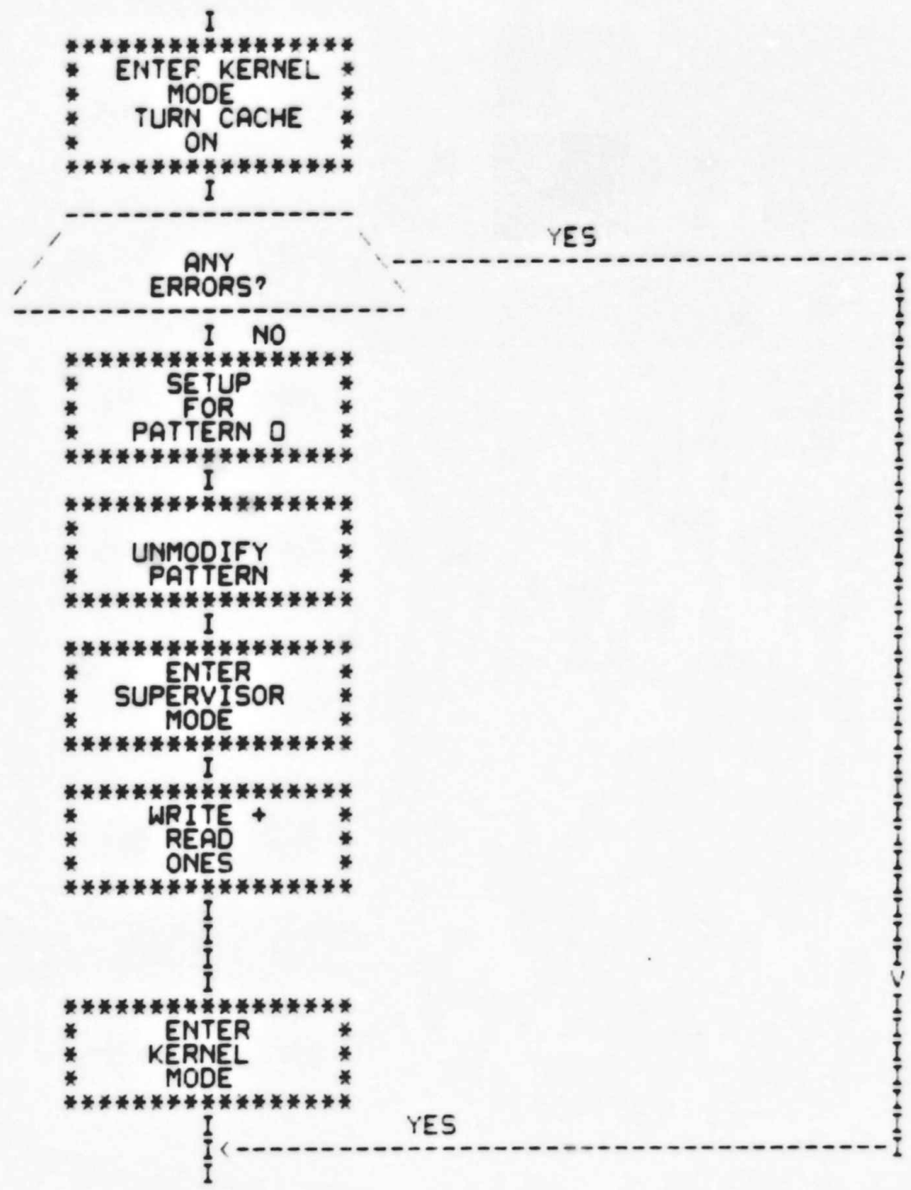

I NO
SET ACCESSED BIT FOR BANK 0
CLEAR "BANK", "LASTBANK"
SETUP TRAPS TO COUNT ERRORS
MOVE MODIFIED PATTERN 0 TO FASTCITY
A

SEQ 0085

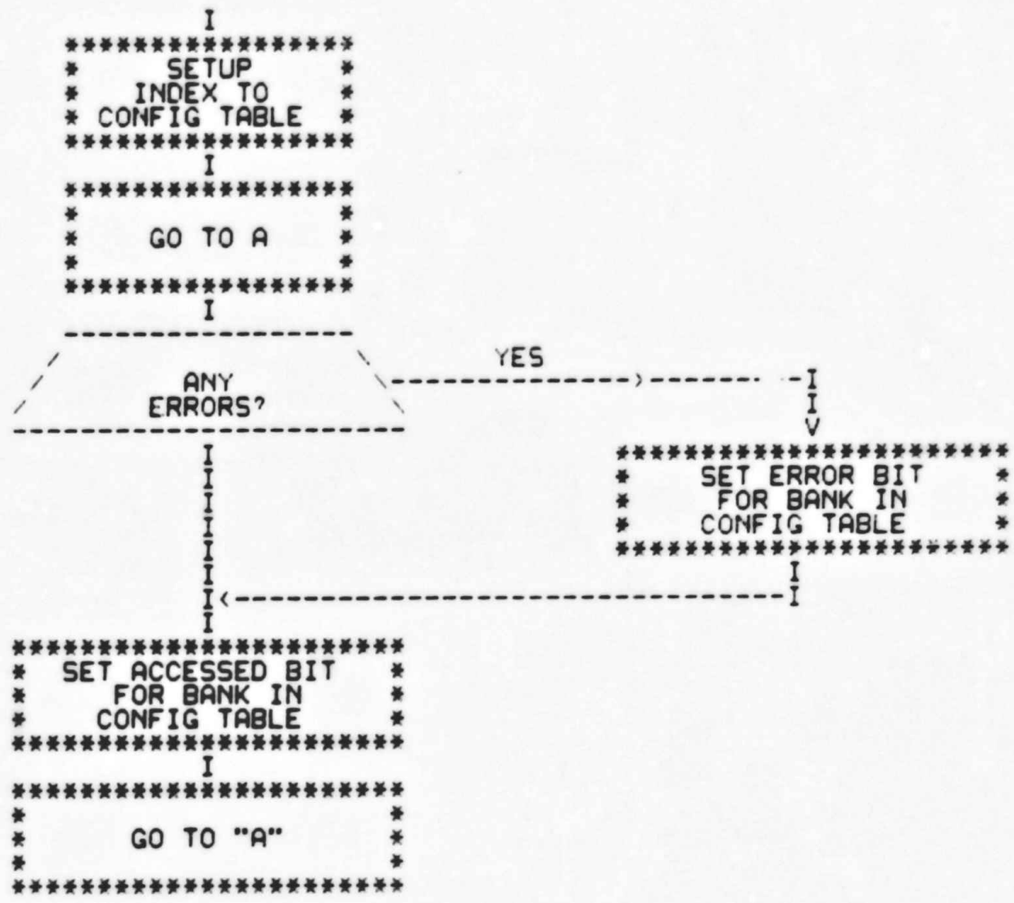

A
INCREMENT BANK
LAST BANK?
YES
NO
IS THIS BANK IN PROTECTED PROGRAM SPACES
YES
NO
CLEAR ERROR COUNTS
MAP SUPERVISOR TO BANK
MODIFY PATTERN TO STOP AFTER WRITE
SETUP FOR PATTERN 0
CLEAR BANK
RESET TRAPS TO 4
RESET PARITY ACTION TO DEFAULT
GO TO NEXT TEST

SEQ 0086

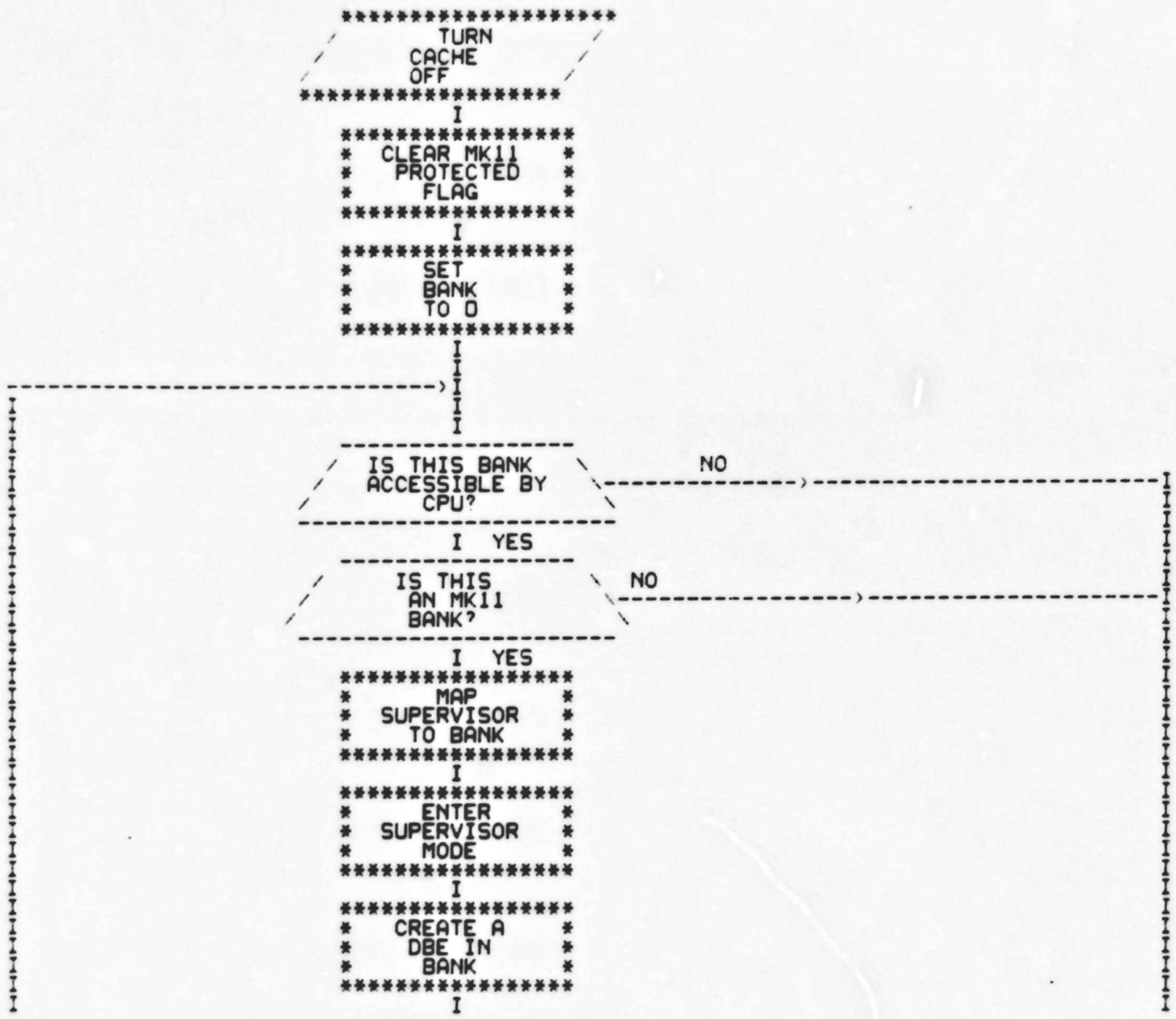

TURN CACHE OFF
ENTER SUPERVISOR MODE
WRITE ZEROS
ENTER KERNEL TURN CACHE ON
ANY NON-EXISTENT MEM TRAPS?
YES
NO
IS THIS THE 1ST MISSING BANK
YES
MOV BANK TO LASTBANK
A
TURN CACHE OFF
ENTER SUPERVISOR MODE
READ ZEROS


ENTER KERNEL MODE
TURN CACHE ON
ANY ERRORS?
YES
NO
SETUP FOR PATTERN 0
UNMODIFY PATTERN
ENTER SUPERVISOR MODE
WRITE + READ ONES
ENTER KERNEL MODE
YES

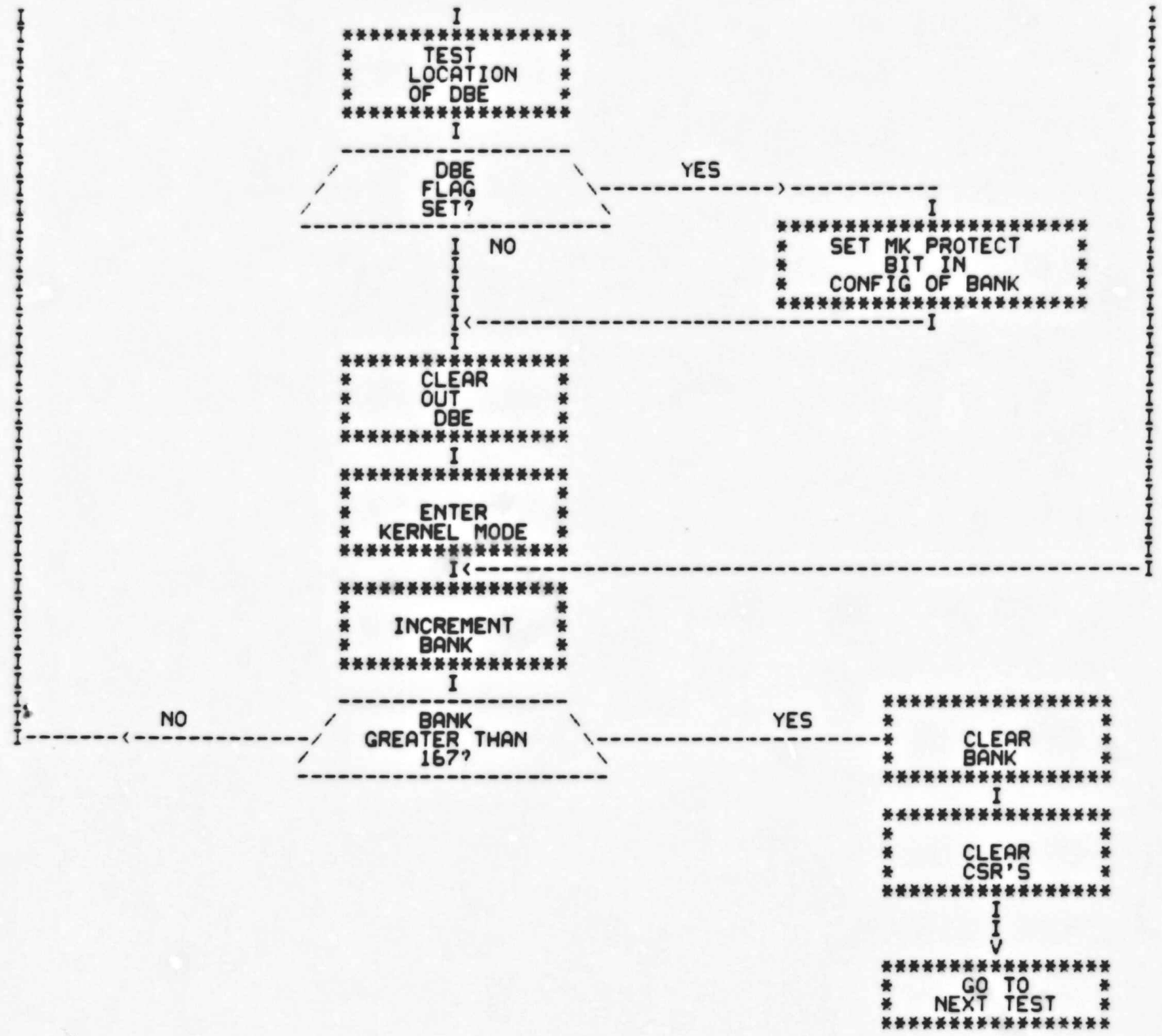
SEG 0088


SETUP INDEX TO CONFIG TABLE
GO TO A
ANY ERRORS?
YES
SET ERROR BIT FOR BANK IN CONFIG TABLE
SET ACCESSED BIT FOR BANK IN CONFIG TABLE
GO TO "A"

8.1.5 Determine Protected Banks Of MK11 -

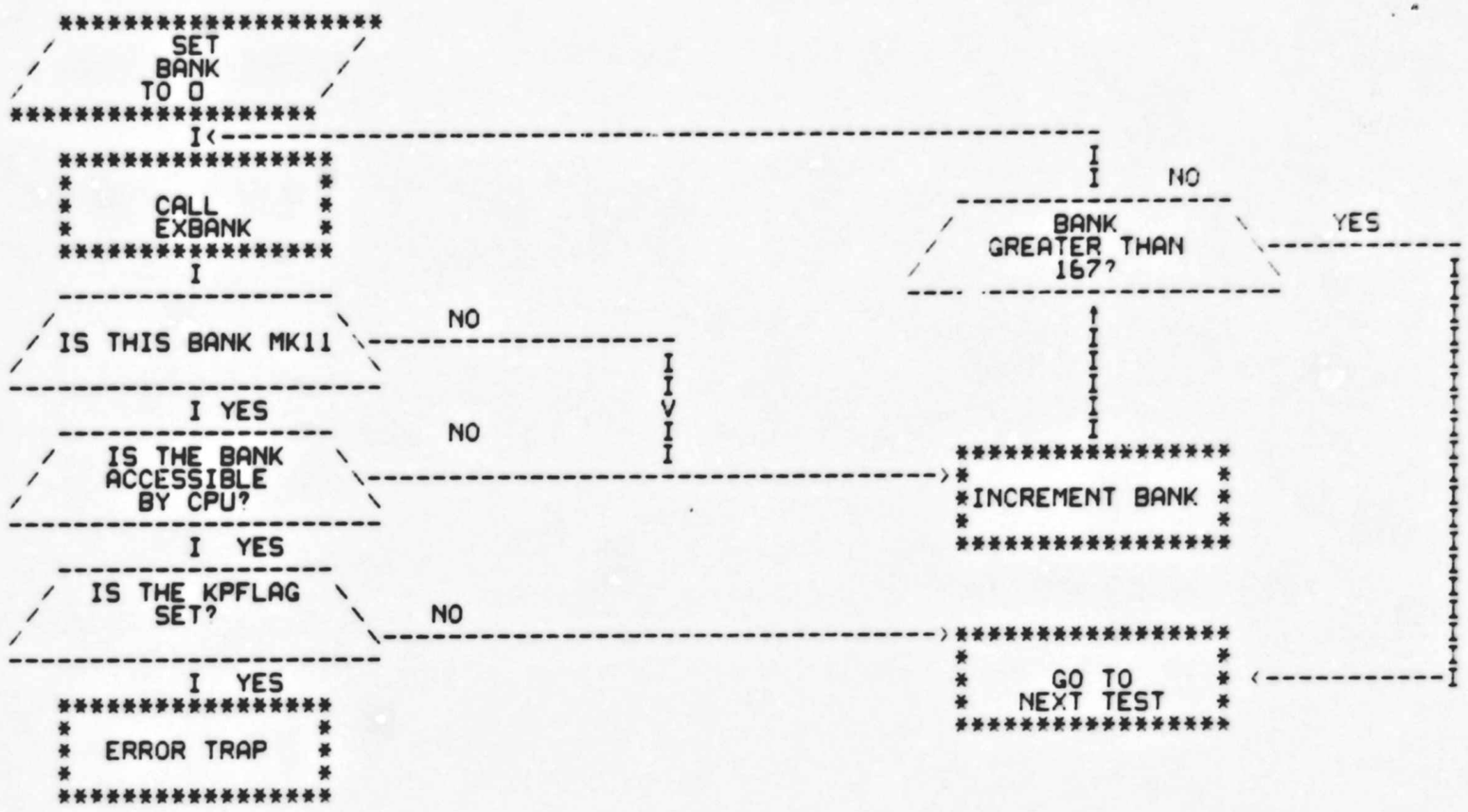


SEG 0090

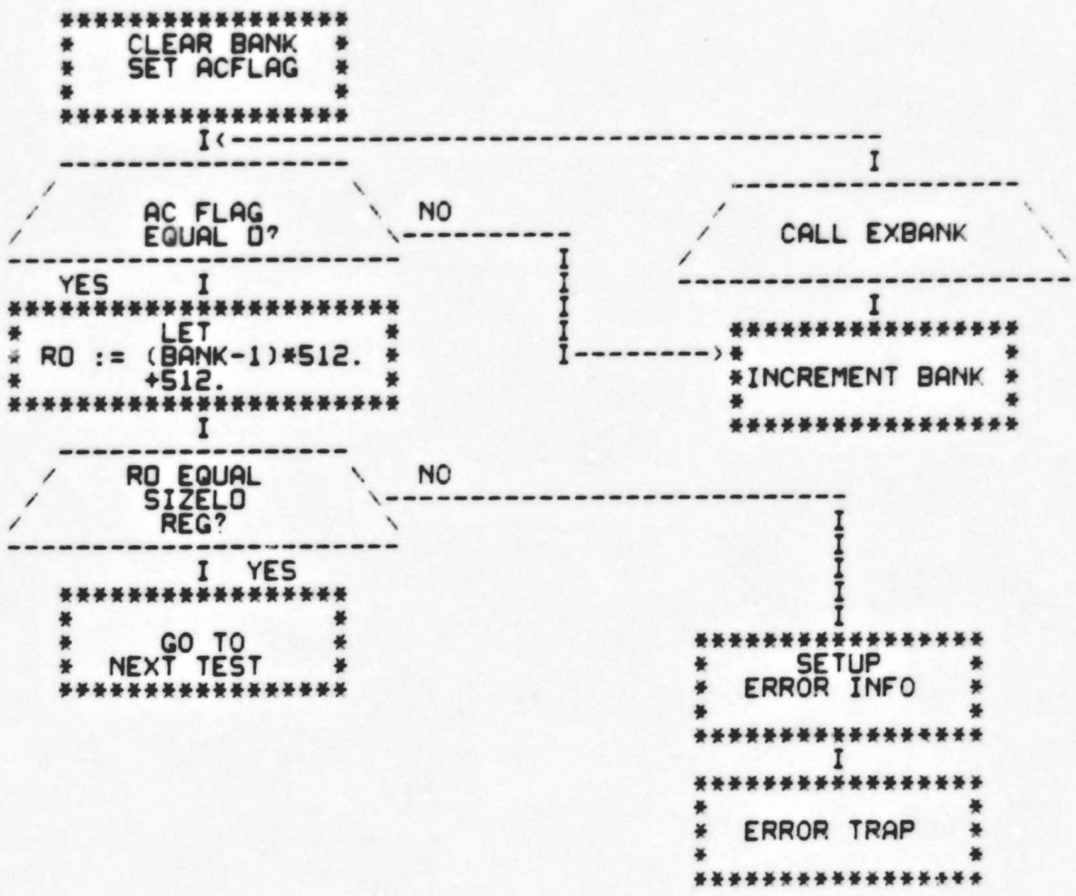

TEST LOCATION OF DBE
DBE FLAG SET?
YES
NO
SET MK PROTECT BIT IN CONFIG OF BANK
CLEAR OUT DBE
ENTER KERNEL MODE
INCREMENT BANK
BANK GREATER THAN 16?
NO
YES
CLEAR BANK
CLEAR CSR'S
GO TO NEXT TEST

SEQ 0091

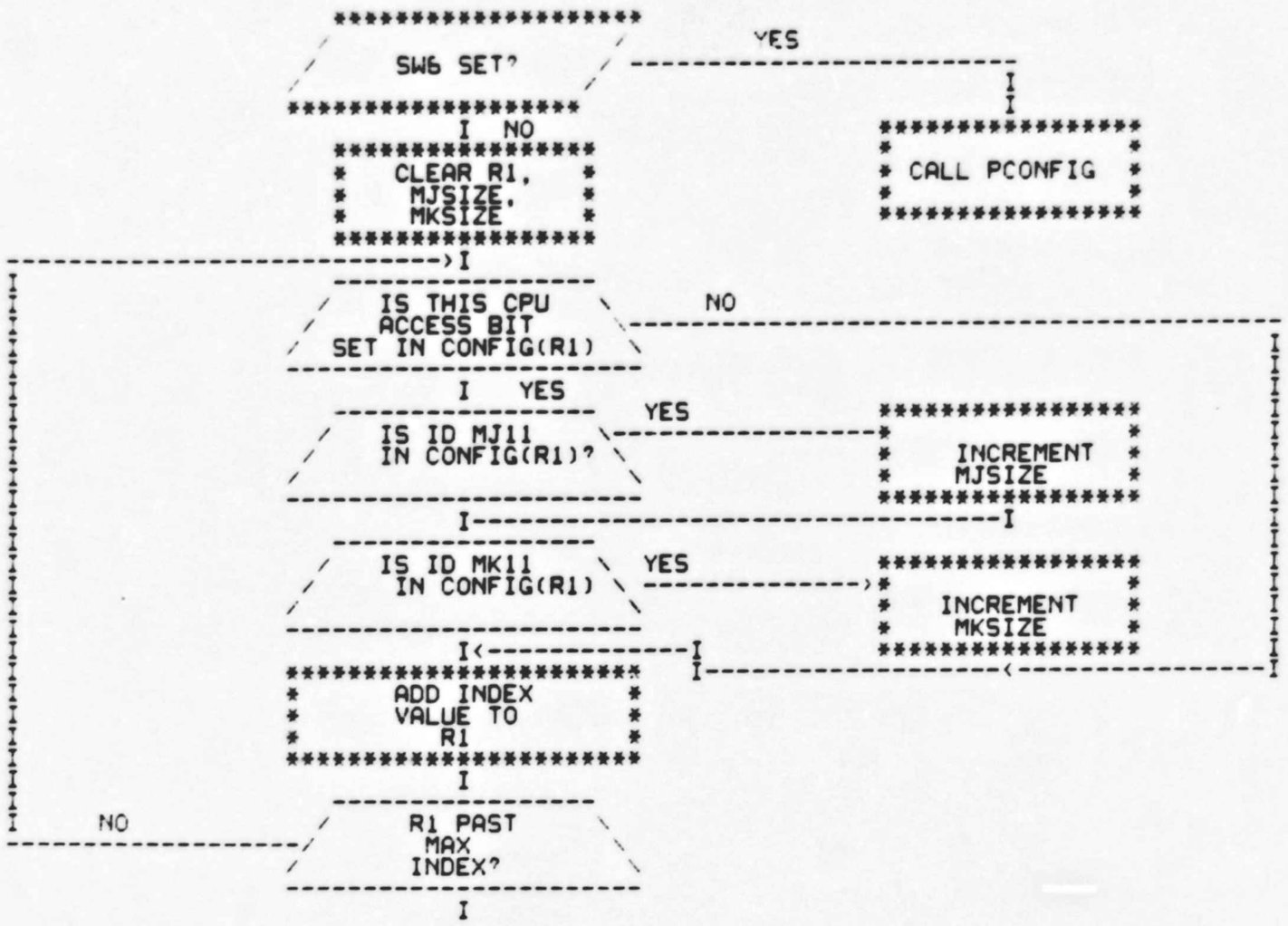
8.1.6 Test MK11 Protection Pointer -



8.1.7 Test System Size Register -

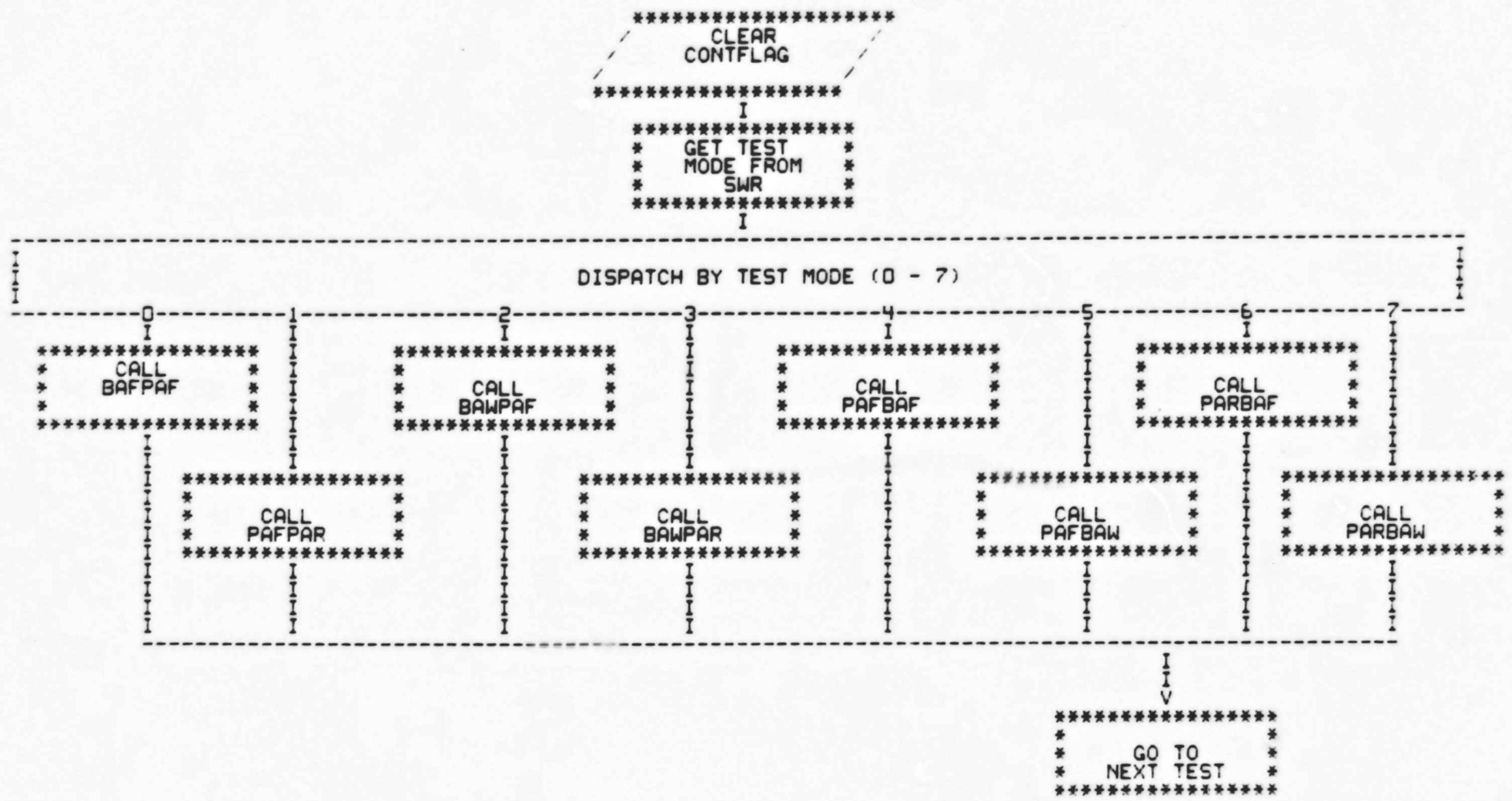


8.1.8 Type Configuration -



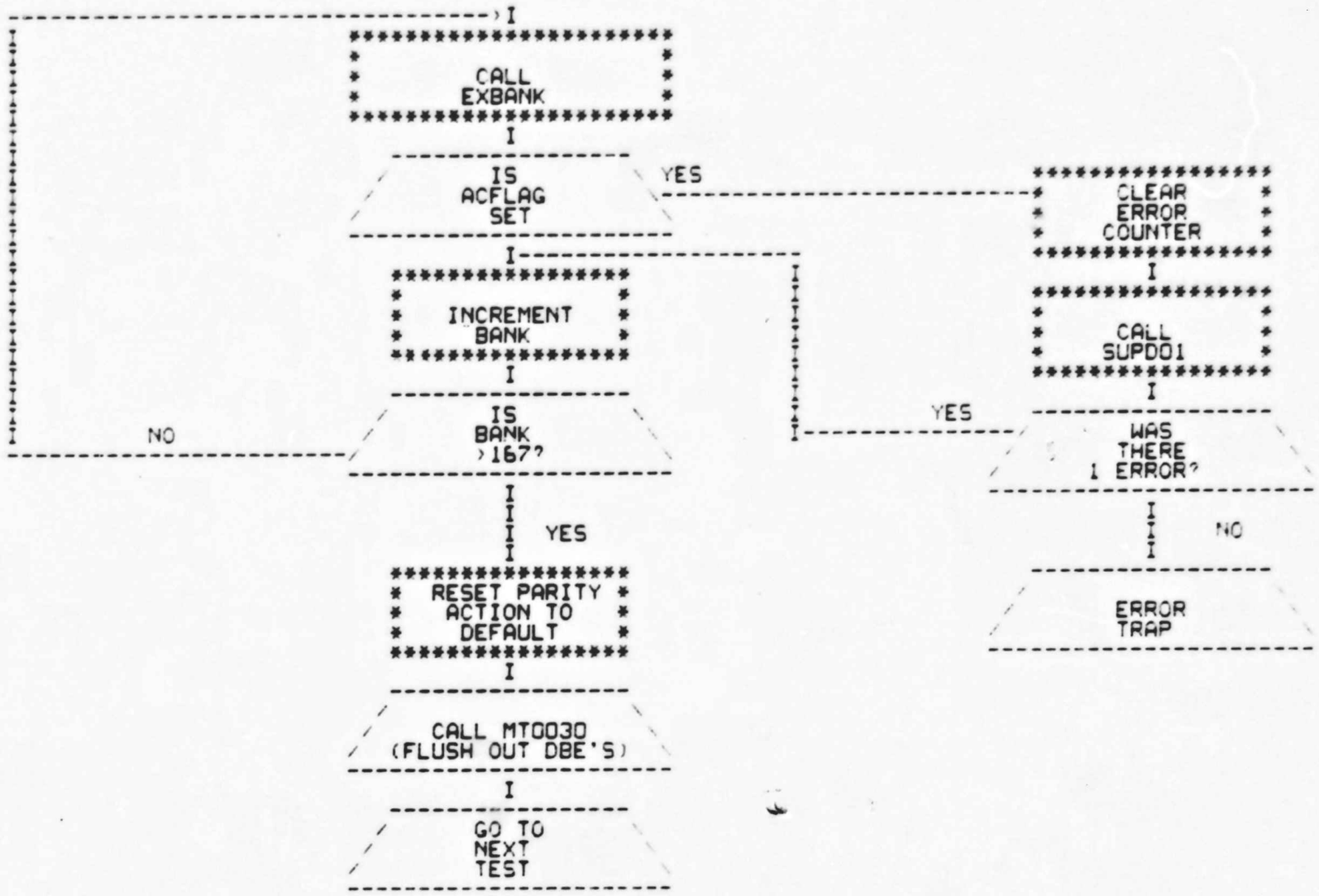
MKSIZE:=MKSIZE * 16.
MJSIZE:=MJSIZE * 16.
I:=MJSIZE + MKSIZE
PRINT I; "K OF MEMORY"
PRINT MJSIZE;"K OF MJ11"
PRINT MKSIZE;"K OF MK11"
GO TO NEXT TEST

8.1.9 Run Bank Level Tests -



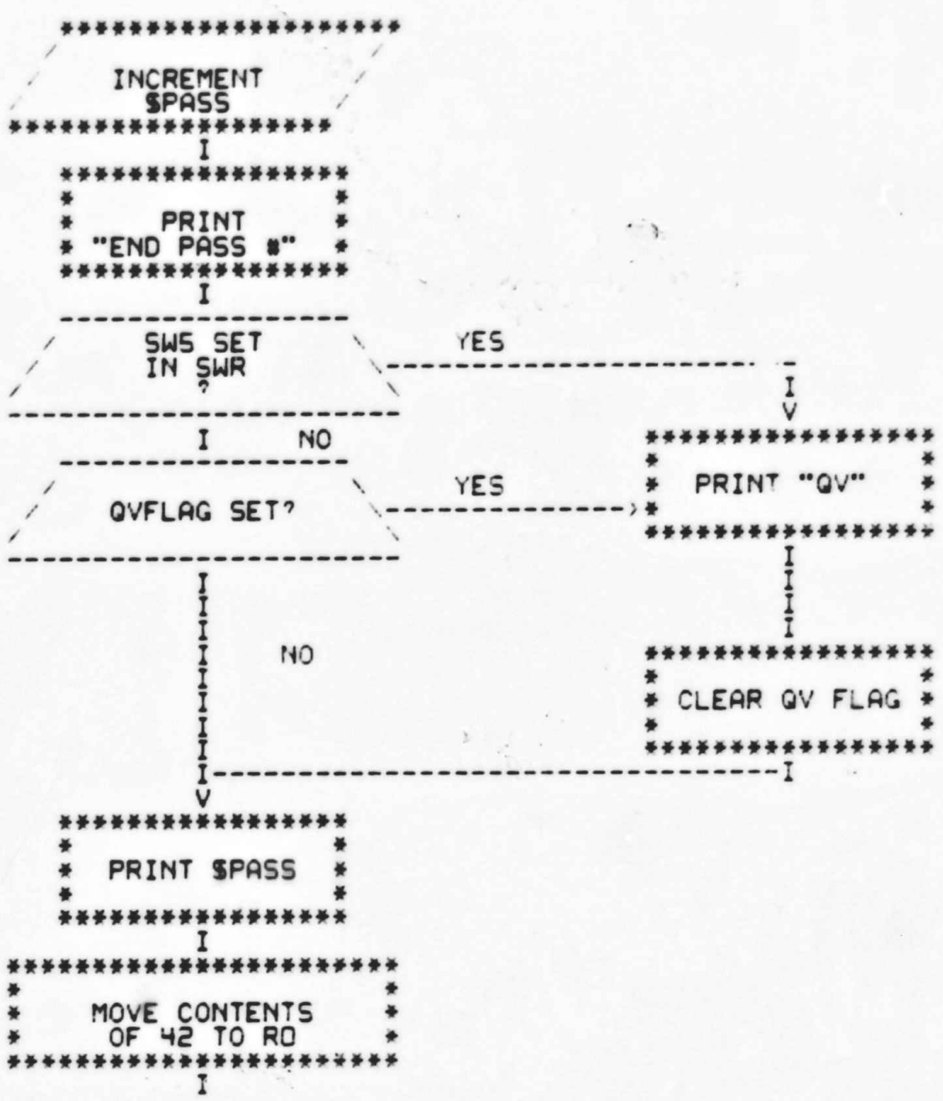
SEG 0096

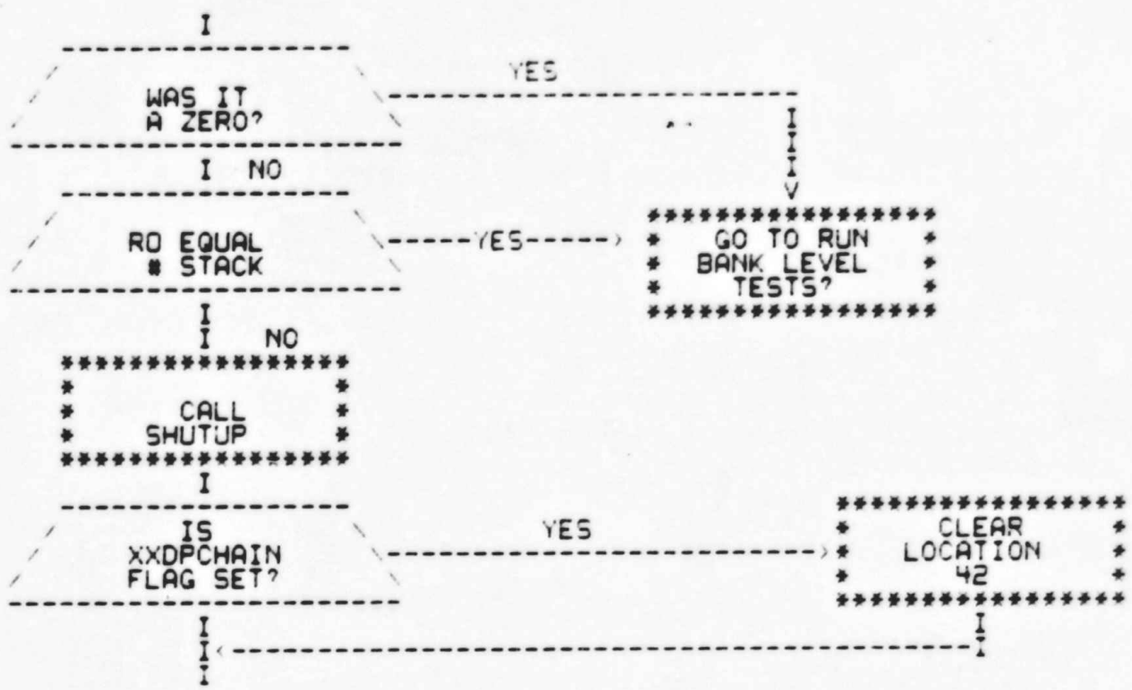
8.1.1C Run System Level Tests -

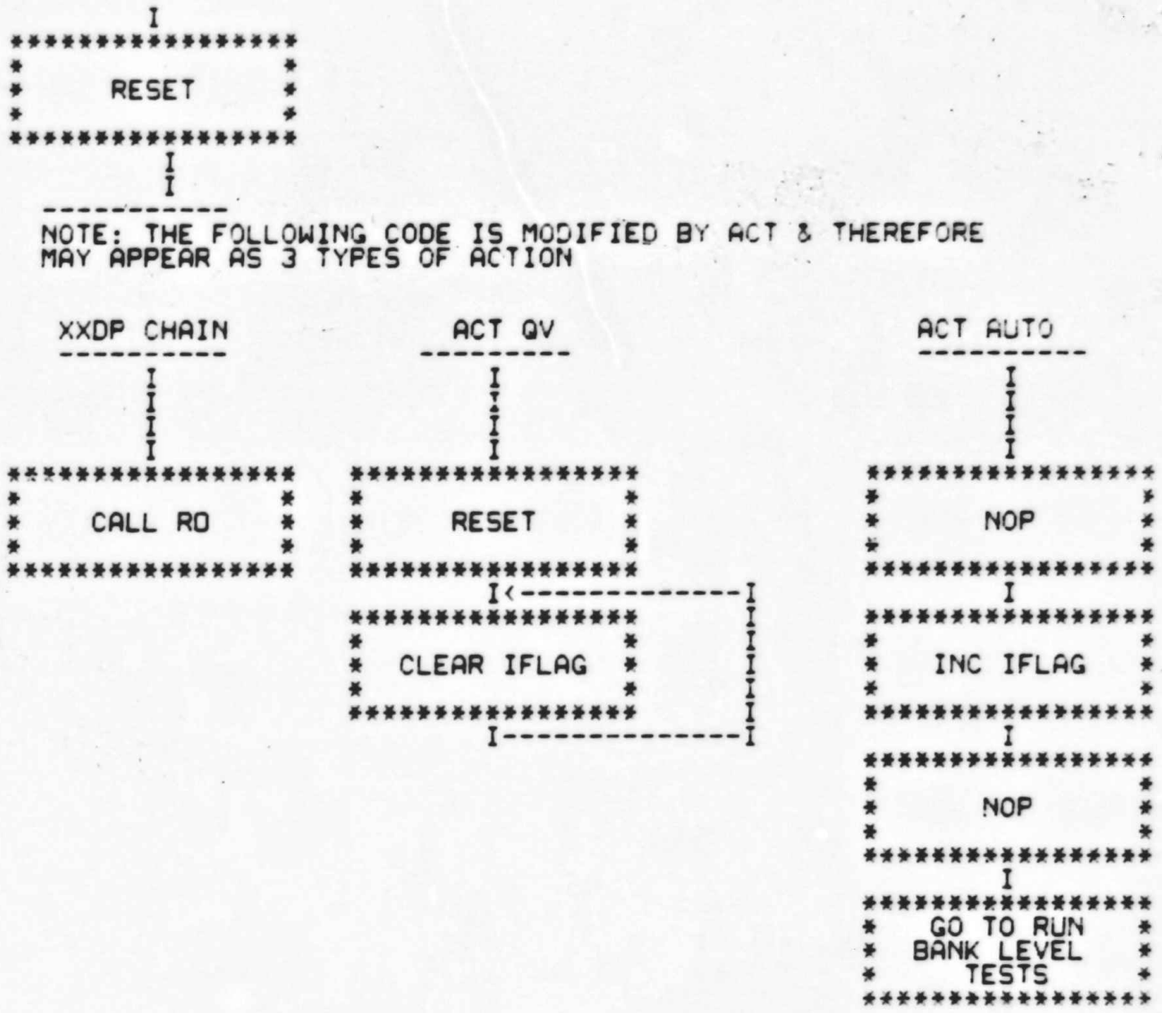

CALL
EXBANK
IS
ACFLAG
SET
YES
CLEAR
ERROR
COUNTER
INCREMENT
BANK
CALL
SUPD01
IS
BANK
>167?
NO
YES
WAS
THERE
1 ERROR?
YES
NO
RESET PARITY
ACTION TO
DEFAULT
ERROR
TRAP
CALL MT0030
(FLUSH OUT DBE'S)
GO TO
NEXT
TEST

SEG 0098

8.1.11 End Of Pass Code -




WAS IT A ZERO?
YES
NO
R0 EQUAL # STACK
YES
GO TO RUN BANK LEVEL TESTS?
NO
CALL SHUTUP
IS XXDPCHAIN FLAG SET?
YES
CLEAR LOCATION 42


RESET
NOTE: THE FOLLOWING CODE IS MODIFIED BY ACT & THEREFORE MAY APPEAR AS 3 TYPES OF ACTION
XXDP CHAIN
CALL RO
ACT QV
RESET
CLEAR IFLAG
ACT AUTO
NOP
INC IFLAG
NOP
GO TO RUN BANK LEVEL TESTS

```
     34            OPERATIONAL SWITCH SETTINGS
     35    ;SWITCH REGISTER DEFINITIONS
     36    ;*
     37    ;*      SWITCH                  USE
     38    ;*      ------          --------------------
     39    ;*        15            HALT ON ERROR
     40    ;*        14            LOOP ON TEST
     41    ;*        13            INHIBIT ERROR TYPEOUTS
     42    ;*        12            INHIBIT RELOCATION
     43    ;*        11            QUICK VERIFY / INHIBIT MARGIN TESTING
     44    ;*        10            BELL ON ERROR
     45    ;*         9            LOOP ON ERROR
     46    ;*         8            HALT PROGRAM (UNRELOCATED & RESTORE LOADERS)
     47    ;*         7            DETAILED ERROR REPORTS
     48    ;*         6            PRINT CONFIGURATION MAP
     49    ;*         5            LIMIT MAX ERRORS PER BANK
     50    ;*         4            FAT TERMINAL (132 COLUMNS OR BETTER)
     51    ;*         3            TEST MODE - SEE DOCUMENT
     52    ;*         2            TEST MODE - SEE DOCUMENT
     53    ;*         1            TEST MODE - SEE DOCUMENT
     54    ;*         0            DETECT SINGLE BIT ERRORS
     75            DEFINE  TRAPS
    162            DEFINE  BASIC PDP11 STUFF
    268            DEFINE  CACHE REGISTERS
    276            DEFINE  CPU REGISTERS
    284            DEFINE  MEMORY MANAGEMENT REGISTERS
    413            DEFINE  UNIBUS MAP REGISTERS
    481            DEFINE  SOFTWARE SWITCH & DISPLAY REGISTERS
    485            DEFINE  MK11 REGISTERS
    488            DEFINE  PARAMETERS
    494            MACRO   FATAL
    510            MACRO   TYPE
    523            MACRO   NEWTST
    572            MACRO   $$NEWTEST
    593            MACRO   SUBTST
    612            MACRO   $SUBTST
    626            MACRO   TYPOCT
    661            MACRO   TYPOCS
    712            MACRO   TYPDEC
    750            MACRO   BMOV
    803            MACRO   MAP
    837            MACRO   SUPERVISOR
    854            MACRO   USER
    871            MACRO   SET
    889            MACRO   SET4 & RES4
    917            MACRO   DLEFT
    946            ACT11 HOOKS
    965            APT11 HOOKS
    981            VARIABLES       INITIALIZED TO ZERO
   1146            VARIABLES       INITIALIZED TO NON ZERO
   1189            TABLE   CONFIGURATION
   1246            INITIALIZE VARIABLES TO ZERO
   1262            CLEAR NON-PROGRAM SPACE
   1271            INITIALIZE VARIABLES TO NON ZERO
   1283            INITIALIZE VECTORS
   1300            INITIALIZE PATTERNS
```

1318 SUBR PLUG IN NULL PATTERNS
1327 CLEAR THE CONFIGURATION TABLE
1337 SIZE FOR A HARDWARE SWITCH REGISTER
1351 SETUP ACT, APT, & XXDP
1378 BEFORE FIRST LOOP CODE
1379 PROTECT PROGRAM & LOADERS
1386 DETERMINE MK11 ACCESS METHOD
1398 SETUP MAP & USER & SUPERVISOR STACK
1411 GET SOFTWARE SWITCH REGISTER IF NECESSARY
1420 SETUP UNIBUS MAP
1429 GET MEMORY MANAGEMENT READY
1434 T1 ANALIZE CONFIGURATION
1522 T2 TEST BANK 0 ACCESSES
1545 ENABLE MK11 FOR CORRECT TRAPS
1552 T3 TEST BANKS 1-167 (OCTAL) FOR ZEROS & ONES
1645 DETERMINE PROTECTED BANKS
1717 MOVE LOADERS IF NECESSARY
1747 T4 MK11 PROTECTION POINTER TEST
1761 CHECK SYSTEM SIZE REGISTER
1826 MULTIPORT HOOK
1881 CHECK FOR UNIQUE SERIAL NUMBERS
1901 PRINT MULTIPORT DIRECTORY
1914 LOAD CSR'S FROM E-TABLE $CDW1 & $CDW2
1929 PRINT MEMORY CONFIGURATION MAP
1964 CHECK APT SIZING
1999 T5 DIAGNOSTIC MODE DISPATCH ROUTINE
2017 T6 UNIQUE BANK TEST
2024 T7 FORCE ADDRESS PARITY ERRORS
2056 FLUSH OUT DBE'S
2060 END OF PASS ROUTINE
2116 BANKS FORWARD,PATTERNS FORWARD **RECURSIVE**
2149 BANKS FORWARD,PATTERNS REVERSE **RECURSIVE**
2182 BANKS WORST FIRST,PATTERNS FORWARD **RECURSIVE**
2222 BANKS WORST FIRST,PATTERNS REVERSE **RECURSIVE**
2263 PATTERNS FORWARD,BANKS FORWARD **RECURSIVE**
2304 PATTERNS FORWARD,BANKS WORST FIRST **RECURSIVE**
2352 PATTERNS REVERSE,BANKS FORWARD **RECURSIVE**
2393 PATTERNS REVERSE,BANKS WORST FIRST **RECURSIVE**
2442 SUBR SETUP MEMORY TEST
2462 SUBR TEST MK11 CONTROLLER LOGIC DISPATCH
2516 IS ECC ON?
2537 CHECK FOR SBE FREE LOCATIONS
2594 SETUP INTERLEAVE INFO
2625 SUBR GET SIZE FROM CSR
2654 CSR PATTERN CASE STATEMENT
2695 DETERMINE ADDRESSING ACCORDING TO INTERLEAVE
2828 DETERMINE IF BANK BELONGS TO CORRECT SIDE
2856 SUBR MK11 TEST DISPATCH
2908 SUBR MJ11 TEST DISPATCH
2956 MEMORY TEST SETUP ROUTINES
2957 MT0000 SETUP DATA PATTERN TEST
2965 MT0001 SETUP ADDRESS TEST
2974 MT0002 SETUP COMPLEMENT ADDRESS TEST
2984 MT0003 SETUP 3 XOR 9 WORST CASE NOISE TEST
3009 MT0004 SETUP ROTATING ZEROS TEST
3020 MT0005 SETUP ROTATING ONES TEST

3031 MT0006 SETUP INITIAL DATA TEST
3038 MT0007 SETUP ADDRESS BIT TEST
3049 MT0010 SETUP BYTE ADDRESSING TEST
3056 MT0011 SETUP CREATE SINGLE BIT ERROR TEST
3065 MT0012 SETUP WRITE BYTE CLEARS SBE TEST
3074 MT0013 SETUP CREATE DOUBLE BIT ERROR TEST
3083 MT0014 SETUP WRITE INHIBIT DURING DATIP WITH DBE
3093 MT0015 SETUP WRITE INHIBIT OF BYTE WITH DBE
3102 MT0016 SETUP WRITE INHIBIT OF WORD WITH DBE
3111 MT0017 SETUP HOLDING 1'S & 0'S
3116 MT0020 SETUP MARCHING 0'S & 1'S IN CHECKBITS TEST
3187 MT0021 SETUP MARCHING 0'S & 1'S TEST
3212 MT0022 SETUP REFRESH TEST
3217 MT0023 SETUP SHIFTING DIAGONAL PATTERN TEST
3224 MT0024 SETUP FAST GALLOPING PATTERN TEST
3232 MT0025 SETUP INTERRUPT ENABLE TEST
3240 MT0026 SETUP RANDOM DATA TEST
3262 MT0027 UNIQUE BANK TEST
3311 MT0030 SETUP FLUSH OUT DBE'S TEST
3350 MT0031 SETUP SOB-A-LONG TEST
3364 MT0999 SETUP NULL TEST
3370 SUBR   EXECUTE PATTERN IN SUPERVISOR
3437 MEMORY TEST PATTERN ROUTINES
3447 MTP000 BASIC DATA TEST
3458 MTP001 ADDRESS TEST
3470 MTP002 COMPLEMENT ADDRESS TEST (WRITE DOWN, READ UP)
3484 MTPA03 3 XOR 9 WORST CASE NOISE TEST (WRITE)
3509 MTPB03 3 XOR 9 WORST CASE NOISE TEST (READ)
3526 MTPC03 TEST DATA SUBPROGRAM
3535 MTPD03 TEST DATA SUBSUBPROGRAM
3544 MTPA04 ROTATING ZEROS TEST
3557 MTPB04 SUBR   ROTATING BIT
3566 MTP005 ROTATION ONES TEST
3580 MTP006 INITIAL DATA TEST
3623 MTP007 ADDRESS BIT TEST
3662 MTP010 BYTE ADDRESSING TEST
3698 MPT011 SINGLE BIT ERROR TEST
3808 MTP012 WRITE BYTE CLEARS SBE TEST
3886 MTP013 CREATE DOUBLE BIT ERROR TEST
3953 MTP014 WRITE INHIBIT DURING DATIP WITH DBE TEST
4035 MTP015 WRITE INHIBIT OF BYTE WITH DBE
4110 MTP016 WRITE INHIBIT OF WORD WITH DBE
4188 MTP017 HOLDING 1'S & 0'S TEST
4221 MTP020 MARCHING 1'S & 0'S IN CHECK BITS TEST
4297 MTPA20 SLOW MARCHING 1'S & 0'S IN CHECK BITS TEST
4429 MTPA21 MARCHING 1'S & 0'S PATTERN TEST
4498 MTP022 REFRESH TEST
4566 SUBR   REFRESH DELAY
4589 MTP023 SHIFTING DIAGONAL PATTERN TEST
4656 MTP024 FAST GALLOPING PATTERN TEST
4734 MTP025 INTERRUPT ENABLE TEST
4823 MTPA26 RANDOM DATA (WRITE)
4830 MTPB26 RANDOM DATA (READ)
4848 RANDOM NUMBER SUBPROGRAM
4861 RANDOM NUMBER SUBSUBPROGRAM
4868 MT0030 FLUSH OUT DBE'S

4874 MTP031 SOB-A-LONG TEST
4901 SUBR COPY R0 TO R4,R1 TO R3, & R2 TO R5
4907 FLIP WARNING CONSTANTS IN WORST CASE NOISE TESTS
4935 SUBR WRITE BACKGROUND
4947 SUBR PRINT CONFIGURATION MAP
4997 SUBR TYPE CONFIGURATION
5151 TRAP PARITY ERROR HANDLER
5196 TRAP NON-EXISTANT MEMORY (HOLES) HANDLER
5216 TRAP TIMEOUT (TRAP TO 4) HANDLER
5220 TRAP MEMORY MANAGEMENT (TRAP TO 250) HANDLER
5223 TRAP RESERVED INSTRUCTION HANDLER
5231 TRAP KERNEL TRAP HANDLER
5239 TRAP ENERGIZE TRAP HANDLER
5243 TRAP DEENERGIZE TRAP HANDLER
5247 TRAP CACHON TRAP HANDLER
5252 TRAP CACHOFF TRAP HANDLER
5258 TRAP LOAD CSR TRAP HANDLER
5309 TRAP READ CSR TRAP HANDLER
5328 TRAP TEST (R1) & READ CSR CAREFULLY
5360 TRAP ECC DISABLE ALL CSR'S TRAP HANDLER
5366 TRAP ECC DISABLE OF 1 SELECTED CSR TRAP HANDLER
5372 TRAP MK11 INITIALIZE ALL CSR'S TRAP HANDLER
5377 TRAP MK11 INITIALIZE 1 SELECTED CSR TRAP HANDLER
5382 TRAP ENABLE SBE PARITY TRAPS ON ALL CSR'S
5388 TRAP ENABLE SBE PARITY TRAPS ON 1 SELECTED CSR
5394 TRAP WRITE CHECKBITS THRU ALL CSR'S TRAP HANDLER
5401 TRAP WRITE CHECKBITS THRU 1 SELECTED CSR TRAP HANDLER
5410 TRAP WAS THERE A SBE ON ANY CSR TRAP HANDLER
5435 TRAP WAS THERE A SBE IN 1 SELECTED CSR TRAP HANDLER
5443 TRAP WAS THERE A DBE ON ANY CSR TRAP HANDLER
5468 TRAP WAS THERE A DBE ON 1 SELECTED CSR TRAP HANDLER
5484 TRAP CLEAR ALL MK11 CSR'S TRAP HANDLER
5489 TRAP CLEAR 1 SELECTED MK11 CSR TRAP HANDLER
5494 TRAP ECC DISABLE, CHECH MODE, & WRITE CHECKBITS IN ALL CSR'S TRAP HANDLER
5501 TRAP ECC DISABLE, CHECH MODE, & WRITE CHECKBITS IN 1 SELECTED CSR TRAP HANDLER
5509 SUBR WRITE IN ALL CSR'S
5525 SUBR GENERATE CHECK BITS
5590 SUBR MAPPER
5622 TRAP MAP KERNEL (ALMOST 1 TO 1) TRAP HANDLER
5639 RELOCATE PROGRAM
5699 UNRELOCATE PROGRAM
5729 SETUP LOWER 16K OF UNIBUS MAP
5741 MOVE BANKS
5787 SUBR MAP USER TO NEW BANK
5807 SUBR SETUP KERNEL PAR'S FOR NEW BANK
5819 SUBR EXAMINE BANK
5882 SUBR BANK OK?
5893 SUBR INCREMENT PATTERN TESTING MK11'S
5901 SUBR SET HIGHEST PATTERN TESTING TYPE
5905 SUBR INCREMENT BANK & TEST
5911 SUBR INCREMENT MARGINS & TEST
5939 DECREMENT MARGINS & TEST
5951 SUBR SET HIGHEST MARGIN
5969 SUBR REWRITE CORE HISTORY
5984 BOOTSTRAP ROUTINE
6012 HALT PROGRAM

6021 SHUTDOWN DIAGNOSTIC
6042 APT SHUTDOWN SEQUENCE
6051 BLOCK MOVE SUBROUTINE
6080 SUBR FIELD SERVICE COMMAND MODE
6120 COMMAND 0 EXIT
6126 FS COMMAND 1 READ CSR
6140 FS COMMAND 2 LOAD CSR
6168 FS COMMAND 3 EXAMINE MEMORY
6206 FS COMMAND 4 MODIFY MEMORY
6255 FS COMMAND 5 SELECT BANK,MARGIN, & PATTERN
6373 FS COMMAND 6 TYPE CONFIGURATION MAP
6379 FS COMMAND 7 BATTERY BACKUP TEST
6424 FS COMMAND 8 SOB-A-LONG TEST
6477 FS COMMAND 9 SUPER TIGHT SCOPE LOOP
6544 FS COMMAND 10 ERROR SUMMARY
6566 FS COMMAND 11 REFRESH TEST
6620 FS COMMAND 12 SET FILL COUNT
6630 SUBR DETERMINE CORRECT CSR
6655 MULTIPORT SLAVE STARTER
6715 INITIALIZE THE LIST
6735 MULTIPORT ACCESS PATHS TEST
6781 MULTIPORT CONTENTION BALANCE TEST
6816 MULTIPORT RUNNING CONTENTION TEST
6852 STOP SLAVES
6907 MOVE PROGRAM TO NEW BANK
6916 WRITE ONLY PATTERN IN HI BYTE
6932 READ ONLY PATTERN IN HI BYTE
6959 SLAVE PATTERNS
6978 RUN RANDOM INDIRECT INSTRUCTIONS TEST
6996 CONTENTION BALANCE TEST
7034 STOP SLAVE PROGRAM
7039 SETUP RANDOM INSTRUCTION TEST
7064 SUBR ERROR DATA SETUP
7318 DATA WAS A WORD
7333 DATA WAS A BYTE
7348 DATA WAS A 7 BIT BYTE
7362 DATA WAS 3 WORDS
7383 DETERMINE XOR OF GOOD & BAD
7400 LOG ERROR ON BAD BANK
7458 GET DATA FROM ABORTED AREA IF POSSIBLE
7475 ROUTINE SCOPE HANDLER
7524 SUBR DISPLAY
7546 ROUTINE ERROR HANDLER
7619 ROUTINE ERROR MESSAGE TYPEOUT
7850 SUBR WHO'S BOX IS THIS?
7912 SUBR DETAILED ERROR REPORT
7985 ROUTINE POWER DOWN AND UP
8127 POWER FAIL WHILE RELOCATED
8151 POWER UP FROM BANK 0 TO RELOCATION
8184 055272 002506 .PRINT 60000-KERLIMIT ;SUPERVISOR ADDRESSES LEFT
8191 ROUTINE TYPE
8251 ROUTINE BINARY TO OCTAL (ASCII) AND TYPE
8329 ROUTINE CONVERT BINARY TO DECIMAL AND TYPE
8386 ROUTINE TTY INPUT
8468 CONTROL T
8586 ROUTINE READ AN OCTAL NUMBER FROM THE TTY

  8635   ROUTINE READ A DECIMAL NUMBER FROM THE TTY
  8694   ROUTINE SAVE AND RESTORE R0-R5
  8730   ROUTINE RANDOM NUMBER GENERATOR
  8759   ROUTINE DOUBLE LENGTH BINARY TO OCTAL ASCII CONVERT
  8801   APT MAILBOX-ETABLE
  8880   ROUTINE TRAP DECODER
  8903   TRAP TABLE
  9005   TABLE   ERROR POINTER
  9235   ERROR MESSAGES (EM)
  9283   ERROR DATA HEADERS (DH)
  9311   ERROR DATA TAGS (DT)
  9336   ERROR DATA FORMATS (DF)
  9350   MESSAGES
 10303 076350 001430   .PRINT 100000-END   ;ADDRESSES LEFT

```
                        .TITLE  CEMKAAD 1170 MAIN MEMORY DIAG
```

IDENTIFICATION

PRODUCT CODE: AC-C644A-MC

PRODUCT NAME: CEMKAAD 1170 MAIN MEMORY DIAG

PRODUCT DATE: 24 FEBRUARY 1978

MAINTAINER: PRODUCT ENHANCEMENT

THE INFORMATION IN THIS DOCUMENT IS SUBJECT TO CHANGE WITHOUT NOTICE AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT CORPORATION. DIGITAL EQUIPMENT CORPORATION ASSUMES NO RESPONSIBILITY FOR ANY ERRORS THAT MAY APPEAR IN THIS MANUAL.

THE SOFTWARE DESCRIBED IN THIS DOCUMENT IS FURNISHED TO THE PURCHASER UNDER A LICENSE FOR USE ON A SINGLE COMPUTER SYSTEM AND CAN BE COPIED (WITH INCLUSION OF DIGITALS COPYRIGHT NOTICE) ONLY FOR USE IN SUCH SYSTEM, EXCEPT AS MAY OTHERWISE BE PROVIDED IN WRITING BY DIGITAL.

DIGITAL EQUIPMENT CORPORATION ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS SOFTWARE ON EQUIPMENT THAT IS NOT SUPPLIED BY DIGITAL.

COPYRIGHT (C) 1978 DIGITAL EQUIPMENT CORPORATION

E09


```
                        .SBTTL  OPERATIONAL SWITCH SETTINGS
                        .SBTTL ;SWITCH REGISTER DEFINITIONS
                        .SBTTL ;*
                        .SBTTL ;*       SWITCH                  USE
                        .SBTTL ;*       ------          --------------------
                        .SBTTL ;*         15            HALT ON ERROR
                        .SBTTL ;*         14            LOOP ON TEST
                        .SBTTL ;*         13            INHIBIT ERROR TYPEOUTS
                        .SBTTL ;*         12            INHIBIT RELOCATION
                        .SBTTL ;*         11            QUICK VERIFY / INHIBIT MARGIN TESTING
                        .SBTTL ;*         10            BELL ON ERROR
                        .SBTTL ;*          9            LOOP ON ERROR
                        .SBTTL ;*          8            HALT PROGRAM (UNRELOCATED & RESTORE LOADERS)
                        .SBTTL ;*          7            DETAILED ERROR REPORTS
                        .SBTTL ;*          6            PRINT CONFIGURATION MAP
                        .SBTTL ;*          5            LIMIT MAX ERRORS PER BANK
                        .SBTTL ;*          4            FAT TERMINAL (132 COLUMNS OR BETTER)
                        .SBTTL ;*          3            TEST MODE - SEE DOCUMENT
                        .SBTTL ;*          2            TEST MODE - SEE DOCUMENT
                        .SBTTL ;*          1            TEST MODE - SEE DOCUMENT
                        .SBTTL ;*          0            DETECT SINGLE BIT ERRORS
```

F09


```
                                .ENABL  ABS
                                .ENABL  AMA
                                .MCALL  SMACIT,..PUSH,..POP,..TAG,..BRAN,.EMIT,.EMITL,.EMITR
                                .MCALL  .IFOPR,.IS,.GENBR,.OPADD,.OPSUB
                                .MCALL  RNE,REQ,RLT,RGE,RGT,RLE,RPL,RMI,RHI,RLOS,RHIS,RLO,RCS,RCC
                                .MCALL  IF,.OR,.IFARI,.LEAVE,.GOTO,OR,AND,DO,THEN,ELSE,WHILE,CASE
                                .MCALL  FOR,TO,DOWNTO,REPEAT,UNTIL,THRU,END,BEGIN
                                .MCALL  $$END,LEAVE,JUMPTO,GOTO,PUSH,POP,LET
                                .MCALL  .SIMPLE,.ARITH,ORB,ANDB,IFB,UNTILB,WHILEB,ON.ERROR,ON.NOERROR
                                .MCALL  FI,PROCEDURE,ENDPROCEDURE,FORMAT,PRTIT,TYPIT,$CALL,$RETURN

                                .NLIST  TTM                     ;I WANT FAT PAPER!
                                .LIST   MC,SYM                  ;LIST MACRO CALLS, SYMBOL TABLE
                                .NLIST  MD,CND,ME               ;DON'T LIST MACRO DEFINITIONS & CONDITIONALS & E
                        000000  LST$$=  0                       ;DEFINED TO LIST SUPERMAC EXPANSIONS
                        163000  $SWR=   163000                  ;USE THESE SYSMAC SWITCHES
                        000001  $TN=    1                       ;FIRST TEST NUMBER TO ONE(1)
000000                          SMACIT
```

```
                        .SBTTL  DEFINE  TRAPS
                        ;ALL ENTRIES HERE MUST HAVE A CORRESPONDING ENTRY IN THE
                        ;TRAP TABLE "$TRPAD" (NEAR END OF PROGRAM).
                        ;*TRAP DEFINITIONS
104401                  TYPEIT= 104401          ;;TTY TYPEOUT ROUTINE
104402                  TYPOC=  104402          ;;TYPE OCTAL NUMBER (WITH LEADING ZEROS)
104403                  TYPOS=  104403          ;;TYPE OCTAL NUMBER (NO LEADING ZEROS)
                        ;TYPON= 104404          ;;TYPE OCTAL NUMBER (AS PER LAST CALL)
104405                  TYPDS=  104405          ;;TYPE DECIMAL NUMBER (WITH SIGN)
                        ;TYPBN= 104406          ;;TYPE BINARY (ASCII) NUMBER

104407                  GTSWR=  104407          ;;GET SOFT-SWR SETTING
104410                  CKSWR=  104410          ;;TEST FOR CHANGE IN SOFT-SWR

104411                  RDCHR=  104411          ;;TTY TYPEIN CHARACTER ROUTINE
104412                  RDLIN=  104412          ;;TTY TYPEIN STRING ROUTINE
104413                  RDOCT=  104413          ;;READ AN OCTAL NUMBER FROM TTY
104414                  RDDEC=  104414          ;;READ A DECIMAL NUMBER FROM TTY

104415                  SAVREG= 104415          ;;SAVE R0-R5 ROUTINE
104416                  RESREG= 104416          ;;RESTORE R0-R5 ROUTINE

104417                  KERNEL= 104417          ;ENTER KERNEL MODE

104420                  ENERGIZE=104420         ;TURN ON MEMORY MANAGEMENT & TRAPS
104421                  DEENERGIZE=104421       ;TURN OFF MEMORY MANAGEMENT & TRAPS

104422                  KMAP=   104422          ;MAP KERNEL 1 TO 1

104423                  CACHON= 104423          ;TURN ON CACHE
104424                  CACHOFF=104424          ;TURN OFF CACHE

104425                  LOADCSR=104425          ;LOAD CORRECT CSR
104426                  READCSR=104426          ;READ CORRECT CSR

104427                  PERR01= 104427          ;PROGRAM DETECTED ERROR
104430                  PERR02= 104430          ;PROGRAM DETECTED ERROR
104431                  PERR03= 104431          ;PROGRAM DETECTED ERROR
104432                  PERR04= 104432          ;PROGRAM DETECTED ERROR
104433                  PERR07= 104433          ;PROGRAM DETECTED ERROR
104434                  PERR10= 104434          ;PROGRAM DETECTED ERROR
104435                  PERR11= 104435          ;PROGRAM DETECTED ERROR
104436                  PERR12= 104436          ;PROGRAM DETECTED ERROR
104437                  PERR13= 104437          ;PROGRAM DETECTED ERROR
104440                  PERR14= 104440          ;PROGRAM DETECTED ERROR
104441                  PERR15= 104441          ;PROGRAM DETECTED ERROR
104442                  PERR16= 104442          ;PROGRAM DETECTED ERROR
104443                  PERR17= 104443          ;PROGRAM DETECTED ERROR
104444                  PERR20= 104444          ;PROGRAM DETECTED ERROR
104445                  PERR21= 104445          ;PROGRAM DETECTED ERROR
104446                  PERR22= 104446          ;PROGRAM DETECTED ERROR
104447                  PERR23= 104447          ;PROGRAM DETECTED ERROR
104450                  PERR24= 104450          ;PROGRAM DETECTED ERROR
104451                  PERR25= 104451          ;PROGRAM DETECTED ERROR
104452                  PERR26= 104452          ;PROGRAM DETECTED ERROR
104453                  PERR27= 104453          ;PROGRAM DETECTED ERROR
```

```
104454          PERR30= 104454          ;PROGRAM DETECTED ERROR
104455          PERR31= 104455          ;PROGRAM DETECTED ERROR
104456          PERR32= 104456          ;PROGRAM DETECTED ERROR
104457          PERR33= 104457          ;PROGRAM DETECTED ERROR
104460          PERR34= 104460          ;PROGRAM DETECTED ERROR
104461          PERR35= 104461          ;PROGRAM DETECTED ERROR
104462          PERR36= 104462          ;PROGRAM DETECTED ERROR
104463          PERR37= 104463          ;PROGRAM DETECTED ERROR
104464          PERR40= 104464          ;PROGRAM DETECTED ERROR
104465          PERR41= 104465          ;PROGRAM DETECTED ERROR
104466          PERR42= 104466          ;PROGRAM DETECTED ERROR
104467          PERR43= 104467          ;PROGRAM DETECTED ERROR

104470          ECCDIS= 104470          ;DISABLE ECC ON ALL CSR'S
104471          ECC1DIS=104471          ;DISABLE ECC ON 1 SELECTED CSR
104472          ECCINIT=104472          ;INITIALIZE ALL MK11 CSR'S
104473          ECC1INIT=104473         ;INITIALIZE 1 SELECTED MK11 CSR
104474          CBCSR=  104474          ;WRITE GENERATED CHECKBITS IN ALL CSR'S
104475          CB1CSR= 104475          ;WRITE GENERATED CHECKBITS IN 1 SELECTED CSR
104476          WASSBE= 104476          ;WAS THERE A SBE ON ANY CSR?
104477          WAS1SBE=104477          ;WAS THERE A SBE ON 1 SELECTED CSR?
104500          WASDBE= 104500          ;WAS THERE A DBE ON ANY CSR?
104501          WAS1DBE=104501          ;WAS THERE A DBE ON 1 SELECTED CSR?
104502          CLRCSR= 104502          ;CLEAR ALL CSR'S
104503          CLR1CSR=104503          ;CLEAR 1 SELECTED CSR
104504          CHKDIS= 104504          ;DISABLE ECC & WRITE CHECKBITS FROM ALL CSR'S
104505          CHK1DIS=104505          ;DISABLE ECC & WRITE CHECKBITS FROM 1 SELECTED CSR
104506          ENASBE= 104506          ;ENABLE TRAPS ON SBE'S FROM ALL CSR'S
104507          ENA1SBE=104507          ;ENABLE TRAPS ON SBE'S FROM 1 SELECTED CSR
104510          TSTREAD=104510          ;TEST LOCATION (R1) & READ CSR (CAREFULLY)
```

```
                        .SBTTL  DEFINE  BASIC PDP11 STUFF

                        ;*INITIAL ADDRESS OF THE STACK POINTER
001400                  STACK=  1400            ;;FIRST ADDRESS OF THE STACK
001400                  KERSTK= STACK           ;;KERNEL STACK
000700                  SUPSTK= STACK-500       ;;SUPERVISOR STACK
000600                  USESTK= STACK-600       ;;USER STACK
104000                  ERROR=EMT               ;;BASIC DEFINITION OF ERROR CALL
000004                  SCOPE=IOT               ;;BASIC DEFINITION OF SCOPE CALL
177776                  PSW=    177776          ;;PROCESSOR STATUS WORD
                        ;STKLMT=        177774          ;;STACK LIMIT REGISTER
                        ;PIRQ=  177772          ;;PROGRAM INTERRUPT REQUEST REGISTER
177570                  DSWR=   177570          ;;HARDWARE SWITCH REGISTER
177570                  DDISP=  177570          ;;HARDWARE DISPLAY REGISTER
177546                  LKS=    177546          ;;LINE CLOCK (KW11-L) STATUS REGISTER

                        ;*MISCELLANEOUS DEFINITIONS
000011                  HT=     11              ;;CODE FOR HORIZONTAL TAB
000012                  LF=     12              ;;CODE LINE FEED
000015                  CR=     15              ;;CODE CARRIAGE RETURN
000200                  CRLF=   200             ;;CODE FOR CARRIAGE RETURN-LINE FEED

                        ;*GENERAL PURPOSE REGISTER DEFINITIONS
000000                  R0=     %0              ;;GENERAL REGISTER
000001                  R1=     %1              ;;GENERAL REGISTER
000002                  R2=     %2              ;;GENERAL REGISTER
000003                  R3=     %3              ;;GENERAL REGISTER
000004                  R4=     %4              ;;GENERAL REGISTER
000005                  R5=     %5              ;;GENERAL REGISTER
000006                  R6=     %6              ;;GENERAL REGISTER
000007                  R7=     %7              ;;GENERAL REGISTER
                        ;R10=R0                 ;;GENERAL REGISTER
                        ;R11=R1                 ;;GENERAL REGISTER
                        ;R12=R2                 ;;GENERAL REGISTER
                        ;R13=R3                 ;;GENERAL REGISTER
                        ;R14=R4                 ;;GENERAL REGISTER
                        ;R15=R5                 ;;GENERAL REGISTER
000006                  SP=R6                   ;;STACK POINTER
                        ;KSP=SP                 ;;KERNEL STACK POINTER
000006                  SSP=SP                  ;;SUPERVISOR STACK POINTER
000006                  USP=SP                  ;;USER STACK POINTER
000007                  PC=R7                   ;;PROGRAM COUNTER

                        ;*PRIORITY LEVEL DEFINITIONS
                        ;PR0=   0               ;;PRIORITY LEVEL 0
                        ;PR1=   40              ;;PRIORITY LEVEL 1
                        ;PR2=   100             ;;PRIORITY LEVEL 2
                        ;PR3=   140             ;;PRIORITY LEVEL 3
                        ;PR4=   200             ;;PRIORITY LEVEL 4
                        ;PR5=   240             ;;PRIORITY LEVEL 5
                        ;PR6=   300             ;;PRIORITY LEVEL 6
                        ;PR7=   340             ;;PRIORITY LEVEL 7

                        ;*"SWITCH REGISTER" SWITCH DEFINITIONS
100000                  SW15=   100000
040000                  SW14=   40000
```

```
                020000          SW13=   20000
                010000          SW12=   10000
                004000          SW11=   4000
                002000          SW10=   2000
                001000          SW9=    1000
                000400          SW8=    400
                000200          SW7=    200
                000100          SW6=    100
                000040          SW5=    40
                000020          SW4=    20
                000010          SW3=    10
                000004          SW2=    4
                000002          SW1=    2
                000001          SW0=    1

                                ;*DATA BIT DEFINITIONS (BIT00 TO BIT15)
                100000          BIT15=  100000
                040000          BIT14=  40000
                020000          BIT13=  20000
                010000          BIT12=  10000
                004000          BIT11=  4000
                002000          BIT10=  2000
                001000          BIT9=   1000
                000400          BIT8=   400
                000200          BIT7=   200
                000100          BIT6=   100
                000040          BIT5=   40
                000020          BIT4=   20
                000010          BIT3=   10
                000004          BIT2=   4
                000002          BIT1=   2
                000001          BIT0=   1

                                ;*BASIC "CPU" TRAP VECTOR ADDRESSES
                000004          ERRVEC= 4               ;;TIME OUT AND OTHER ERRORS
                000010          RESVEC= 10              ;;RESERVED AND ILLEGAL INSTRUCTIONS
                                ;TBITVEC=14             ;;"T" BIT
                                ;TRTVEC=        14              ;;TRACE TRAP
                                ;BPTVEC=        14              ;;BREAKPOINT TRAP (BPT)
                000020          IOTVEC= 20              ;;INPUT/OUTPUT TRAP (IOT) **SCOPE**
                000024          PWRVEC= 24              ;;POWER FAIL
                000030          EMTVEC= 30              ;;EMULATOR TRAP (EMT) **ERROR**
                000034          TRAPVEC=34              ;;"TRAP" TRAP
                000060          TKVEC=  60              ;;TTY KEYBOARD VECTOR
                                ;TPVEC= 64              ;;TTY PRINTER VECTOR
                                ;LKVEC= 100             ;;LINE CLOCK (KW11-L) VECTER
                000114          CACHVEC=114             ;;CACHE ERROR INTERRUPT VECTOR
                000114          PARVEC=CACHVEC
                                ;PIRQVEC=240            ;;PROGRAM INTERRUPT REQUEST VECTOR
                000250          MMVEC=  250             ;;MEMORY MANAGEMENT VECTOR
                                .SBTTL  DEFINE  CACHE REGISTERS
                177740          LOADRS = 177740         ;;LOWER 16 BITS OF ADDRESS THAT CAUSED ERROR
                177742          HIADRS = 177742         ;;UPPER SIX BITS OF ADDRESS THAT CAUSED ERROR
                177744          MEMERR = 177744         ;;CACHE ERROR REGISTER
                177746          CONTRL = 177746         ;;MEMORY CONTROL REGISTER
                177750          MAINT  = 177750         ;;MEMORY MAINTENENCE REGISTER
```

```
                        ;HITMIS = 177752                ;;HIT MISS REGISTER "1" IMPLIES HIT IN CACHE

                                .SBTTL  DEFINE  CPU REGISTERS
177760                  SIZELO = 177760         ;;MEMORY SIZE REGISTER NUMBER TO PUT INTO A PAR
                                                ;;TO GET TO THE LAST 32 WORDS OF MEMORY
                        ;SIZEHI = 177762                ;;HIGH SIZE REGISTER, RESERVED FOR FUTURE USE
                                                ;;CURRENTLY ALL ZERO
177764                  SYSTID = 177764         ;;SYSTEM ID REGISTER
177766                  CPUERR = 177766         ;;CPU ERROR REGISTER HOLDS CONDITION THAT CAUSED

                                .SBTTL  DEFINE  MEMORY MANAGEMENT REGISTERS
                                ;*MEMORY MANAGEMENT STATUS REGISTER ADDRESSES
177572                  MMR0=   177572
177574                  MMR1=   177574
177576                  MMR2=   177576
172516                  MMR3=   172516

                                ;*USER "I" PAGE DESCRIPTOR REGISTERS
177600                  UIPDR0= 177600
                        ;UIPDR1=         177602
                        ;UIPDR2=         177604
                        ;UIPDR3=         177606
                        ;UIPDR4=         177610
                        ;UIPDR5=         177612
                        ;UIPDR6=         177614
                        ;UIPDR7=         177616

                                ;*USER "D" PAGE DESCRIPTOR REGISTORS
                        ;UDPDR0=         177620
                        ;UDPDR1=         177622
                        ;UDPDR2=         177624
                        ;UDPDR3=         177626
                        ;UDPDR4=         177630
                        ;UDPDR5=         177632
                        ;UDPDR6=         177634
177636                  UDPDR7= 177636

                                ;*USER "I" PAGE ADDRESS REGISTERS
177640                  FASTCITY=UIPAR0
177640                  UIPAR0= 177640          ;PATTERN PROGRAM SPACE
                        ;UIPAR1=         177642         ;PATTERN PROGRAM SPACE
177644                  UIPAR2= 177644          ;PATTERN PROGRAM SPACE
177646                  UIPAR3= 177646          ;PATTERN PROGRAM SPACE
177650                  UIPAR4= 177650          ;PATTERN PROGRAM SPACE
                        ;UIPAR5=         177652         ;PATTERN PROGRAM SPACE
177654                  UIPAR6= 177654          ;PATTERN PROGRAM SPACE
177656                  UIPAR7= 177656          ;PATTERN PROGRAM SPACE

                                ;*USER "D" PAGE ADDRESS REGISTERS
                        ;UDPAR0=         177660         ;PATTERN PROGRAM SPACE
                        ;UDPAR1=         177662         ;PATTERN PROGRAM SPACE
                        ;UDPAR2=         177664         ;PATTERN PROGRAM SPACE
                        ;UDPAR3=         177666         ;PATTERN PROGRAM SPACE
                        ;UDPAR4=         177670         ;PATTERN PROGRAM SPACE
                        ;UDPAR5=         177672         ;PATTERN PROGRAM SPACE
                        ;UDPAR6=         177674         ;PATTERN PROGRAM SPACE
```

```
                177676          UDPAR7= 177676          ;PATTERN PROGRAM SPACE

                                ;*SUPERVISOR "I" PAGE DESCRIPTOR REGISTERS
                172200          SIPDR0= 172200
                                ;SIPDR1=        172202
                                ;SIPDR2=        172204
                                ;SIPDR3=        172206
                                ;SIPDR4=        172210
                                ;SIPDR5=        172212
                                ;SIPDR6=        172214
                                ;SIPDR7=        172216

                                ;*SUPERVISOR "D" PAGE DESCRIPTOR REGISTERS
                                ;SDPDR0=        172220
                                ;SDPDR1=        172222
                                ;SDPDR2=        172224
                                ;SDPDR3=        172226
                                ;SDPDR4=        172230
                                ;SDPDR5=        172232
                                ;SDPDR6=        172234
                172236          SDPDR7= 172236

                                ;*SUPERVISOR "I" PAGE ADDRESS REGISTERS
                172240          SIPAR0= 172240
                                ;SIPAR1=        172242
                                ;SIPAR2=        172244
                172246          SIPAR3= 172246          ;TEST AREA
                                ;SIPAR4=        172250          ;TEST AREA
                                ;SIPAR5=        172252          ;TEST AREA
                                ;SIPAR6=        172254          ;TEST AREA
                                ;SIPAR7=        172256

                                ;*SUPERVISOR "D" PAGE ADDRESS REGISTERS
                172260          SDPAR0= 172260
                                ;SDPAR1=        172262
                                ;SDPAR2=        172264
                                ;SDPAR3=        172266
                                ;SDPAR4=        172270
                                ;SDPAR5=        172272
                                ;SDPAR6=        172274
                172276          SDPAR7= 172276

                                ;*KERNEL "I" PAGE DESCRIPTOR REGISTERS
                172300          KIPDR0= 172300
                                ;KIPDR1=        172302
                                ;KIPDR2=        172304
                                ;KIPDR3=        172306
                                ;KIPDR4=        172310
                                ;KIPDR5=        172312
                                ;KIPDR6=        172314
                                ;KIPDR7=        172316

                                ;*KERNEL "D" PAGE DESCRIPTOR REGISTERS
                                ;KDPDR0=        172320
                                ;KDPDR1=        172322
                                ;KDPDR2=        172324
```

```
                                ;KDPDR3=        172326
                                ;KDPDR4=        172330
                                ;KDPDR5=        172332
                                ;KDPDR6=        172334
            172336              KDPDR7= 172336

                                        ;*KERNEL "I" PAGE ADDRESS REGISTERS
            172340              KIPAR0= 172340
                                ;KIPAR1=        172342
                                ;KIPAR2=        172344
                                ;KIPAR3=        172346
                                ;KIPAR4=        172350
                                ;KIPAR5=        172352
                                ;KIPAR6=        172354
                                ;KIPAR7=        172356

                                        ;*KERNEL "D" PAGE ADDRESS REGISTERS
            172360              KDPAR0= 172360
                                ;KDPAR1=        172362
                                ;KDPAR2=        172364
                                ;KDPAR3=        172366
                                ;KDPAR4=        172370
                                ;KDPAR5=        172372
                                ;KDPAR6=        172374
            172376              KDPAR7= 172376
```

```
                                .SBTTL  DEFINE  UNIBUS MAP REGISTERS
                                ;*THE LOWER 16 BITS OF THE MAP REGISTERS ARE LABELED 'MAPLXX'
                                ;*THE UPPER 6 BITS OF THE MAP REGISTERS ARE LABELED 'MAPHXX'
 170200                         MAPL0 = 170200
 170202                         MAPH0 = 170202
 170204                         MAPL1 = 170204
                                ;MAPH1 = 170206
                                ;MAPL2 = 170210
                                ;MAPH2 = 170212
                                ;MAPL3 = 170214
                                ;MAPH3 = 170216
                                ;MAPL4 = 170220
                                ;MAPH4 = 170222
                                ;MAPL5 = 170224
                                ;MAPH5 = 170226
                                ;MAPL6 = 170230
                                ;MAPH6 = 170232
                                ;MAPL7 = 170234
                                ;MAPH7 = 170236
                                ;MAPL10 = 170240
                                ;MAPH10 = 170242
                                ;MAPL11 = 170244
                                ;MAPH11 = 170246
                                ;MAPL12 = 170250
                                ;MAPH12 = 170252
                                ;MAPL13 = 170254
                                ;MAPH13 = 170256
                                ;MAPL14 = 170260
                                ;MAPH14 = 170262
                                ;MAPL15 = 170264
                                ;MAPH15 = 170266
                                ;MAPL16 = 170270
                                ;MAPH16 = 170272
                                ;MAPL17 = 170274
                                ;MAPH17 = 170276
                                ;MAPL20 = 170300
                                ;MAPH20 = 170302
                                ;MAPL21 = 170304
                                ;MAPH21 = 170306
                                ;MAPL22 = 170310
                                ;MAPH22 = 170312
                                ;MAPL23 = 170314
                                ;MAPH23 = 170316
                                ;MAPL24 = 170320
                                ;MAPH24 = 170320
                                ;MAPL25 = 170324
                                ;MAPH25 = 170326
                                ;MAPL26 = 170330
                                ;MAPH26 = 170332
                                ;MAPL27 = 170334
                                ;MAPH27 = 170336
                                ;MAPL30 = 170340
                                ;MAPH30 = 170342
                                ;MAPL31 = 170344
                                ;MAPH31 = 170346
                                ;MAPL32 = 170350
```

```
                                ;MAPH32 = 170352
                                ;MAPL33 = 170354
                                ;MAPH33 = 170356
                                ;MAPL34 = 170360
                                ;MAPH34 = 170362
                                ;MAPL35 = 170364
                                ;MAPH35 = 170366
170370                          MAPL36 = 170370
170372                          MAPH36 = 170372
                                ;MAPL37 = 170374
170376                          MAPH37 = 170376

                                .SBTTL  DEFINE  SOFTWARE SWITCH & DISPLAY REGISTERS
000174                          DISPREG=174
000176                          SWREG=  176

                                .SBTTL  DEFINE  MK11 REGISTERS
172100                          MK11ADD=172100

                                .SBTTL  DEFINE  PARAMETERS
060000                          FIRST=60000             ;START OF THE 16K TEST PATTERN AREA
157776                          LAST=157776             ;END OF THE 16K TEST PATTERN AREA
040000                          SIZE=40000              ;SIZE OF THE 16K TEST PATTERN AREA (FOR SOB INST
```

```
                        .LIST   MD                      ;BE NICE TO SEE MY DEFINITIONS
                        .SBTTL  MACRO   FATAL
                ;********************** FATAL ****************************
                ;
                ;FATAL IS USED TO REPORT FATAL ERRORS (ERRORS THAT PREVENT
                ;       THE PROGRAM FROM CONTINUING).
                ;
                ;*********************************************************
                        .MACRO  FATAL   ARG             ;***MACRO***MACRO***MACRO***
                        .NLIST
                        .IIF DF LST$$ .LIST  ME
                        .LIST
                        INC     FATAL$                  ;SET FATAL INDICATOR
                        ERROR   +ARG
                        .IIF DF LST$$ .NLIST  ME
                        .ENDM   FATAL

                        .SBTTL  MACRO   TYPE
                        .MACRO  TYPE    ARG
                        .NLIST
                        .IIF DF LST$$ .LIST  ME
                        .LIST
                        .IF B ARG
                        TYPEIT
                        .IFF
                        TYPEIT  ,ARG
                        .ENDC
                        .IIF DF LST$$ .NLIST  ME
                        .ENDM   TYPE
```

```
                        .SBTTL  MACRO   NEWTST
                ;********************* NEWTST *********************
                ;NEWTST IS USED AS THE FIRST INSTRUCTION OF A TEST.
                ;IT WILL:
                ;1)  GENERATE A TEST NUMBER FOR THE LABEL OF THIS TEST
                ;2)  PUT STARS BEFORE AND AFTER A MESSAGE
                ;ARGUMENTS
                ;1)  ASCII   --  THIS IS THE MESSAGE THAT WILL APPEAR
                ;                ON THE LISTING
                ;2)  ICOUNT --   IF NON-BLANK AND BIT 11 OF $SWR = 1 IT WILL BE
                ;                THE NUMBER OF ITERATIONS TO MAKE ON THIS TEST
                ;3)  RETURN --   IF NON-BLANK WILL BE THE ADDRESS TO
                ;                WHICH THE NEXT SCOPE STATEMENT WILL
                ;                LOOP BACK TO.
                ;4)  COMAND --   IF NON-BLANK WILL BE THE FIRST
                ;                INSTRUCTION OF THE TEST
                ;                IF BLANK SCOPE WILL BE THE
                ;                FIRST INSTRUCTION
                ;*************************************************
                        .MACRO  NEWTST  ASCII,ICOUNT,RETURN,COMAND
                        $STN=1
                        $NWTST=0
                        .NLIST  MC
                        .IF B <COMAND>
                        $$NEWTEST       \$TN,<ASCII>,SCOPE
                        .IFF
                        $$NEWTEST       \$TN,<ASCII>,<COMAND>
                        .ENDC
                        .NLIST
                        .LIST   ME
                        .LIST
                        .IF NE 4000&$SWR
                        .IF NB ICOUNT
                        .IF LE <ICOUNT-1>
                        MOV     #1,$TIMES       ;;DO 1 ITERATION
                        .IFF
                        MOV     #ICOUNT,$TIMES  ;;DO ICOUNT ITERATIONS
                        .ENDC
                        .ENDC
                        .IF NB RETURN
                        MOV     #RETURN,$LPADR  ;;SET SCOPE LOOP ADDRESS
                        .ENDC
                        .ENDC
                        .NLIST
                        .LIST   MC
                        .LIST
                        .NLIST  ME
                        .ENDM   NEWTST
```

```
                        .SBTTL  MACRO   $$NEWTEST
                        .MACRO  $$NEWTEST       A,ASC,COMND
                        .IRP    ASCI,<ASC>
                        .IF EQ $NWTST
                        $NWTST=1
                        .SBTTL  T'A'    ASCI
                        .NLIST
                        .LIST   ME
                        .LIST
                ;*******************************************************************
                ;*TEST A         ASCI
                        .IFF
                        ASCI
                        .ENDC
                        .ENDM
                ;*******************************************************************
                TST'A:  COMND
                        .NLIST  ME
                        $TN=$TN+1
                        .ENDM   $$NEWTEST

                        .SBTTL  MACRO   SUBTST
                ;************** SUBTST ******************************
                ;
                ;THIS MACRO WILL FORMAT A SUBTEST HEADING WITH STARS
                ;A .SBTTL WILL BE FORCED & .NLISTED FOR THE TABLE OF CONTENTS.
                ;
                ;ARGUMENT:
                ;1) TXT --       THIS IS THE MESSAGE THAT WILL APPEAR IN THE TABLE OF CONTENTS & LISTING.
                ;
                ;EXAMPLE:        SUBTST  <<THIS IS A FUN SUBTST>>
                ;
                ;*******************************************************

                        .MACRO  SUBTST  ASCII
                        .NLIST  MC
                        $SUBTST <ASCII>
                        .LIST   MC
                        .ENDM   SUBTST

                        .SBTTL  MACRO   $SUBTST
                        .MACRO  $SUBTST ASC
                        .IRP    ASCI,<ASC>
                        .SBTTL  ASCI
                        .NLIST
                        .LIST   ME
                        .LIST
                ;*******************************************************************
                ;*SUBTEST        ASCI
                        .ENDM
                ;*******************************************************************
                        .NLIST  ME
                        .ENDM   $SUBTST
```

```
                        .SBTTL  MACRO   TYPOCT
                ;******************** TYPOCT ********************
                ;
                ;TYPOCT IS USED TO CHANGE A BINARY NUMBER
                ;       TO A 6 DIGIT OCTAL NUMBER AND TYPE IT
                ;
                ;ARGUMENTS:
                ;
                ;1)     NUM     THE NUMBER TO BE TYPED
                ;
                ;2)     REMARK  ALLOWS A COMMENT TO BE MADE
                ;
                ;ROUTINES REQUIRED
                ;
                ;1)     CONVERT BINARY TO OCTAL AND TYPE (.$TYPOCT)
                ;
                ;2)     TYPE AN ASCIZ STRING (.$TYPE)
                ;
                ;EXAMPLES:
                ;
                ;1)     TYPOCT  HILMT,<TYPES THE CONTENTS OF HILMT>
                ;2)     TYPOCT  #5,<TYPES ' 000005'>
                ;
                ;*****************************************************************

                        .MACRO  TYPOCT  NUM,REMARK
                        .NLIST
                        .IIF DF LST$$ .LIST  ME
                        .LIST
                        MOV     NUM,-(SP)       ;;SAVE NUM FOR TYPOUT
                        .IIF NB <REMARK>,
                                                        ;;REMARK
                        TYPOC                   ;;GO TYPE--OCTAL ASCII(ALL DIGITS)
                        .IIF DF LST$$ .NLIST  ME
                        .ENDM   TYPOCT
```

```
                        .SBTTL  MACRO   TYPOCS
                ;********************* TYPOCS *********************
                ;
                ;TYPOCS IS USED TO CHANGE A BINARY NUMBER TO AN OCTAL
                ;       NUMBER AND TYPE 1 TO 6 DIGITS
                ;       WITH OR WITHOUT LEADING ZEROS.
                ;
                ;ARGUMENTS:
                ;
                ;1)     NUM     NUMBER TO BE TYPED
                ;
                ;2)     REMARK  ALLOWS A COMMENT TO BE MADE
                ;
                ;3)     N       NUMBER OF DIGITS (1 TO 6) TO BE TYPED
                ;
                ;4)     Z       BLANK=SUPPRESS LEADING ZEROS (TYPES SPACES)
                ;               NON-BLANK=TYPE LEADING ZEROS
                ;
                ;ROUTINES REQUIRED
                ;
                ;1)     CONVERT BINARY TO OCTAL AND TYPE (.$TYPOCT)
                ;
                ;2)     TYPE AN ASCIZ STRING (.$TYPE)
                ;
                ;EXAMPLES:
                ;1)     TYPOCS  #12345,<TYPES "5">,1
                ;2)     TYPOCS  #004,<TYPES "04">,2,X
                ;3)     TYPOCS  #004,<TYPES " 4">,2
                ;
                ;*****************************************************************

                        .MACRO  TYPOCS  NUM,REMARK,N,Z
                        .NLIST
                        .IIF DF LST$$ .LIST  ME
                        .LIST
                MOV     NUM,-(SP)               ;;SAVE NUM FOR TYPEOUT
                        .IIF NB <REMARK>,                       ;;REMARK
                TYPOS                           ;;GO TYPE--OCTAL ASCII
                        .IF NB N
                        .BYTE   N               ;;TYPE N DIGIT(S)
                        .IFF
                        .BYTE   6               ;;TYPE 6 DIGITS
                        .ENDC
                        .IF NB Z
                        .BYTE   1               ;;TYPE LEADING ZEROS
                        .IFF
                        .BYTE   0               ;;SUPPRESS LEADING ZEROS
                        .ENDC
                        .IIF DF LST$$ .NLIST  ME
                        .ENDM   TYPOCS
```

```
                        .SBTTL  MACRO   TYPDEC
                ;******************** TYPDEC ********************
                ;
                ;TYPDEC IS USE TO CHANGE A BINARY NUMBER TO A SIGNED
                ;       DECIMAL NUMBER AND TYPE IT REPLACING LEADING ZERO
                ;       WITH SPACES.
                ;NOTE:  IF THE NUMBER IS NEGATIVE A
                ;       MUNUS SIGN WILL BE TYPED.
                ;
                ;ARGUMENTS:
                ;
                ;1)     NUM     NUMBER TO BE TYPED
                ;
                ;2)     REMARK ALLOWS A COMMENT TO BE MADE
                ;
                ;ROUTINES REQUIRED
                ;
                ;1)     CONVERT BINARY TO DECIMAL AND TYPE (.$TYPDEC)
                ;
                ;2)     TYPE AN ASCIZ STRING (.$TYPE)
                ;
                ;EXAMPLES
                ;
                ;1)     TYPDEC  SIZE,<TYPE THE CONTENTS OF SIZE>
                ;2)     TYPDEC  #-10.,<TYPE A MINUS TEN>
                ;
                ;*************************************************************

                        .MACRO  TYPDEC  NUM,REMARK
                        .NLIST
                        .IIF DF LST$$ .LIST  ME
                        .LIST
                        MOV     NUM,-(SP)       ;;SAVE NUM FOR TYPEOUT
                        .IIF NB <REMARK>,                       ;;REMARK
                        TYPDS                   ;;GO TYPE--DECIMAL ASCII WITH SIGN
                        .IIF DF LST$$ .NLIST  ME
                        .ENDM   TYPDEC
```

```
                        .SBTTL  MACRO   BMOV
                ;****************** BMOV ****************************
                ;
                ; THIS MACRO MOVES A BLOCK OF DATA.
                ;
                ;ARGUEMENTS:
                ;
                ;1)      FROMHERE        THE FIRST ADDRESS OF THE SOURCE BLOCK.
                ;
                ;2)      TOHERE          THE FIRST ADDRESS OF THE DESTINATION BLOCK.
                ;                        IF BLANK THE 1ST ADDRESS OF THE USER INSTRUCTION
                ;                        PAR'S IS USED (FASTCITY).
                ;
                ;3)      SIZE            THE SIZE OF THE SOURCE BLOCK.
                ;                        IF BLANK A 16 WORD TRANSFER IS ASSUMED.
                ;                        "WHY DEFAULT TO 16 WORDS?" YOU AS..!
                ;                        "BECAUSE THAT'S HOW MANY WORDS TO THE USER PAR
                ;                        REGISTERS & THAT'S WHERE I INTEND TO MOVE LOTS
                ;                        OF STUFF." I REPLY!
                ;
                ;************************************************

                .MACRO  BMOV    FROMHERE,TOHERE,SIZE
                  .IF B TOHERE
                    .NLIST
                    .IIF DF LST$$ .LIST  ME
                    .LIST
                    JSR R5,BLOCK1
                    FROMHERE
                    .IIF DF LST$$ .NLIST  ME
                    .MEXIT
                  .ENDC
                  .IF NB SIZE
                    .NLIST
                    .IIF DF LST$$ .LIST  ME
                    .LIST
                    JSR R5,BLOCK2
                    TOHERE
                    FROMHERE
                    .IIF DF LST$$ .NLIST  ME
                    .MEXIT
                  .IFF
                    .NLIST
                    .IIF DF LST$$ .LIST  ME
                    .LIST
                    JSR R5,BLOCK3
                    SIZE
                    TOHERE
                    FROMHERE
                    .IIF DF LST$$ .NLIST  ME
                  .ENDC
                .ENDM   BMOV
```

```
                .SBTTL  MACRO   MAP
        ;***************** MAP ****************************
        ;
        ; THIS MACRO MAPS A MEMORY BANK (16K) INTO THE
        ; TEST PATTERN AREA (SUPERVISOR VIRTUAL (60000-157777)).
        ;
        ;ARGUEMENTS:
        ;
        ;1)     BANK    THE BANK OF 16K WORDS TO BE MAPPED.
        ;               THERE ARE 120 BANKS OF 16K WORDS
        ;
        ;EXAMPLES
        ;
        ;       MAP     LOC             ;LOCATION "LOC" CONTAINS THE # OF THE BANK TO MAP
        ;
        ;       MAP     #28.            ;BANK 34 (OCTAL) WILL BE MAPPED
        ;
        ;*************************************************

                .MACRO  MAP     BANK
                PUSH    R3
                .NLIST
                .IIF DF LST$$ .LIST  ME
                .LIST
                .IF B BANK
                MOV     #120.,R3
                .IFF
                MOV     BANK,R3
                .ENDC
                CALL    MAPPER
                .IIF DF LST$$ .NLIST  ME
                POP     R3
                .ENDM   MAP

                .SBTTL  MACRO   SUPERVISOR
        ;***************** SUPERVISOR *********************
        ;
        ; THIS MACRO SWITCHES TO SUPERVISOR MODE.
        ;
        ;ARGUEMENTS: NONE.
        ;
        ;*************************************************

                .MACRO  SUPERVISOR
                .NLIST
                .IIF DF LST$$ .LIST  ME
                .LIST
                BIS     #BIT14,PSW              ;DO IT TO IT!
                .IIF DF LST$$ .NLIST  ME
                .ENDM   SUPERVISOR

                .SBTTL  MACRO   USER
        ;***************** USER ***************************
        ;
        ; THIS MACRO SWITCHES TO USER MODE.
        ;
```

```
                        ;ARGUEMENTS: NONE.
                        ;
                        ;*****************************************************

                                .MACRO  USER
                                .NLIST
                                .IIF DF LST$$ .LIST  ME
                                .LIST
                                BIS     #BIT15!BIT14,PSW                ;DO IT TO IT!
                                .IIF DF LST$$ .NLIST  ME
                                .ENDM   USER
```

```
                        .SBTTL  MACRO   SET
                ;****************** SET ****************************
                ;
                ; THIS MACRO SIMULATES A SETO INSTRUCTION ON THE PDP10
                ; (SET TO ONES).
                ;
                ;ARGUEMENTS:    LOC     ;THE LOCATION TO BE SET
                ;
                ;*****************************************************

                        .MACRO  SET     ARG
                        .NLIST
                        .IIF DF LST$$ .LIST  ME
                        .LIST
                        MOV     #-1,ARG
                        .IIF DF LST$$ .NLIST  ME
                        .ENDM   SET

                        .SBTTL  MACRO   SET4 & RES4
                ;****************** SET4 & RES4 ********************
                ;
                ; THESE MACROS SET & RESTORE VECTOR 4(TIMEOUT TRAP)
                ;
                ; IN IT'S RESTORED MODE TRAPS ARE REPORTED AS SUCH.
                ;
                ;ARGUEMENTS:    LOC     ;THE LOCATION TO VECTOR TO (ONLY USED IN "SET4" NOT "RES4")
                ;
                ;*****************************************************

                        .MACRO  SET4    ARG
                        .NLIST
                        .IIF DF LST$$ .LIST  ME
                        .LIST
                        MOV     ARG,4                   ;SET NEW VECTOR
                        .IIF DF LST$$ .NLIST  ME
                        .ENDM   SET4

                        .MACRO  RES4
                        .NLIST
                        .IIF DF LST$$ .LIST  ME
                        .LIST
                        MOV     #TIMEOUT,4              ;RESTORE OLD (DEFAULT) VECTOR
                        CLR     CPUERR
                        .IIF DF LST$$ .NLIST  ME
                        .ENDM   RES4

                        .SBTTL  MACRO   DLEFT
                ;****************** DLEFT ***************************
                ;
                ; THIS MACRO DOES A DOUBLE WORD LEFT SHIFT
                ;
                ;ARGUEMENTS:    LOC     ;THE LOCATION TO BE SHIFTED LEFT (CARRY TO LOC+2)
                ;
                ;*****************************************************

                        .MACRO  DLEFT   ARG
```

```
                                        .NLIST
                                        .IIF DF LST$$ .LIST  ME
                                        .LIST
                                        ROL     ARG
                                        ROL     ARG+2
                                        .IIF DF LST$$ .NLIST  ME
                                        .ENDM   DLEFT
                                        .NLIST  MD              ;DON'T NEED TO SEE THEM ANY MORE
            000000                      .ASECT
                                        .TITLE  COMMON

                                        ;TRAP CATCHER
            000000                      .=0
000000      000000  000000              .WORD   0,0

                                        .SBTTL  ACT11 HOOKS
                                ;*THE HOOKS REQUIRED BY ACT11 ARE DEFINED AND SETUP BELOW:
                                ;*      DEFINATIONS:
                                ;*                      1)LOC.46        "END-OF-PASS" HOOK
                                ;*                              =ADDRESS OF END OF PASS ROUTINE
                                ;*                               MODIFIED BY ACT11.
                                ;*                      2)LOC.52        PROGRAM NEEDS HOOK
                                ;*                              BIT 15=1 PROGRAM SHOULD BE POWER
                                ;*                                       FAILED WHILE RUNNING
                                ;*                                    =0 NO POWER FAIL
                                ;*                              BIT 14=1 PROGRAM MEMORY SIZE DEPENDENT
                                ;*                                    =0 NOT MEMORY SIZE DEPENDENT
                                ;*                              BIT 13=1 PROGRAM REQUIRES MANUAL INTERVENTION
                                ;*                                    =0  MANUAL INTERVENTION NOT REQUIRED
                                ;*                              BITS 12-0  MUST BE ZERO'S
            000046              .=46
000046      011650              $ENDAD                          ;;1)SET LOC.46 TO ADDRESS OF $ENDAD IN .$EOP
            000052              .=52
000052      000020              .WORD   BIT4                    ;;2)SET LOC.52 TO INDICATE MEMORY SIZE DEPENDANT
                                .SBTTL  APT11 HOOKS
            000024              .=24    ;;SET POWER FAIL TO POINT TO START OF PROGRAM
000024      000200              200     ;;FOR APT START UP
            000042              .=42
000042      001400              $TACK                           ;SO RT11 CAN START WITH RUN COMMAND
            000044              .=44    ;;POINT TO APT INDIRECT ADDRESS PNTR.
000044      060264              $APTHDR ;;POINT TO APT HEADER BLOCK
            000200              .=200
000200      000437              BR      START1
000202      000442              BR      START2
            000300              .=300
000300      005037  002052      START1: CLR     RESTART
000304      000137  003614              JMP     START
000310                          START2: SET     RESTART
(1) 000310  012737  177777  002052      MOV     #-1,RESTART
000316      000137  003614              JMP     START
            001400              .=$TACK
                                .SBTTL  VARIABLES       INITIALIZED TO ZERO
                                ;*THIS TABLE CONTAINS VARIOUS COMMON STORAGE LOCATIONS
                                ;*USED IN THE PROGRAM.
001400                          $CMTAG:                         ;;START OF COMMON TAGS
001400  000000                  .WORD   0
```

```
                                        ;NEXT TWO BYTES ARE DISPLAYED IN THE DISPLAY REGISTER
001402  000             $PATMAR:.BYTE   0               ;PATTERN NUMBER & MARGIN VALUE
001403  000             $BANK:  .BYTE   0               ;BANK & SIGN = MARGIN INDICATOR
001404  000             $ERFLG: .BYTE   0               ;;CONTAINS ERROR FLAG
001405                  .EVEN
001406  000000          $ERTTL: .WORD   0               ;;CONTAINS TOTAL ERRORS DETECTED
001410  000000          LASTERROR:.WORD 0               ;ERRORS ON LAST PASS
001412  000000          SBEFLAG:.WORD   0               ;A SINGLE BIT ERROR HAS OCCURED THIS PASS
001414  000             $ITEMB: .BYTE   0               ;;CONTAINS ITEM CONTROL BYTE
001416                  .EVEN
001416  000000          ERRORPC:.WORD   0               ;CONTAINS PC OF LAST ERROR INSTRUCTION
001420  000000          BADPC:  .WORD   0               ;CONTAINS PC OF WHERE THE ERROR OCCURED
001422  000000          ADDRESS:.WORD   0               ;;CONTAINS ADDRESS OF 'BAD' DATA
001424  000000          PADDRESS:.WORD  0               ;ADDRESS OF PARITY ERROR
001426  000000  000000  PHYADD: .WORD   0,0             ;22 BIT PHYSICAL ADDRESS
001432  000000          GOOD:   .WORD   0               ;;CONTAINS 'GOOD' DATA
001434  000000          GOOD2:  .WORD   0               ;;CONTAINS 'GOOD2' DATA
001436  000000          GOOD3:  .WORD   0               ;;CONTAINS 'GOOD3' DATA
001440  000000          BAD:    .WORD   0               ;;CONTAINS 'BAD' DATA
001442  000000          BAD2:   .WORD   0               ;;CONTAINS 'BAD2' DATA
001444  000000          BAD3:   .WORD   0               ;;CONTAINS 'BAD3' DATA
001446  000000          BADXOR: .WORD   0               ;XOR OF GOOD & BAD = BAD BITS!
001450  000000          $AUTO:  .WORD   0               ;;AUTOMATIC MODE INDICATOR
001452  000             $INTAG: .BYTE   0               ;;INTERRUPT MODE INDICATOR
001454                  .EVEN
001454  000000          FATAL$: .WORD   0               ;FATAL ERROR INDICATOR
001456  000000          NEMCNT: 0                       ;NON-EXISTANT MEMORY COUNTER (HOLES)
001460  000000          PARCNT: 0                       ;PARITY ERROR COUNTER
001462  000000          PATERR: 0                       ;PATTERN ERROR COUNTER
001464  000000          NOPAR:  0                       ;NO PARITY ERROR FLAG
001466  000000          NONEM:  0                       ;NO NON-EXISTANT MEMORY (HOLES) FLAG
001470  000000          BANK:   0                       ;MEMORY BANK UNDER TEST
001472  000000          BANKINDEX:0                     ;USED TO INDEX INTO CONFIG TABLE
001474  000000          MARGIN: 0                       ;INDICATES MARGINS IN PROGRESS
001476  000000          CPUBIT: 0                       ;CONTAINS 1 BIT TO IDENTIFY CPU TO CONFIGURATION TABLE
001500  000000          MUT:    0                       ;MEMORY UNDER TEST FLAG
001502  000000          PATTERN:0                       ;PATTERN UNDER TEST
001504  000000          KPFLAG: 0                       ;BANK IS PROTECTED REGION OF MK11
001506  000000          ACFLAG: 0                       ;BANK CAN BE ACCESSED BY THIS CPU
001510  000000          MKFLAG:0                        ;IF SET INDICATES MK11 UNDER TEST
001512  000000          PFLAG:  0                       ;BANK IS IN PROGRAM SPACE
001514  000000          RRFLAG: 0                       ;BANK IS IN SPOT WHERE PROGRAM RELOCATION IS REQUIRED TO
001516  000000          RLFLAG: 0                       ;PROGRAM IS RELOCATED FLAG
001520  000000          BMFLAG: 0                       ;BAD MEMORY FLAG
001522  000000          TMFLAG: 0                       ;TYPE OF MEMORY TO TEST FLAG; 0 = MK11
001524  000000          ABORTFLAG:0                     ;ABORT INDICATOR
001526  000000  000000  CSR:      .WORD   0,0           ;DATA TO OR FROM CSR
001532  000010          CSR0:     .BLKW   8.            ;SAVE CSR'S AFTER GETSIZE CALL
001552  000010          CSR2:     .BLKW   8.            ;SAVE CSR+2'S AFTER GETSIZE CALL
001572  000010          CSRFB:    .BLKW   8.            ;SAVE FIRSTB'S AFTER GETSIZE CALL
001612  000010          CSRLB:    .BLKW   8.            ;SAVE LASTB'S AFTER GETSIZE CALL
001632  000000          CSRREL: 0                       ;CSR'S ARE RELOCATED TO SHADOW MEMORY FLAG
001634  000000          CSRNO:  0                       ;CSR ADDRESS NUMBER (3 LSB'S)
001636  000000          FIRSTB: 0                       ;FIRST BANK
001640  000000          LASTB:  0                       ;LAST BANK
001642  000000          INTERL: 0                       ;INTERLEAVE FACTOR (0,1,2,3)
```

```
                                        ;THESE LOCATIONS STORE GPR'S DURING SUPERVISOR TESTS
001644  000000          SUPDR0: 0
001646  000000          SUPDR1: 0
001650  000000          SUPDR2: 0
001652  000000          SUPDR3: 0
001654  000000          SUPDR4: 0
001656  000000          SUPDR5: 0
001660  000000          SUPDR6: 0
001662  000000          DUMMY:  0                       ;DUMMY LOCATION FOR ADDRESS PASSING
                                        ;THESE LOCATIONS STORE GPR'S DURING DETAILED ERROR PRINTOUTS
001664  000000          DETR0:  0
001666  000000          DETR1:  0
001670  000000          DETR2:  0
001672  000000          DETR3:  0
001674  000000          DETR4:  0
001676  000000          DETR5:  0
001700  000000          DETR6:  0
001702  000000          DETFLAG:0                       ;DETAILED REPORT FLAG
001704  000000          CONTFLAG:0                      ;MK11 CONTROL'S HAVE BEEN TESTED FLAG
001706  000000          MKCSRS: .WORD   0       ;LO BYTE;1 BIT PER EXISTING CSR
                                                        ;CSR'S 0,1,2,3,4,5,6,7 REPRESENTED BY BITS
                                                        ;7,6,5,4,3,2,1,0 RESPECTIVELY
                                                ;HI BYTE;1 BIT PER EXISTING CSR WITH INTERNAL INTERLEAVE
                                                        ;INTERNAL INTERLEAVE REPRESENTED BY BITS
                                                        ;15,14,13,12,11,10,9,8 RESPECTIVELY
001710  000000          CSRFIRST:0                      ;FIRST ADDRESS UNDER CONTROL OF THIS CSR
001712  000000  000000  DATBUF: .WORD   0,0            ;TWO WORD DATA BUFFER
001716  000000  000000  TSTDAT: .WORD   0,0            ;TWO WORD TEST DATA
001722  000000  000000  SBEMSK: .WORD   0,0            ;TWO WORD SINGLE BIT ERROR MASK
001726  000000  000000  DBEMSK: .WORD   0,0            ;TWO WORD DOUBLE BIT ERROR MASK
001732  000000          SUPDOADD:0                      ;ADDRESS OF SUBROUTINE TO EXECUTE IN SUPERVISOR MODE
001734      000         PASFLG: .BYTE   0               ;LOCAL LOOP PASS CONTROL
001735      000         UPPFLG: .BYTE   0               ;LOCAL LOOP PASS CONTROL
001736  000000          LOWBNK: 0                       ;LOWBNK CONTAINS ADDRESS OF THE LOWEST LOCATION
                                                        ;IN THE SEMIBANK (4K) THAT CAN BE TESTED
001740  000000          STRTDI: 0                       ;LOAD STRTDI WITH THE STARTING ADDRESS OF THE DIAGONAL
001742  000000          GALWORD:0
001744  000000          REALPAT:0                       ;REAL PATTERN UNDER TEST
001746  000000          OLDCACHE:0                      ;BACKED UP VALUE OF CACHE CONTROL REGISTER
001750  000000          PARTHERE:0              ;PARITY TRAPS GO TO ADDRESS STORED HERE
001752  000000          FSSTACK:0                       ;STACK SAVED HERE IF IN FIELD SERVICE MODE
001754  000000          OLDMARGIN:0                     ;BACKED UP VALUE OF LAST MARGIN SETTING
001756  000000          NEWBANK:0                       ;USED FOR RELOCATION TO A NEW BANK
001760  000000          SOURCE:  .WORD 0        ;SOURCE OF DATA WORDS FOR CHECKBIT GENERATION SUBROUTINE
001762  000000          CHECK:   .WORD 0        ;CHECKBITS TO BE LOADED INTO CSR
001764  000000          PCBUMP:  .WORD 0                ;VALUE TO BUMP THE PC BY AFTER A PARITY TRAP
001766  000000          CSRFBANK:0                              ;FIRST BANK OWNED (AT LEAST IN PART) BY THIS CSR
001770  000000          CSRLBANK:0                              ;LAST BANK OWNED (AT LEAST IN PART) BY THIS CSR
001772  000000          CSRINC:  0                      ;VALUE TO INCREMENT ADDRESS BY TO REMAIN IN THE SAME CSR
001774      000         CSRINDEX:.BYTE  0               ;AN INDEX VARIABLE FOR A CSR LOOP
001775      000         CSROUDEX:.BYTE  0               ;AN INDEX VARIABLE FOR A CSR LOOP
001776  000000          CSRLOOP:0                       ;LOOP CONTROL FOR CSR TESTING
002000  000000          SIDE:    0                      ;INDEX VARIABLE FOR EACH SIDE (CONTROLLER) OF A CSR
002002  000000          SUCCESS:0                       ;FLAG SET BY SUCCESSFULL TASK OR SUBROUTINE
002004  000000          ZERO:    0                      ;FOR AID IN MOVE INSTRUCTIONS
002006  000000          ZEROS:   0                      ;FOR AID IN MOVE INSTRUCTIONS
```

```
002010  000000  SPECIAL:0                  ;FLAG FOR SPECIAL CASE OF NO INTERNAL INTERLEAVE
002012  000000  TIME:   0                  ;SECONDS THAT BATTERIES SHOULD LAST
002014  000000  NULLFLAG:0                 ;SET WHEN RUNNING NULL PATTERNS
002016  000000  QVFLAG: 0                  ;FLAGS QUICK VERIFY PASS UNDER APT, ACT, OR XXDP CHAIN M
002020  000000  ACTFLAG:0                  ;FLAGS ACT AUTOMATIC MODE PROGRAMMING RULES
002022  000000  APTFLAG:0                  ;FLAGS APT AUTOMATIC MODE PROGRAMMING RULES
002024  000000  XXDPCHAIN:0                ;FLAGS XXDP CHAIN MODE MODE PROGRAMMING RULES
                ;NOTE: THESE TWO BYTES MUST STAY TOGETHER
002026     000  $NULL:  .BYTE   0          ;;CONTAINS NULL CHARACTER FOR FILLS
002027     000  $FILLS: .BYTE   0          ;CONTAINS # OF FILL CHARACTERS
002030     000  $TPFLG: .BYTE   0          ;;"TERMINAL NOT AVAILABLE" FLAG
        002032          .EVEN
002032  000000  $ESCAPE:0                  ;;ESCAPE ON ERROR ADDRESS
002034  000000  EVEN:   0                  ;USED FOR ALTERNATE DATA PATTERNS
002036  000000  STRIPES:0                  ;COUNTS DIAGONAL STRIPES
002040  000000  COUNT:  0                  ;BACKED UP COPY OF STRIPES
002042  000000  INTOVR: 0                  ;INTERNAL INTERLEAVE OVERRIDE FLAG
002044  000000  NOTAB:  0                  ;NO TABLE BEING PRINTED - NOW
002046  000000  MJSIZE: 0                  ;SIZE OF MJ11 MEMORY IN K WORDS
002050  000000  MKSIZE: 0                  ;SIZE OF MK11 MEMORY IN K WORDS
002052  000000  RESTART:        0                ;RESTART (START ADD 202) FLAG
002054  000000  TOOMANY:0                  ;FLAGS WHEN TOO MANY ERRORS HAVE BEEN PRINTED FOR A BANK
002056  000000  NOECCFLAG:0                ;ECC CAPABILITIE DISABLED ON THE CONTROL PANEL
002060  000000  GDSWITCH:0                 ;GOOD VALUE FOR SOME SWITCHES
002062  000000  BDSWITCH:0                 ;BAD VALUE FOR SOME SWITCHES
002064  000000  WRITEONLY:0                ;FLAG TO PATTERNS TO WRITE ONLY
002066  000000  READONLY:0                 ;FLAG TO PATTERNS TO READ ONLY
002070  000000  TESTADD:0                  ;THE ADDRESS TO RUN CSR TESTS ON
002076  000000  HIBANK: 0                  ;HIGHEST ACCESSABLE BANK ON MEMORY
002100  000000  UNITOP: 0                  ;HIGHEST ACCESSABLE BANK OF MEMORY THRU UNIBUS MAP
002102  000000  STOPOK: 0                  ;FLAG TO ALLOW STOPING WITH SWITCH REGISTER
002104  000000  APTCORE:0                  ;AMOUNT OF CORE MEMORY ACCORDING TO APT
002106  000000  APTMOS: 0                  ;AMOUNT OF MOS MEMORY ACCORDING TO APT
002110  000000  NOFSMODE:0                 ;FLAG TO DISABLE FIELD SERVICE MODE
002112  000000  NOERROR:0                  ;THIS IS NOT AN ERROR FLAG
002114  000000  LOADHOME:0                 ;HOME BANK OF LOADERS
002116  000000  LOADBANK:0                 ;BANK LOADERS ARE RELOCATED TO
002120  000000  TEMP:   0                  ;USED FOR JUNK
002122  000000  QUICK:  0                  ;QUICK STOP FLAG FOR APT POWER FAIL
002124  000000  NOSCOPE:0                  ;NO SCOPE LOOP ALLOWED
002126  000000  FSINFLAG:       0          ;FIELD SERVICE "NO INTERNAL INTERLEAVE FLAG"
002130  000000  I:      0                  ;USED FOR GENERAL PURPOSE INDEXING
002132          $CMTGE: ;*END OF COMMON TAGS
```

D11


```
                                        .SBTTL  VARIABLES       INITIALIZED TO NON ZERO
002132  000001                  CACHK:  1                       ;CACHE CONSTANT (MOVED TO CONTROL TO TURN ON CACHE)
002134  000012                  ERRMAX: 10.                     ;MAX # OF ERRORS / BANK WITH SW11
002136  000031                  SOBK:   25.                     ;SOB CONSTANT
002140  177777                  WORST:  177777                  ;SET IF TESTING BANKS IN WORST FIRST MODE(1ST PASS)
002142  176543                  SEEDHI: 176543                  ;WORKING SEED HI
002144  123456                  SEEDLO: 123456                  ;WORKING SEED LO
002146  177777                  HEADER: 177777                  ;USED TO PRINT HEADINGS ONLY ONCE
002150  176543                  MSEEDH: 176543                  ;MASTER SEED HI
002152  123456                  MSEEDL: 123456                  ;MASTER SEED LO
002154  177777                  ONES:   177777                  ;ALL ONES
002156  077417                  PDRONES:77417                   ;PDR ALL ONES
002160  000003                  FLIPLOC:3                       ;LOCATION FOR FLIPING DATA ON WORST CASE NOISE TEST
002162  166000                  IISTACR:.WORD   166000          ;ADDRESS OF THE IIST ACR
002164  166002                  IISTADR:.WORD   166002          ;ADDRESS OF THE IIST ADR
002166  000170                  IISTVEC:.WORD   170             ;ADDRESS OF THE IIST VECTOR
002170  000000                  $LPADR: .WORD   0               ;;CONTAINS SCOPE LOOP ADDRESS
002172  000000                  $LPERR: .WORD   0               ;;CONTAINS SCOPE RETURN FOR ERRORS

                                ;********************** NOTE THESE TWO LOCATIONS MUST STAY TOGETHER ************
002174  000377                  BAKPAT: .WORD   377             ;BACKGROUND PATTERN                             *
002176  177400                  SWAPAT: .WORD   177400          ;SWAPPED BAKPAT                                 *
                                ;*******************************************************************************

002200  177570                  SWR:    .WORD   DSWR            ;;ADDRESS OF SWITCH REGISTER
002202  177570                  DISPLAY: .WORD  DDISP           ;;ADDRESS OF DISPLAY REGISTER
002204  177560                  $TKS:   177560                  ;;TTY KBD STATUS
002206  177562                  $TKB:   177562                  ;;TTY KBD BUFFER
002210  177564                  $TPS:   177564                  ;;TTY PRINTER STATUS REG. ADDRESS
002212  177566                  $TPB:   177566                  ;;TTY PRINTER BUFFER REG. ADDRESS
002214      012                 $FILLC: .BYTE   12              ;;INSERT FILL CHARS. AFTER A "LINE FEED"
002215      207 177777      000 $BELL:  .ASCIZ  <207><377><377> ;;CODE FOR BELL
002221      077                 $QUES:  .ASCII  /?/             ;;QUESTION MARK
002222      015                 $CRLF:  .ASCII  <15>            ;;CARRIAGE RETURN
002223      012         000     $LF:    .ASCIZ  <12>            ;;LINE FEED
            002226                      .EVEN
```

E11


```
                                        .SBTTL  TABLE   CONFIGURATION
                                ;CONFIG:FIRST 16K CONFIGURATION WORDS (3 EACH)
                                ;       2ND     16K CONFIGURATION WORDS (3 EACH)
                                ;       .....................
                                ;       167TH    16K CONFIGURATION WORDS (3 EACH)
                                ;
                                ;CONFIGURATION WORDS:
                                ;               LOW:    BIT 0-2   INTERLEAVE FACTOR
                                ;                       BIT 3     EXTERNAL INTERLEAVE
                                ;                       BIT 4     CPU CAN ACCESS
                                ;                       BIT 5-7   SPARE
                                ;                       BIT 8-10  BOX NUMBER (0-7)
                                ;                       BIT 11    ERRORS PRESENT
                                ;                       BIT 12    PROTECTED REGION OF A MK11 BOX
                                ;                       BIT 13-14 MEMORY TYPE;0=MJ11;1=MK11
                                ;                       BIT 15    PROTECTED (PROGRAM SPACE)
                                ;               MED:    BIT 0-5   NOT USED
                                ;                       BIT 6-8   MK11 CSR ADDRESS
                                ;                       BIT 9-14  NOT USED
                                ;                       BIT 15    LOADERS HOME BANK
                                ;               HIGH:   BIT 0     INDICATES BANK IS ADDRESSED BY CSR 172100
                                ;                       BIT 1     INDICATES BANK IS ADDRESSED BY CSR 172104
                                ;                       BIT 2     INDICATES BANK IS ADDRESSED BY CSR 172110
                                ;                       BIT 3     INDICATES BANK IS ADDRESSED BY CSR 172114
                                ;                       BIT 4     INDICATES BANK IS ADDRESSED BY CSR 172120
                                ;                       BIT 5     INDICATES BANK IS ADDRESSED BY CSR 172124
                                ;                       BIT 6     INDICATES BANK IS ADDRESSED BY CSR 172130
                                ;                       BIT 7     INDICATES BANK IS ADDRESSED BY CSR 172134
                                ;                       BIT 8-15  NUMBER OF ERRORS
002230  000550                  CONFIG: .BLKW   120.*3
003550                          CONFIEND:
```

```
                                        .TITLE  MAIN SKELETON

                                START:  SUBTST  <<INITIALIZE VARIABLES TO ZERO>>
                                ;*****************************************************************************
                                ;*SUBTEST        INITIALIZE VARIABLES TO ZERO
                                ;*****************************************************************************
                                        RESET
                                        MOV     #STACK,SP       ;;SETUP THE STACK POINTER
                                        IF PDP70M EQ ONES THEN JUMPTO BEGIN
                                                                                        CMP PDP70M,ONES
                                                                                        BNE LO
                                                                                        JMP BEGIN
                                                                                LO:;;;;;;
                                        MOV     #$CMTAG,R0      ;;FIRST LOCATION TO BE CLEARED
                                        CLR     (R0)+           ;;CLEAR MEMORY LOCATION
                                        CMP     #$CMTGE,R0      ;;DONE?
                                        BNE     .-6             ;;LOOP BACK IF NO
                                        SUBTST  <<CLEAR NON-PROGRAM SPACE>>
                                ;*****************************************************************************
                                ;*SUBTEST        CLEAR NON-PROGRAM SPACE
                                ;*****************************************************************************
                                        MOV     #1000,R0
                                9$:     CLR     (R0)+
                                        CMP     R0,#STACK
                                        BLO     9$
                                        MOV     #END+2,R0
                                8$:     CLR     (R0)+
                                        CMP     R0,#177776-3000.+2-8192.
                                        BLO     8$
                                        SUBTST  <<INITIALIZE VARIABLES TO NON ZERO>>
                                ;*****************************************************************************
                                ;*SUBTEST        INITIALIZE VARIABLES TO NON ZERO
                                ;*****************************************************************************
                                        SET     WORST
                                        MOV     #-1,WORST
                                        MOV     #3,FLIPLOC
                                        SET     HEADER
                                        MOV     #-1,HEADER
                                        MOV     #176543,MSEEDH
                                        MOV     #123456,MSEEDL
                                        MOV     MSEEDH,SEEDHI   ;PRIME THE RANDOM NUMBER GENERATOR
                                        MOV     MSEEDL,SEEDLO   ;BOTH HIGH AND LOW WORDS
                                        MOV     #377,BAKPAT
                                        MOV     #177400,SWAPAT
                                        MOV     #BEGIN,$LPADR   ;;INITIALIZE THE LOOP ADDRESS FOR SCOPE
                                        MOV     #BEGIN,$LPERR   ;;SETUP THE ERROR LOOP ADDRESS
                                        SUBTST  <<INITIALIZE VECTORS>>
                                ;*****************************************************************************
                                ;*SUBTEST        INITIALIZE VECTORS
                                ;*****************************************************************************
                                        MOV     #$SCOPE,IOTVEC  ;;IOT VECTOR FOR SCOPE ROUTINE
                                        MOV     #340,IOTVEC+2   ;;LEVEL 7
                                        MOV     #$ERROR,EMTVEC  ;;EMT VECTOR FOR ERROR ROUTINE
                                        MOV     #340,EMTVEC+2   ;;LEVEL 7
                                        MOV     #$TRAP,TRAPVEC  ;;TRAP VECTOR FOR TRAP CALLS
                                        MOV     #340,TRAPVEC+2;LEVEL 7
```

```
004114  012737  053552  000024          MOV     #$PWRDN,PWRVEC ;;POWER FAILURE VECTOR
004122  012737  000340  000026          MOV     #340,PWRVEC+2   ;;LEVEL 7
004130  012737  032234  000114          MOV     #PARITY,PARVEC;GET READY FOR PARITY ERRORS
004136  012737  000340  000116          MOV     #340,PARVEC+2
004144  012737  032512  000010          MOV     #PDP1105,RESVEC;RESERVED INSTRUCTION TRAP
004152  012737  000340  000012          MOV     #340,RESVEC+2
004160  012737  032466  000004          MOV     #TIMEOUT,ERRVEC;SETUP TIMEOUT ERRORS
004166  012737  000340  000006          MOV     #340,ERRVEC+2   ;SET PRIORITY OF ERROR TRAPS
004174  012737  032500  000250          MOV     #MMTRAP,MMVEC   ;VECTOR FOR MEMORY MANAGEMENT
004202  012737  000340  000252          MOV     #340,MMVEC+2
004210                                  SUBTST  <<INITIALIZE PATTERNS>>
                                ;*******************************************************************
                                ;*SUBTEST        INITIALIZE PATTERNS
                                ;*******************************************************************
004210  012700  060250                  MOV     #$DDW0,R0
004214  012001                          MOV     (R0)+,R1
004216  012703  014676                  MOV     #MKCSRT,R3
004222  004737  004276                  CALL    PATPLUG
004226  012001                          MOV     (R0)+,R1
004230  004737  004276                  CALL    PATPLUG
004234  012001                          MOV     (R0)+,R1
004236  012703  016056                  MOV     #MKPAT,R3
004242  004737  004276                  CALL    PATPLUG
004246  012001                          MOV     (R0)+,R1
004250  004737  004276                  CALL    PATPLUG
004254  012001                          MOV     (R0)+,R1
004256  012703  016212                  MOV     #MJPAT,R3
004262  004737  004276                  CALL    PATPLUG
004266  012001                          MOV     (R0)+,R1
004270  004737  004276                  CALL    PATPLUG
004274  000420                          BR      SUBAAA
004276                          PATPLUG:SUBTST  <<SUBR  PLUG IN NULL PATTERNS>>
                                ;*******************************************************************
                                ;*SUBTEST        SUBR    PLUG IN NULL PATTERNS
                                ;*******************************************************************
004276                                  FOR I := #1 TO #16.
004276  012737  000001  002130                                                  MOV #1,I
004304                                                                  B0:;;;;;;
004304  006001                            ROR   R1
004306                                    ON.NOERROR            ;IF CARRY CLEAR
004306  103402                                                                  BCS L1
004310  012713  021532                      MOV #MT0999,(R3)
004314                                    END ;OF ON.ERROR
004314                                                                  L1:;;;;;;
004314  062703  000002                    ADD #2,R3
004320                                  END ;OF FOR
004320  005237  002130                                                          INC I
004324  023727  002130  000020                                                  CMP I,#16.
004332  003764                                                                  BLE B0
004334                                                                  E0:;;;;;;;
004334  000207                          RETURN
004336                          SUBAAA: SUBTST  <<CLEAR THE CONFIGURATION TABLE>>
                                ;*******************************************************************
                                ;*SUBTEST        CLEAR THE CONFIGURATION TABLE
                                ;*******************************************************************
                                        .ENABLE LSB
```

```
                        IF RESTART EQ #0                                                TST RESTART
                                                                                        BNE L2
                          MOV   #CONFIG,R0
1$:                       CLR   (R0)+
                          CMP   #CONFIEND,R0
                          BNE   1$
                        END ;OF IF RESTART
                                                                        L2:::::::

                        .DSABL  LSB
                        MOV     #BIT4,CPUBIT                    ;SET ID BIT
                        SUBTST  <<SIZE FOR A HARDWARE SWITCH REGISTER>>
                        ;*************************************************************************
                        ;*SUBTEST       SIZE FOR A HARDWARE SWITCH REGISTER
                        ;*************************************************************************
                        ;;IF NOT FOUND OR IT IS
                        ;;EQUAL TO A "-1", SETUP FOR A SOFTWARE SWITCH REGISTER.
                        SET4    #3$
                        MOV     #3$,4                           ;SET NEW VECTOR
                        MOV     #DSWR,SWR       ;;SETUP FOR A HARDWARE SWITCH REGISTER
                        MOV     #DDISP,DISPLAY  ;;AND A HARDWARE DISPLAY REGISTER
                        CMP     #-1,@SWR        ;;TRY TO REFERENCE HARDWARE SWR
                        BNE     SUBAAB  ;BRANCH IF NO TRAP OCCURED &  THE HARDWARE SWR IS NOT = -1
                        BR      2$              ;;BRANCH IF NO TIMEOUT
3$:                     MOV     #2$,(SP)        ;;SET UP FOR TRAP RETURN
                        RTI
2$:                     RES4
                        MOV     #TIMEOUT,4                      ;RESTORE OLD (DEFAULT) VECTOR
                        CLR     CPUERR
                        MOV     #SWREG,SWR      ;;POINT TO SOFTWARE SWR
                        MOV     #DISPREG,DISPLAY
SUBAAB:                 SUBTST  <<SETUP ACT, APT, & XXDP>>
                        ;*************************************************************************
                        ;*SUBTEST       SETUP ACT, APT, & XXDP
                        ;*************************************************************************
                        CLR     $PASS                   ;CLEAR PASS COUNT
                        IFB #BIT5 SET.IN $ENVM
                                                                                        BITB #BIT5,$ENVM
                                                                                        BEQ L3
                          SET   $TPFLG                  ;INDICATE NO TERMINAL
                        MOV     #-1,$TPFLG
                        END ;OF IFB #BIT5
                                                                        L3:::::::
                        IFB #BIT7 SET.IN $ENVM
                                                                                        BITB #BIT7,$ENVM
                                                                                        BEQ L4
                          MOV   #$SWREG,SWR     ;USE APT SWR
                        END ;OF IFB #BIT7
                                                                        L4:::::::
                        IFB $ENV EQ #1
                                                                                        CMPB $ENV,#1
                                                                                        BNE L5
                          SET   APTFLAG
                        MOV     #-1,APTFLAG
                          SET   QVFLAG
                        MOV     #-1,QVFLAG
```

```
1362  004542                                    SET   $AUTO
 (1)  004542  012737  177777  001450           MOV     #-1,$AUTO
1363  004550                                    SET   QUICK
 (1)  004550  012737  177777  002122           MOV     #-1,QUICK
1364  004556  012737  037236  000024             MOV   #APTDOWN,PWRVEC
1365  004564  000430                             BR    SUBAAC
1366  004566                                    END ;OF IFB $ENV
 (3)  004566                                                          L5:::::::
1367  004566                                    IF 42 NE #STACK AND 42 NE #0
 (4)  004566  023727  000042  001400                                        CMP 42,#STACK
 (6)  004574  001424                                                       BEQ L6
 (6)  004576  005737  000042                                              TST 42
 (8)  004602  001421                                                       BEQ L6
1368  004604                                    SET   QVFLAG
 (1)  004604  012737  177777  002016           MOV     #-1,QVFLAG
1369  004612                                    SET   $AUTO
 (1)  004612  012737  177777  001450           MOV     #-1,$AUTO
1370  004620                                      IF 42 EQ #$ENDAD
 (4)  004620  023727  000042  011650                                        CMP 42,#$ENDAD
 (6)  004626  001004                                                       BNE L7
1371  004630                                        SET ACTFLAG
 (1)  004630  012737  177777  002020           MOV     #-1,ACTFLAG
1372  004636                                      ELSE
 (3)  004636  000403                                                       BR L10
 (3)  004640                                                          L7:::::::
1373  004640                                        SET XXDPCHAIN
 (1)  004640  012737  177777  002024           MOV     #-1,XXDPCHAIN
1374  004646                                      END ;OF IF 42
 (3)  004646                                                          L10:::::::
1375  004646                                    END ;OF IF 42
 (3)  004646                                                          L6:::::::
```

```
1378  004646                                SUBAAC: SUBTST  <<BEFORE FIRST LOOP CODE>>
 (3)                                      ;*****************************************************************************
 (3)                                      ;*SUBTEST       BEFORE FIRST LOOP CODE
 (2)                                      ;*****************************************************************************
1379  004646                                       SUBTST  <<PROTECT PROGRAM & LOADERS>>
 (3)                                      ;*****************************************************************************
 (3)                                      ;*SUBTEST       PROTECT PROGRAM & LOADERS
 (2)                                      ;*****************************************************************************
1380  004646  052737  100000  002230              BIS     #BIT15,CONFIG           ;PROTECT PROGRAM SPACE (BANK 0)
1381  004654  052737  100000  002236              BIS     #BIT15,CONFIG+6         ;PROTECT LOADER SPACE (BANK 1)
1382  004662  022737  011650  000042              CMP     #$ENDAD,42              ;;ACT-11?
1383  004670  001402                             BEQ     BEGIN                   ;;BRANCH IF YES
1384  004672                                      TYPE    MSG000                  ;TYPE PROGRAM TITLE
 (1)  004672  104401  066114                      TYPEIT  ,MSG000
1385  004676  000240                       BEGIN: NOP
1386  004700                                      SUBTST  <<DETERMINE MK11 ACCESS METHOD>>
 (3)                                      ;*****************************************************************************
 (3)                                      ;*SUBTEST       DETERMINE MK11 ACCESS METHOD
 (2)                                      ;*****************************************************************************
1387  004700                                      SET4    #5$
 (1)  004700  012737  004726  000004              MOV     #5$,4                   ;SET NEW VECTOR
1388  004706  012700  000010                      MOV     #8.,R0                  ;TEST FOR 8 CSR'S
1389  004712  012702  172100                      MOV     #MK11ADD,R2             ;R2 = CSR ADDRESS
1390  004716  005712                       4$:    TST     (R2)                    ;ANSWER R0 TRAP
1391  004720  005037  001632                      CLR     CSRREL                  ;ANSWER - CSR'S NOT RELOCATED
1392  004724  000410                             BR      6$                      ;BRANCH - OUT
1393  004726  062706  000004              5$:    ADD     #4,SP                   ;FIX STACK FROM TRAP
1394  004732  062702  000004                      ADD     #4,R2                   ;NEXT CSR = LAST CSR + 4
1395  004736  077011                             SOB     R0,4$                   ;LOOP TILL DONE
1396  004740                                      SET     CSRREL
 (1)  004740  012737  177777  001632              MOV     #-1,CSRREL
1397  004746                                6$:   RES4
 (1)  004746  012737  032466  000004              MOV     #TIMEOUT,4              ;RESTORE OLD (DEFAULT) VECTOR
 (1)  004754  005037  177766                      CLR     CPUERR
1398  004760                                      SUBTST  <<SETUP MAP & USER & SUPERVISOR STACK>>
 (3)                                      ;*****************************************************************************
 (3)                                      ;*SUBTEST       SETUP MAP & USER & SUPERVISOR STACK
 (2)                                      ;*****************************************************************************
1399  004760  104421                             DEENERGIZE
1400                                           ;SET PREVIOUS MODE TO SUPERVISOR
1401  004762  042737  030000  177776              BIC     #BIT13!BIT12,PSW
1402  004770  052737  010000  177776              BIS     #BIT12,PSW
1403  004776                                      PUSH    #SUPSTK
 (2)  004776  012746  000700                                                     MOV #SUPSTK,-(SP)
1404  005002  006606                             MTPI    SSP
1405
1406                                           ;SET PREVIOUS MODE TO USER
1407  005004  052737  030000  177776              BIS     #BIT13!BIT12,PSW
1408  005012                                      PUSH    #USESTK
 (2)  005012  012746  000600                                                     MOV #USESTK,-(SP)
1409  005016  006606                             MTPI    USP
1410
1411  005020                                      SUBTST  <<GET SOFTWARE SWITCH REGISTER IF NECESSARY>>
 (3)                                      ;*****************************************************************************
 (3)                                      ;*SUBTEST       GET SOFTWARE SWITCH REGISTER IF NECESSARY
 (2)                                      ;*****************************************************************************
```

```
1412  005020                                  IF $AUTO NE #0 THEN GOTO SUBAAD
 (5)  005020  005737  001450                                                                  TST $AUTO
 (5)  005024  001005                                                                          BNE SUBAAD
1413  005026  023727  002200  000176          CMP     SWR,#SWREG              ;;SOFTWARE SWITCH REG SELECTED?
1414  005034  001001                          BNE     SUBAAD                  ;BRANCH IF NO
1415  005036  104407                          GTSWR                           ;;GET SOFT-SWR SETTINGS
```

```
1417  005040  012737  000020  172516  SUBAAD: MOV     #20,MMR3                ;SET 22 BIT MODE
1418  005046  005037  177750                  CLR     MAINT                   ;CLEAR MARGINS
1419
1420  005052                          SUBAAP: SUBTST  <<SETUP UNIBUS MAP>>
 (3)                                  ;*************************************************************************
 (3)                                  ;*SUBTEST       SETUP UNIBUS MAP
 (2)                                  ;*************************************************************************
1421  005052  012737  000077  170372          MOV     #77,MAPH36
1422  005060  012737  160000  170370          MOV     #160000,MAPL36
1423  005066  005037  170202                  CLR     MAPHO
1424  005072  005037  170200                  CLR     MAPLO
1425  005076  004737  035524                  CALL    LOWMAP                  ;SETUP LOWER UNIBUS MAP
1426  005102  052737  000040  172516          BIS     #BIT5,MMR3              ;ENERGIZE UNIBUS MAP
1427
```

```
 1429  005110                                  SUBTST  <<GET MEMORY MANAGEMENT READY>>
  (3)                                ;*****************************************************************************
  (3)                                ;*SUBTEST       GET MEMCRY MANAGEMENT READY
  (2)                                ;*****************************************************************************
 1430  005110  104422                         KMAP
 1431  005112                                 MAP
  (3)  005112  010346                                                                    MOV R3,-(SP)
  (1)  005114  012703  000170                 MOV     #120.,R3
  (1)  005120  004737  034510                 CALL    MAPPER
  (3)  005124  012603                                                                    MOV (SP)+,R3
 1432  005126  104420                         ENERGIZE
 1433
 1434  005130                                  NEWTST  <<ANALIZE CONFIGURATION>>
  (3)                                ;*****************************************************************************
  (3)                                ;*TEST 1        ANALIZE CONFIGURATION
  (2)                                ;*****************************************************************************
  (2)  005130  000004                TST1:   SCOPE
 1435  005132  005037  001634                 CLR     CSRNO                          ;1ST CSR IS 0
 1436  005136                                 PUSH    CONTRL                         ;SAVE CACHE STATUS
  (2)  005136  013746  177746                                                            MOV CONTRL,-(SP)
 1437  005142                                 SET4    #CONFGT
  (1)  005142  012737  005472  000004        MOV     #CONFGT,4                              ;SET NEW VECTOR
 1438  005150  010637  005560                 MOV     SP,CONFSP                      ;SAVE STACK
 1439  005154  006137  001706          CONFG1: ROL     MKCSRS
 1440  005160  104426                         READCSR                                ;TRAP IF IT DOES NOT EXIST
 1441  005162  052737  000001  001706         BIS     #1,MKCSRS                      ;MARK CSR AS PRESENT
 1442  005170  005001                         CLR     R1
 1443  005172  013700  001530                 MOV     CSR+2,R0
 1444  005176  072027  177765                 ASH     #-11.,R0
 1445  005202  042700  177770                 BIC     #^C7,R0
 1446  005206                                 IF #BIT0 SET.IN R0 THEN LET MKCSRS := MKCSRS SET.BY #BIT8
  (4)  005206  032700  000001                                                            BIT #BIT0,R0
  (6)  005212  001403                                                                    BEQ L11
  (6)  005214  052737  000400  001706                                                    BIS #BIT8,MKCSRS
  (3)  005222                                                                    L11:;;;;;;
 1447  005222                                 CASE R0
  (2)  005222  010046                                                                    MOV R0,-(SP)
  (2)  005224  006316                                                                    ASL @SP
  (3)  005226  004737  005252                                                            JSR PC,L12
 1448  005232  005562                           INTER0
 1449  005234  005564                           INTER1
 1450  005236  005564                           INTER2
 1451  005240  005572                           INTER3
 1452  005242  005572                           INTER4
 1453  005244  005600                           INTER5
 1454  005246  005572                           INTER6
 1455  005250  005600                           INTER7
 1456  005252                                 END ;OF CASE R0
  (3)  005252                                                                    L12:;;;;;;
  (2)  005252  062616                                                                    ADD (SP)+,@SP
  (2)  005254  013646                                                                    MOV @(SP)+,-(SP)
  (2)  005256  004736                                                                    JSR PC,@(SP)+
 1457  005260  010137  001642                 MOV     R1,INTERL                      ;SET INTERLEAVE FACTOR
 1458  005264  004737  014472                 CALL    GETSIZE
 1459                                    ;THE NEXT 6 LINES SAVE THE READ STATUS OF THE BOX CAPACITY
 1460  005270  013701  001634                 MOV     CSRNO,R1
```

```
 1461  005274  006201                          ASR     R1
 1462  005276  013761  001526  001532          MOV     CSR,CSR0(R1)
 1463  005304  013761  001530  001552          MOV     CSR+2,CSR2(R1)
 1464  005312  013761  001636  001572          MOV     FIRSTB,CSRFB(R1)
 1465  005320  013761  001640  001612          MOV     LASTB,CSRLB(R1)
 1466  005326  013701  001636                  MOV     FIRSTB,R1
 1467  005332  070127  000006                  MUL     #6,R1           ;R1 = BANK * 6
 1468  005336  010137  001472                  MOV     R1,BANKINDEX
 1469  005342  013701  001472          CONFG3: MOV     BANKINDEX,R1    ;GET NEXT INDEX (SAME FIRST TIME)
 1470  005346  053761  001642  002230          BIS     INTERL,CONFIG(R1) ;SETUP INTERLEAVE BITS
 1471  005354  052761  020000  002230          BIS     #BIT13,CONFIG(R1) ;IDENTIFY AS MK11
 1472  005362  042761  040000  002230          BIC     #BIT14,CONFIG(R1)
 1473  005370  032737  030000  001530          BIT     #BIT13!BIT12,CSR+2 ;ANY EXTERNAL INTERLEAVE?
 1474  005376  001404                          BEQ     1$              ;NO - SKIP
 1475  005400  052761  000010  002230          BIS     #BIT3,CONFIG(R1) ;IDENTIFY IT
 1476  005406  000417                          BR      CONFG4
 1477  005410  013700  001634          1$:     MOV     CSRNO,R0        ;GET CSR LSB'S
 1478  005414  072027  177776                  ASH     #-2,R0          ;MAKE (0-7)
 1479  005420                                  PUSH    R0                                      MOV R0,-(SP)
   (2) 005420  010046
 1480  005422  072027  000010                  ASH     #8.,R0          ;MOVE TO BITS 10-8
 1481  005426  050061  002230                  BIS     R0,CONFIG(R1)   ;IDENTIFY AS CSR & BOX NUMBER
 1482  005432                                  POP     R0                                      MOV (SP)+,R0
   (2) 005432  012600
 1483  005434  012702  000001                  MOV     #1,R2
 1484  005440  072200                          ASH     R0,R2           ;R2 = 1,2,4,10,20,40,100, OR 200
 1485  005442  050261  002234                  BIS     R2,CONFIG+4(R1) ;SETUP CSR WORD
 1486  005446  062737  000006  001472  CONFG4: ADD     #6,BANKINDEX    ;NEXT TABLE ENTRY
 1487  005454  005237  001636                  INC     FIRSTB
 1488  005460  023737  001636  001640          CMP     FIRSTB,LASTB    ;DONE?
 1489  005466  101725                          BLOS    CONFG3          ;NO - LOOP
 1490  005470  000412                          BR      CONFG5
 1491  005472  013706  005560          CONFGT: MOV     CONFSP,SP       ;FIX STACK FROM MULTIPLE TRAP
 1492  005476                                  IF #BIT4 OFF.IN CPUERR                  BIT #BIT4,CPUERR
   (4) 005476  032737  000020  177766                                                  BNE L13
   (6) 005504  001004
 1493  005506                                    LET GOOD := #BIT4                     MOV #BIT4,GOOD
   (3) 005506  012737  000020  001432
 1494  005514  104037                            ERROR +37
 1495  005516                                  END ;OF IF CPUERR                               L13:;;;;;;
   (3) 005516
 1496  005516  062737  000004  001634  CONFG5: ADD     #4,CSRNO        ;BUMP CSR
 1497  005524  022737  000040  001634          CMP     #40,CSRNO       ;DONE?
 1498  005532  001402                          BEQ     1$              ;YES - SKIP
 1499  005534  000137  005154                  JMP     CONFG1          ;NO - LOOP
 1500  005540                          1$:     RES4
   (1) 005540  012737  032466  000004          MOV     #TIMEOUT,4      ;RESTORE OLD (DEFAULT) VECTOR
   (1) 005546  005037  177766                  CLR     CPUERR
 1501  005552                                  POP     CONTRL          ;RESTORE CACHE STATUS
   (2) 005552  012637  177746                                                          MOV (SP)+,CONTRL
 1502  005556  000413                          BR      SUBAAS
 1503  005560  000000                  CONFSP: 0                       ;STACK SAVED HERE!
 1504
 1505                                  ;TOTAL INTERLEAVE (0, 2-1, 4-1, OR 8-1)
 1506  005562  000207                  INTERO: RETURN                  ;NO INTERLEAVE
 1507
```

```
1508  005564                          INTER1:
1509  005564  012701  000001          INTER2:  MOV     #1,R1          ;2 WAY INTERLEAVE
1510  005570  000207                           RETURN
1511
1512  005572                          INTER3:
1513  005572                          INTER4:
1514  005572  012701  000002          INTER6:  MOV     #2,R1          ;4 WAY INTERLEAVE
1515  005576  000207                           RETURN
1516
1517  005600                          INTER5:
1518  005600  012701  000004          INTER7:  MOV     #4,R1          ;8 WAY INTERLEAVE
1519  005604  000207                           RETURN
```

```
005606  104472                  SUBAAS: ECCINIT
005610                                  NEWTST  <<TEST BANK 0 ACCESSES>>
                                ;*******************************************************************
                                ;*TEST 2          TEST BANK 0 ACCESSES
                                ;*******************************************************************
005610  000004                  TST2:   SCOPE
005612  005037  001460                  CLR     PARCNT                  ;CLEAR PARITY ERROR COUNTER
005616  012737  000001  001464          MOV     #1,NOPAR                ;SET THE NO PARITY ERROR FLAG
005624  005037  001456                  CLR     NEMCNT                  ;CLEAR NON-EXISTANT MEMORY ERROR COUNTER
005630  012737  000001  001466          MOV     #1,NONEM                ;SET THE NON-EXISTANT MEMORY ERROR MODE TO COUNT
005636                                  SET4    #NONEXIST
005636  012737  032422  000004          MOV     #NONEXIST,4                     ;SET NEW VECTOR
005644  005000                          CLR     R0
005646  012701  040000                  MOV     #SIZE,R1
005652  104424                          CACHOFF
005654  005720                  1$:     TST     (R0)+
005656  077102                          SOB     R1,1$
005660  104423                          CACHON
                                        ;SEE IF ANY FAILURES
005662  005737  001460                  TST     PARCNT                  ;ANY PARITY ERRORS?
005666  001403                          BEQ     2$                      ;NO - SKIP
005670                                  FATAL   3
005670  005237  001454                  INC     FATAL$                  ;SET FATAL INDICATOR
005674  104003                          ERROR   +3
005676  005737  001456          2$:     TST     NEMCNT                  ;ANY NON-EXISTANT MEMORY (HOLES)?
005702  001406                          BEQ     3$                      ;SKIP IF EQUAL
005704  162737  000002  001422          SUB     #2,ADDRESS              ;UPDATE 1ST ADDRESS FAILURE FROM AUTO INCREMENT
005712                                  FATAL   4
005712  005237  001454                  INC     FATAL$                  ;SET FATAL INDICATOR
005716  104004                          ERROR   +4
005720  053737  001476  002230  3$:     BIS     CPUBIT,CONFIG           ;SET CORRECT ACCESSED BIT ON BANK 0
005726                                  RES4
005726  012737  032466  000004          MOV     #TIMEOUT,4              ;RESTORE OLD (DEFAULT) VECTOR
005734  005037  177766                  CLR     CPUERR
```

D12
MAIN SKELETON MACY11 30A(1052) 28-FEB-78 09:20 PAGE 6-1
CEMKAA.MA1 24-FEB-78 11:01 T2 TEST BANK 0 ACCESSES SEQ 0146

```
 1545  005740                                  SUBTST  <<ENABLE MK11 FOR CORRECT TRAPS>>
  (3)                                  ;*****************************************************************************
  (3)                                  ;*SUBTEST        ENABLE MK11 FOR CORRECT TRAPS
  (2)                                  ;*****************************************************************************
 1546  005740                                  IF #SW0 SET.IN @SWR OR ACTFLAG NE #0
  (5)  005740  032777  000001  174232                                                          BIT #SW0,@SWR
  (5)  005746  001003                                                                          BNE L14
  (6)  005750  005737  002020                                                                  TST ACTFLAG
  (8)  005754  001402                                                                          BEQ L15
  (4)  005756                                                                          L14:;;;;;;
 1547  005756  104506                                ENASBE
 1548  005760                                  ELSE
  (3)  005760  000401                                                                          BR L16
  (3)  005762                                                                          L15:;;;;;;
 1549  005762  104472                                ECCINIT
 1550  005764                                  END ;OF IF #SW0
  (3)  005764                                                                          L16:;;;;;;
```

```
                                        NEWTST  <<TEST BANKS 1-167 (OCTAL) FOR ZEROS & ONES>>
                                ;*******************************************************************************
                                ;*TEST 3         TEST BANKS 1-167 (OCTAL) FOR ZEROS & ONES
                                ;*******************************************************************************
                                TST3:   SCOPE
                                        CLR     BANK
                                        MOV     #1,NOPAR                ;SET NO PARITY ERROR FLAG
                                        MOV     #2,NONEM                ;SET NON-EXISTANT MEMORY MODE TO EXIT TEST LOOP
                                        SET4    #NONEXIST
                                        MOV     #NONEXIST,4                     ;SET NEW VECTOR
                                        BMOV    MTP000                  ;PUT IN FAST MEMORY
                                            JSR R5,BLOCK1
                                            MTP000
                                        MOV     #005237,UIPAR6  ;PUT "INC " INSTRUCTION IN SUBROUTINE
                                        MOV     #PATERR,UIPAR7  ;PUT IN ADDRESS OF OPERAND FOR ABOVE
                                TAG9$:  INC     BANK
                                        CMP     #120.,BANK              ;DONE?
                                        BEQ     TAG2$                   ;YES - SKIP TO NEXT TEST
                                        MOV     BANK,R1
                                        MUL     #6,R1                   ;BANK*3*2
                                        MOV     R1,BANKINDEX
                                        CLR     PATERR                  ;CLEAR PATTERN ERROR COUNTER
                                        CLR     PARCNT                  ;CLEAR PARITY ERROR COUNTER
                                        CLR     NEMCNT                  ;CLEAR NON-EXISTANT MEMORY COUNTER (HOLES)
                                        MAP     BANK                    ;MAP A BANK
                                                                                MOV R3,-(SP)
                                        MOV     BANK,R3
                                        CALL    MAPPER
                                                                                MOV (SP)+,R3
                                        TST     CONFIG(R1)              ;IS THIS BANK PROTECTED?
                                        BMI     TSTBANK                 ;YES - GO TEST BANK SPECIAL
                                WARN1:  MOV     #207,UIPAR2             ;PUT "RETURN" INSTRUCTION AFTER WRITE ROUTINE
                                        MOV     #FIRST,R0
                                        MOV     R0,R4
                                        MOV     #SIZE,R1
                                        MOV     R1,R3
                                        CLR     R2                      ;DATA IS ZEROS
                                        CACHOFF
                                        SUPERVISOR
                                        BIS     #BIT14,PSW              ;DO IT TO IT!
                                        CALL    FASTCITY
                                        KERNEL
                                        CACHON
                                        BR      TAG3$                   ;SKIP NEXT INSTRUCTION
                                TAG2$:  CLR     BANK
                                        RES4
                                        MOV     #TIMEOUT,4              ;RESTORE OLD (DEFAULT) VECTOR
                                        CLR     CPUERR
                                        CLR     NOPAR                   ;INDICATE DEFAULT PARITY ACTION
                                        BR      SUBAAQ
                                TAG3$:  TST     NEMCNT                  ;ANY TRAPS?
                                        BEQ     1$                      ;NO - SKIP
                                        BR      TAG9$                   ;NOW - TRY NEXT BANK
                                1$:     CACHOFF
                                        SUPERVISOR
                                        BIS     #BIT14,PSW              ;DO IT TO IT!
```

```
006236  004737  177646                  CALL    UIPAR3          ;FINISH PATTERN
006242  104417                          KERNEL
006244  104423                          CACHON
006246  005737  001462                  TST     PATERR                  ;ANY PATTERN ERRORS
006252  001031                          BNE     2$                      ;YES - SKIP
006254  005737  001460                  TST     PARCNT                  ;ANY PARITY ERRORS
006260  001026                          BNE     2$                      ;YES - SKIP
006262  005737  001456                  TST     NEMCNT                  ;ANY NON EXISTANT MEMORY
006266  001023                          BNE     2$                      ;YES - SKIP
006270  012700  060000                  MOV     #FIRST,R0
006274  010004                          MOV     R0,R4
006276  012701  040000                  MOV     #SIZE,R1
006302  010103                          MOV     R1,R3
006304  013702  002154                  MOV     ONES,R2                 ;DATA IS ONES
006310  012737  000240  177644          MOV     #000240,UIPAR2          ;PUT "NOP" INSTRUCTION BACK IN SUBROUTINE
006316  104424                          CACHOFF
006320                                  SUPERVISOR
006320  052737  040000  177776          BIS     #BIT14,PSW              ;DO IT TO IT!
006326  004737  177640                  CALL    FASTCITY
006332  104417                          KERNEL
006334  104423                          CACHON
006336  013700  001472          2$:     MOV     BANKINDEX,R0
006342  005737  001462                  TST     PATERR                  ;ANY PATTERN ERRORS?
006346  001006                          BNE     3$                      ;YES - SKIP
006350  005737  001460                  TST     PARCNT                  ;ANY PARITY ERRORS?
006354  001003                          BNE     3$                      ;YES - SKIP
006356  005737  001456                  TST     NEMCNT                  ;ANY HOLES?
006362  001403                          BEQ     4$                      ;NONE - SKIP
006364  052760  004000  002230  3$:     BIS     #BIT11,CONFIG(R0)       ;SET ERROR BIT IN THIS BANK
006372  053760  001476  002230  4$:     BIS     CPUBIT,CONFIG(R0)       ;SET ACCESSED BIT
006400  000137  006036                  JMP     TAG9$

                                        ;TEST A PROTECTED BANK
006404                          TSTBANK:PUSH    R1
006404  010146                                                                  MOV R1,-(SP)
006406  012737  000001  001466          MOV     #1,NONEM                ;SET NON-EXISTANT MEMORY TO COUNT
006414  012700  060000                  MOV     #FIRST,R0
006420  012701  040000                  MOV     #SIZE,R1
006424  104424                          CACHOFF
006426                                  SUPERVISOR
006426  052737  040000  177776          BIS     #BIT14,PSW              ;DO IT TO IT!
006434  005720                  4$:     TST     (R0)+
006436  077102                          SOB     R1,4$
006440  104417                          KERNEL
006442  104423                          CACHON
006444  012737  000002  001466          MOV     #2,NONEM                ;RESET NON-EXISTANT MEMORY TO EXIT TEST LOOP
006452                                  POP     R1
006452  012601                                                                  MOV (SP)+,R1
006454                                  IF PARCNT NE #0
006454  005737  001460                                                          TST PARCNT
006460  001403                                                                  BEQ L17
006462  052761  004000  002230            BIS   #BIT11,CONFIG(R1)       ;ERROR BANK
006470                                  END ;OF IF PARCNT
006470                                                                  L17:::::::
006470                                  IF NEMCNT EQ #0
006470  005737  001456                                                          TST NEMCNT
```

G12

```
MAIN SKELETON   MACY11 30A(1052)  28-FEB-78  09:20  PAGE 6-4
CEMKAA.MA1    24-FEB-78 11:01          T3      TEST BANKS 1-167 (OCTAL) FOR ZEROS & ONES                                    SEQ 0149

  (6)  006474  001003                                                                                  BNE L20
 1641  006476  053761  001476  002230          BIS    CPUBIT,CONFIG(R1)        ;ACCESSED BANK
 1642  006504                                 END :OF IF NEMCNT
  (3)  006504                                                                                  L20:::::::
 1643  006504  000137  006036                 JMP     TAG9$
```

```
1645  006510                          SUBAAQ: SUBTST  ;;DETERMINE PROTECTED BANKS
 (3)                                  ;*****************************************************************************
 (3)                                  ;*SUBTEST       DETERMINE PROTECTED BANKS
 (3)                                  ;*****************************************************************************
1646  006510  000004                          SCOPE
1647  006512  104424                          CACHOFF
1648  006514  005037  001504                  CLR     KPFLAG
1649  006520  012701  060000                  MOV     #FIRST,R1
1650  006524  005004                          CLR     R4
1651  006526                                  FOR BANK := #0 TO #167
 (4)  006526  005037  001470                                                          CLR BANK
 (4)  006532                                                                  B1:::::::
1652  006532  013703  001470                    MOV   BANK,R3
1653  006536  070327  000006                    MUL   #6,R3
1654  006542  010337  001472                    MOV   R3,BANKINDEX
1655  006546                                    IF CPUBIT SET.IN CONFIG(R3) ;IF ACCESSABLE
 (4)  006546  033763  001476  002230                                                  BIT CPUBIT,CONFIG(R3)
 (6)  006554  001507                                                                 BEQ L21
1656  006556                                      IF #BIT13 SET.IN CONFIG(R3) AND #BIT14 OFF.IN CONFIG(R3) ;IF MK11
 (4)  006556  032763  020000  002230                                                 BIT #BIT13,CONFIG(R3)
 (6)  006564  001503                                                                 BEQ L22
 (6)  006566  032763  040000  002230                                                 BIT #BIT14,CONFIG(R3)
 (8)  006574  001077                                                                 BNE L22
1657  006576                                        MAP BANK
 (3)  006576  010346                                                                 MOV R3,-(SP)
 (1)  006600  013703  001470              MOV      BANK,R3
 (1)  006604  004737  034510              CALL     MAPPER
 (3)  006610  012603                                                                 MOV (SP)+,R3
1658  006612  005037  001526                    CLR       CSR
1659  006616                                    SUPERVISOR
 (1)  006616  052737  040000  177776          BIS     #BIT14,PSW              ;DO IT TO IT!
1660  006624  104504                          CHKDIS
1661  006626  010411                          MOV       R4,(R1)                       ;WRITE CHECKBITS (ALL ZEROS)
1662  006630  010461  000002                  MOV       R4,2(R1)
1663  006634  104502                          CLRCSR
1664  006636  005711                          TST       (R1)                          ;READ CHECKBITS INTO REAL CSR
1665  006640  104500                          WASDBE
1666                                  ;THIS MAKES SURE THAT SBE'S DON'T LOOK LIKE PROTECTED AREAS
1667  006642                          ON.NOERROR ;1
 (8)  006642  103441                                                                 BCS L23
1668  006644  005037  001526                    CLR     CSR
1669  006650  104504                            CHKDIS
1670  006652  013711  002154                    MOV     ONES,(R1)
1671  006656  013761  002154  000002            MOV     ONES,2(R1)
1672  006664  104502                            CLRCSR
1673  006666  005711                            TST     (R1)
1674  006670  104500                            WASDBE
1675  006672                                    ON.NOERROR ;2
 (8)  006672  103425                                                                 BCS L24
1676  006674  012737  007400  001526              MOV   #7400,CSR
1677  006702  104504                            CHKDIS
1678  006704  010411                            MOV   R4,(R1)
1679  006706  010461  000002                    MOV   R4,2(R1)
1680  006712  104502                            CLRCSR
1681  006714  005711                            TST   (R1)
1682  006716  104500                            WASDBE
```

```
006720                                          ON.NOERROR ;3
006720  103412                                                                  BCS L25
006722  012737  074000  001526                    MOV #74000,CSR
006730  104504                                    CHKDIS
006732  010411                                    MOV R4,(R1)
006734  010461  000002                            MOV R4,2(R1)
006740  104502                                    CLRCSR
006742  005711                                    TST (R1)
006744  104500                                    WASDBE
006746                                          END ;OF ON.NOERROR ;3
006746                                                                  L25:;;;;;;
006746                                        END ;OF ON.NOERROR ;2
006746                                                                  L24:;;;;;;
006746                                      END ;OF ON.NOERROR ;1
006746                                                                  L23:;;;;;;
006746                                      ON.NOERROR
006746  103405                                                                  BCS L26
                                              ;MARK PROTECTED BANK
006750  013703  001472                        MOV     BANKINDEX,R3
006754  052763  010000  002230                BIS     #BIT12,CONFIG(R3)     ;IDENTIFY AS PROTECTED BY MK11
006762                                      END ;OF ON.NOERROR
006762                                                                  L26:;;;;;;
006762  104470                                ECCDIS
006764  010411                                MOV     R4,(R1)               ;CLEAR OUT DOUBLE BIT ERROR!
006766  010461  000002                        MOV     R4,2(R1)                      ;CLEAR OUT DOUBLE BIT ERROR!
006772  104417                                KERNEL
006774                                    END ;OF IF
006774                                                                  L22:;;;;;;
006774                                  END ;OF IF
006774                                                                  L21:;;;;;;
006774                                END ;OF FOR
006774  005237  001470                                                          INC BANK
007000  023727  001470  000167                                                  CMP BANK,#167
007006  003651                                                                  BLE B1
007010                                                                  E1:;;;;;;;
007010                                MAP
007010  010346                                                                  MOV R3,-(SP)
007012  012703  000170                MOV     #120.,R3
007016  004737  034510                CALL    MAPPER
007022  012603                                                                  MOV (SP)+,R3
007024  005037  001470                CLR     BANK

007030                                IF #SWO SET.IN @SWR OR ACTFLAG NE #0
007030  032777  000001  173142                                                  BIT #SWO,@SWR
007036  001003                                                                  BNE L27
007040  005737  002020                                                          TST ACTFLAG
007044  001402                                                                  BEQ L30
007046                                                                  L27:;;;;;;
007046  104506                          ENASBE
007050                                ELSE
007050  000401                                                                  BR L31
007052                                                                  L30:;;;;;;
007052  104472                          ECCINIT
007054                                END ;OF IF #SWO
007054                                                                  L31:;;;;;;
```

```
1715  007054  104423                          CACHON
1716
1717  007056                                SUBTST  <<MOVE LOADERS IF NECESSARY>>
 (3)                                        ;*****************************************************************************
 (3)                                        ;*SUBTEST       MOVE LOADERS IF NECESSARY
 (2)                                        ;*****************************************************************************
1718  007056                                        IF CPUBIT EQ #BIT4                                      CMP CPUBIT,#BIT4
 (4)  007056  023727  001476  000020                                                                        BNE L32
 (6)  007064  001071
1719  007066  042737  100000  002236          BIC    #BIT15,CONFIG+6          ;UNPROTECT OLD LOADER SPACE
1720  007074                                  BEGIN FIRSTPROTECT
 (3)  007074                                                                                        B2:::::::
1721  007074                                    FOR BANK := #1 TO #167                                      MOV #1,BANK
 (4)  007074  012737  000001  001470                                                                B3:::::::
 (4)  007102
1722  007102  004737  036052                    CALL EXBANK
1723  007106                                    IF ACFLAG NE #0 AND BMFLAG EQ #0                            TST ACFLAG
 (4)  007106  005737  001506                                                                              BEQ L33
 (6)  007112  001412                                                                                      TST BMFLAG
 (6)  007114  005737  001520                                                                              BNE L33
 (8)  007120  001007
1724  007122                                      IF KPFLAG NE #0 OR MKFLAG EQ #0                           TST KPFLAG
 (5)  007122  005737  001504                                                                              BNE L34
 (5)  007126  001003                                                                                      TST MKFLAG
 (6)  007130  005737  001510                                                                              BNE L35
 (8)  007134  001001
 (4)  007136                                                                                        L34:::::::
1725  007136                                        LEAVE FIRSTPROTECT                                      BR E2
 (3)  007136  000413
1726  007140                                      END ;OF IF KPFLAG
 (3)  007140                                                                                        L35:::::::
1727  007140                                    END ;OF IF ACFLAG                                   L33:::::::
 (3)  007140
1728  007140                                  END ;OF FOR BANK                                            INC BANK
 (6)  007140  005237  001470                                                                              CMP BANK,#167
 (5)  007144  023727  001470  000167                                                                      BLE B3
 (5)  007152  003753                                                                                E3:::::::
 (5)  007154
1729  007154  012737  000001  001470            MOV #1,BANK
1730  007162  004737  036052                    CALL        EXBANK
1731  007166                                  END FIRSTPROTECT
 (3)  007166                                                                                        E2:::::::
1732  007166  013737  001470  002114          MOV    BANK,LOADHOME
1733  007174  013700  001472                  MOV    BANKINDEX,R0
1734  007200  052760  100000  002230          BIS    #BIT15,CONFIG(R0)
1735  007206                                  IF LOADHOME NE #1                                             CMP LOADHOME,#1
 (4)  007206  023727  002114  000001                                                                        BEQ L36
 (6)  007214  001415
1736  007216  012737  000004  001464            MOV #4,NOPAR                ;PARITY ACTION = COUNT & IGNORE
1737  007224  013700  002114                    MOV LOADHOME,R0             ;DESTINATION BANK
1738  007230  012701  000001                    MOV #1,R1                   ;SOURCE BANK
1739  007234  004737  035570                    CALL BANKMOV
1740  007240  005037  001464                    CLR NOPAR
1741  007244  005037  001460                    CLR PARCNT
1742  007250                                  END ;OF IF LOADHOME
 (3)  007250                                                                                        L36:::::::
```

```
 1743  007250                          END ;OF IF CPUBIT
   (3)  007250                                                  L32:::::::
```

```
1745  007250                          SET     STOPOK
 (1)  007250  012737  177777  002102  MOV     #-1,STOPOK
1746
1747  007256                          NEWTST  <<MK11 PROTECTION POINTER TEST>>
 (3)                          ;*****************************************************************************
 (3)                          ;*TEST 4        MK11 PROTECTION POINTER TEST
 (2)                          ;*****************************************************************************
 (2)  007256  000004          ST4:    SCOPE
1748                                  ;INSURE THAT BOTTOM OF MK11 WAS PROTECTED WITH PROTECTION POINTER CLEAR
1749  007260                          BEGIN PROTECTIONTEST
 (3)  007260                                                                  B4:::::::
1750  007260                          FOR BANK := #0 TO #167
 (4)  007260  005037  001470                                                      CLR BANK
 (4)  007264                                                                  B5:::::::
1751  007264  004737  036052            CALL EXBANK
1752  007270                            IF MKFLAG NE #0 AND ACFLAG NE #0
 (4)  007270  005737  001510                                                      TST MKFLAG
 (6)  007274  001410                                                              BEQ L37
 (6)  007276  005737  001506                                                      TST ACFLAG
 (8)  007302  001405                                                              BEQ L37
1753  007304                              IF KPFLAG EQ #0
 (4)  007304  005737  001504                                                      TST KPFLAG
 (6)  007310  001001                                                              BNE L40
1754  007312  104040                        ERROR     +40     ;MK11 PROTECTION POINTER FAILURE - DID NOT PROTECT BANK
1755  007314                              END ;OF IF KPFLAG
 (3)  007314                                                                  L40:::::::
1756  007314                              LEAVE PROTECTIONTEST
 (3)  007314  000406                                                              BR E4
1757  007316                            END ;OF IF MKFLAG
 (3)  007316                                                                  L37:::::::
1758  007316                          END ;OF FOR BANK
 (6)  007316  005237  001470                                                      INC BANK
 (5)  007322  023727  001470  000167                                              CMP BANK,#167
 (5)  007330  003755                                                              BLE B5
 (5)  007332                                                                  E5:::::::
1759  007332                          END PROTECTIONTEST
 (3)  007332                                                                  E4:::::::
```

```
                                        SUBTST  <<CHECK SYSTEM SIZE REGISTER>>
                                ;*******************************************************************************
                                ;*SUBTEST        CHECK SYSTEM SIZE REGISTER
                                ;*******************************************************************************
                                        CACHOFF
                                        CLR     MAPH36                  ;SET MAP TO BANK 1
                                        MOV     #100000,MAPL36
                                        SET4    #2$
                                        MOV     #2$,4                   ;SET NEW VECTOR
                                        MOV     #4,NOPAR                ;PARITY ACTION = COUNT & IGNORE
                                        MOV     #140000,R0              ;PAR6 SELECTED
                                1$:     TST     (R0)                    ;TRAP IF NON-EXISTANT
                                        ADD     #100000,MAPL36          ;BUMP 1 BANK
                                        ADC     MAPH36
                                        CMP     #73,MAPH36              ;INTO 128K SHADOW MEMORY?
                                        BNE     1$                      ;NO - LOOP
                                        BR      3$
                                2$:     ADD     #4,SP                   ;FIX STACK FROM TRAP
                                3$:     SUB     #100000,MAPL36
                                        SBC     MAPH36
                                        RES4
                                        MOV     #TIMEOUT,4              ;RESTORE OLD (DEFAULT) VECTOR
                                        CLR     CPUERR
                                        CLR     NOPAR                   ;RESTORE DEFAULT PARITY ACTION
                                        ROL     MAPL36
                                        ROL     MAPH36
                                        MOV     MAPH36,UNITOP
                                        MOV     #77,MAPH36              ;RESTORE MAP36
                                        MOV     #160000,MAPL36
                                        CACHON
                                        CLR     BANK
                                        SET     ACFLAG
                                        MOV     #-1,ACFLAG
                                        WHILE ACFLAG NE #0 OR MKFLAG NE #0
                                                                                B6:;;;;;;
                                                                                        TST ACFLAG
                                                                                        BNE L41
                                                                                        TST MKFLAG
                                                                                        BEQ L42
                                                                                L41:;;;;;;
                                          INC   BANK
                                          CALL  EXBANK
                                        END ;OF WHILE ACFLAG
                                                                                        BR B6
                                                                                L42:;;;;;;
                                                                                E6:;;;;;;
                                        MOV     BANK,R0
                                        DEC     R0
                                        MOV     R0,HIBANK
                                        IF R0 LT UNITOP THEN LET R0 := UNITOP
                                                                                        CMP R0,UNITOP
                                                                                        BGE L43
                                                                                        MOV UNITOP,R0
                                                                                L43:;;;;;;
                                        ASH     #9.,R0
                                        BIS     #777,R0                 ;R0 = LAST GOOD ADDRESS (LESS 6 LSB'S)
```

```
 1797  007600  020037  177760                  CMP     R0,SIZELO               ;IS IT RIGHT?
 1798  007604  001503                          BEQ     SUBAAT
 1799  007606  012737  177760  001422          MOV     #SIZELO,ADDRESS
 1800  007614  010037  001432                  MOV     R0,GOOD
 1801  007620  013737  177760  001440          MOV     SIZELO,BAD
 1802  007626  000300                          SWAB    R0
 1803  007630  004737  007670                  CALL    SYSSIZE
 1804  007634  010137  002060                  MOV     R1,GDSWITCH
 1805  007640  013700  177760                  MOV     SIZELO,R0
 1806  007644  000300                          SWAB    R0
 1807  007646  004737  007670                  CALL    SYSSIZE
 1808  007652  010137  002062                  MOV     R1,BDSWITCH
 1809  007656  104023                          ERROR   +23
 1810  007660  013737  001406  001410          MOV     $ERTTL,LASTERROR        ;DON'T COUNT THIS ERROR AS "DETECTED BUT NOT ISO
 1811  007666  000452                          BR      SUBAAT
 1812
 1813                                          ;DETERMINE SWITCH SETTINGS IN R1 FROM ADDRESS IN R0
 1814  007670  005001                  SYSSIZE:CLR     R1
 1815  007672                                  IF #BIT0 SET.IN R0 THEN LET R1 := R1 SET.BY #BIT2
  (4)  007672  032700  000001                                                  BIT #BIT0,R0
  (6)  007676  001402                                                          BEQ L44
  (6)  007700  052701  000004                                                  BIS #BIT2,R1
  (3)  007704                                                          L44:;;;;;;;
 1816  007704                                  IF #BIT1 SET.IN R0 THEN LET R1 := R1 SET.BY #BIT0
  (4)  007704  032700  000002                                                  BIT #BIT1,R0
  (6)  007710  001402                                                          BEQ L45
  (6)  007712  052701  000001                                                  BIS #BIT0,R1
  (3)  007716                                                          L45:;;;;;;;
 1817  007716                                  IF #BIT2 SET.IN R0 THEN LET R1 := R1 SET.BY #BIT4
  (4)  007716  032700  000004                                                  BIT #BIT2,R0
  (6)  007722  001402                                                          BEQ L46
  (6)  007724  052701  000020                                                  BIS #BIT4,R1
  (3)  007730                                                          L46:;;;;;;;
 1818  007730                                  IF #BIT3 SET.IN R0 THEN LET R1 := R1 SET.BY #BIT7
  (4)  007730  032700  000010                                                  BIT #BIT3,R0
  (6)  007734  001402                                                          BEQ L47
  (6)  007736  052701  000200                                                  BIS #BIT7,R1
  (3)  007742                                                          L47:;;;;;;;
 1819  007742                                  IF #BIT4 SET.IN R0 THEN LET R1 := R1 SET.BY #BIT3
  (4)  007742  032700  000020                                                  BIT #BIT4,R0
  (6)  007746  001402                                                          BEQ L50
  (6)  007750  052701  000010                                                  BIS #BIT3,R1
  (3)  007754                                                          L50:;;;;;;;
 1820  007754                                  IF #BIT5 SET.IN R0 THEN LET R1 := R1 SET.BY #BIT1
  (4)  007754  032700  000040                                                  BIT #BIT5,R0
  (6)  007760  001402                                                          BEQ L51
  (6)  007762  052701  000002                                                  BIS #BIT1,R1
  (3)  007766                                                          L51:;;;;;;;
 1821  007766                                  IF #BIT6 SET.IN R0 THEN LET R1 := R1 SET.BY #BIT5
  (4)  007766  032700  000100                                                  BIT #BIT6,R0
  (6)  007772  001402                                                          BEQ L52
  (6)  007774  052701  000040                                                  BIS #BIT5,R1
  (3)  010000                                                          L52:;;;;;;;
 1822  010000                                  IF #BIT7 SET.IN R0 THEN LET R1 := R1 SET.BY #BIT6
  (4)  010000  032700  000200                                                  BIT #BIT7,R0
  (6)  010004  001402                                                          BEQ L53
```

```
    (6)  010006  052701  000100          BIS #BIT6,R1
    (3)  010012                       L53:::::::
 1823    010012  000207                RETURN
```

```
 1913
 1914  010506                          SUBAAR: SUBTST  ;<LOAD CSR'S FROM E-TABLE $CDW1 & $CDW2>
  (3)                                  ;*****************************************************************
  (3)                                  ;*SUBTEST       LOAD CSR'S FROM E-TABLE $CDW1 & $CDW2
  (2)                                  ;*****************************************************************
 1915  010506                                IF $CDW1 NE #0 OR $CDW2 NE #0
  (5)  010506  005737  060244                                                          TST $CDW1
  (5)  010512  001003                                                                  BNE L63
  (6)  010514  005737  060246                                                          TST $CDW2
  (8)  010520  001430                                                                  BEQ L64
  (4)  010522                                                                    L63:::::::
 1916  010522  013701  001706            MOV   MKCSRS,R1
 1917  010526  013737  060244  001526    MOV   $CDW1,CSR
 1918  010534  013737  060246  001530    MOV   $CDW2,CSR+2
 1919  010542                            FOR CSRNO := #0 TO #34 BY #2
  (3)  010542  005037  001634                                                          CLR CSRNO
  (3)  010546                                                                    B14:::::::
 1920  010546  106301                      ASLB        R1
 1921  010550                            ON.ERROR
  (8)  010550  103001                                                                  BCC L65
 1922  010552  104425                        LOADCSR
 1923  010554                            END ;OF ON.ERROR
  (3)  010554                                                                    L65:::::::
 1924  010554                            END ;OF FOR CSRNO
  (6)  010554  062737  000002  001634                                                  ADD #2,CSRNO
  (5)  010562  023727  001634  000034                                                  CMP CSRNO,#34
  (5)  010570  003766                                                                  BLE B14
  (5)  010572                                                                    E14:::::::
 1925  010572  005237  060172            INC   $PASS
 1926  010576  000137  011720            JMP   APTHANG          ;GO LOOP FOREVER KEEPING APT ALIVE
 1927  010602                          END ;OF IF $CDW1
  (3)  010602                                                                    L64:::::::
```

```
 1929  010602                                  SUBTST  <<PRINT MEMORY CONFIGURATION MAP>>
  (3)                                   ;*****************************************************************************
  (3)                                   ;*SUBTEST        PRINT MEMORY CONFIGURATION MAP
  (2)                                   ;*****************************************************************************
 1930  010602  005037  002046                  CLR     MJSIZE
 1931  010606  005037  002050                  CLR     MKSIZE
 1932  010612                                  FOR R1 := #0 TO #167*3*2 BY #6
  (3)  010612  005001                                                                                  CLR R1
  (3)  010614                                                                                  B15:;;;;;;
 1933  010614                                    IF CPUBIT SET.IN CONFIG(R1)
  (4)  010614  033761  001476  002230                                                                  BIT CPUBIT,CONFIG(R1)
  (6)  010622  001420                                                                                  BEQ L66
 1934  010624                                      IF #BIT14!BIT13 OFF.IN CONFIG(R1) THEN LET MJSIZE := MJSIZE + #1
  (4)  010624  032761  060000  002230                                                                  BIT #BIT14!BIT13,CONFIG(
  (6)  010632  001002                                                                                  BNE L67
  (7)  010634  005237  002046                                                                          INC MJSIZE
  (3)  010640                                                                                  L67:;;;;;;
 1935  010640                                      IF #BIT14 OFF.IN CONFIG(R1) AND #BIT13 SET.IN CONFIG(R1)
  (4)  010640  032761  040000  002230                                                                  BIT #BIT14,CONFIG(R1)
  (6)  010646  001006                                                                                  BNE L70
  (6)  010650  032761  020000  002230                                                                  BIT #BIT13,CONFIG(R1)
  (8)  010656  001402                                                                                  BEQ L70
 1936  010660                                        LET MKSIZE := MKSIZE + #1
  (5)  010660  005237  002050                                                                          INC MKSIZE
 1937  010664                                      END ;OF IF MK11 MEMORY
  (3)  010664                                                                                  L70:;;;;;;
 1938  010664                                    END ;OF IF ACCESSABLE BY CPU
  (3)  010664                                                                                  L66:;;;;;;
 1939  010664                                  END ;OF FOR ALL BANKS IN TABLE
  (6)  010664  062701  000006                                                                          ADD #6,R1
  (5)  010670  020127  001312                                                                          CMP R1,#167*3*2
  (5)  010674  003747                                                                                  BLE B15
  (5)  010676                                                                                  E15:;;;;;;;
 1940  010676  006337  002046                  ASL     MJSIZE
 1941  010702  006337  002046                  ASL     MJSIZE
 1942  010706  006337  002046                  ASL     MJSIZE
 1943  010712  006337  002046                  ASL     MJSIZE                  ;MJSIZE := MJSIZE * 16.
 1944  010716  006337  002050                  ASL     MKSIZE
 1945  010722  006337  002050                  ASL     MKSIZE
 1946  010726  006337  002050                  ASL     MKSIZE
 1947  010732  006337  002050                  ASL     MKSIZE                  ;MKSIZE := MKSIZE * 16.
 1948  010736  005237  002100                  INC     UNITOP
 1949  010742  006337  002100                  ASL     UNITOP
 1950  010746  006337  002100                  ASL     UNITOP
 1951  010752  006337  002100                  ASL     UNITOP
 1952  010756  006337  002100                  ASL     UNITOP          ;UNITOP := UNITOP * 16.
 1953  010762                                  LET I := MJSIZE + MKSIZE
  (3)  010762  013737  002046  002130                                                                  MOV MJSIZE,I
  (5)  010770  063737  002050  002130                                                                  ADD MKSIZE,I
 1954  010776                                  IF I LT UNITOP THEN LET I := UNITOP
  (4)  010776  023737  002130  002100                                                                  CMP I,UNITOP
  (6)  011004  002003                                                                                  BGE L71
  (5)  011006  013737  002100  002130                                                                  MOV UNITOP,I
  (3)  011014                                                                                  L71:;;;;;;
 1955  011014                                  TYPE    $CRLF
  (1)  011014  104401  002222                  TYPEIT  ,$CRLF
```

E13


```
 1956  011020                                          TYPDEC  I
  (1)  011020  013746  002130                         MOV     I,-(SP) ;;SAVE I FOR TYPEOUT
  (1)  011024  104405                                 TYPDS                   ;;GO TYPE--DECIMAL ASCII WITH SIGN
 1957  011026                                          TYPE    MSG070
  (1)  011026  104401  071557                         TYPEIT  ,MSG070
 1958  011032                                          TYPDEC  MJSIZE
  (1)  011032  013746  002046                         MOV     MJSIZE,-(SP)    ;;SAVE MJSIZE FOR TYPEOUT
  (1)  011036  104405                                 TYPDS                   ;;GO TYPE--DECIMAL ASCII WITH SIGN
 1959  011040                                          TYPE    MSG071
  (1)  011040  104401  071602                         TYPEIT  ,MSG071
 1960  011044                                          TYPDEC  MKSIZE
  (1)  011044  013746  002050                         MOV     MKSIZE,-(SP)    ;;SAVE MKSIZE FOR TYPEOUT
  (1)  011050  104405                                 TYPDS                   ;;GO TYPE--DECIMAL ASCII WITH SIGN
 1961  011052                                          TYPE    MSG072
  (1)  011052  104401  071615                         TYPEIT  ,MSG072
 1962  011056                                          IF #SW6 SET.IN @SWR THEN $CALL PCONFIG
  (4)  011056  032777  000100  171114                                                    BIT #SW6,@SWR
  (6)  011064  001402                                                                    BEQ L72
  (4)  011066  004737  031162                                                            JSR PC,PCONFIG
  (3)  011072                                                                    L72:::::::
```

F13


```
 1964  011072                                  SUBTST  <<CHECK APT SIZING>>
  (3)                                  ;*****************************************************************************
  (3)                                  ;*SUBTEST        CHECK APT SIZING
  (2)                                  ;*****************************************************************************
 1965  011072                                  IF APTFLAG NE #0
  (4)  011072  005737  002022                                                                  TST APTFLAG
  (6)  011076  001466                                                                          BEQ L73
 1966  011100  005037  002120                    CLR    TEMP
 1967  011104  012700  060214                    MOV    #$MAMS1,R0
 1968  011110                                    FOR R2 := #0 TO #4
  (4)  011110  005002                                                                          CLR R2
  (4)  011112                                                                          B16:;;;;;;
 1969  011112                                      IFB 1(R0) NE #0
  (5)  011112  105760  000001                                                                  TSTB 1(R0)
  (7)  011116  001441                                                                          BEQ L74
 1970  011120  111001                                MOVB       (R0),R1
 1971  011122  042701  177400                        BIC        #177400,R1
 1972  011126                                        IF 2(R0) LT #0
  (4)  011126  005760  000002                                                                  TST 2(R0)
  (6)  011132  002002                                                                          BGE L75
 1973  011134  000261                                  SEC
 1974  011136                                        ELSE
  (3)  011136  000401                                                                          BR L76
  (3)  011140                                                                          L75:;;;;;;
 1975  011140  000241                                  CLC
 1976  011142                                        END ;OF IF 2(R0)
  (3)  011142                                                                          L76:;;;;;;
 1977  011142  006101                                ROL        R1
 1978  011144  005201                                INC        R1                ;TO COMPENSATE FOR 4 BANKS BEING (0-3)
 1979  011146  006301                                ASL        R1
 1980  011150  006301                                ASL        R1
 1981  011152  006301                                ASL        R1
 1982  011154  006301                                ASL        R1
 1983  011156  163701  002120                        SUB        TEMP,R1
 1984  011162  010137  002120                        MOV        R1,TEMP
 1985  011166                                        IFB 1(R0) EQ #1
  (5)  011166  126027  000001  000001                                                          CMPB 1(R0),#1
  (7)  011174  001002                                                                          BNE L77
 1986  011176  060137  002104                          ADD      R1,APTCORE
 1987  011202                                        END ;OF IFB 1(R0)
  (3)  011202                                                                          L77:;;;;;;
 1988  011202                                        IFB 1(R0) EQ #4
  (5)  011202  126027  000001  000004                                                          CMPB 1(R0),#4
  (7)  011210  001002                                                                          BNE L100
 1989  011212  060137  002106                          ADD      R1,APTMOS
 1990  011216                                        END ;OF IFB 1(R0)
  (3)  011216                                                                          L100:;;;;;;
 1991  011216  062700  000004                        ADD        #4,R0
 1992  011222                                      END ;OF IFB 1(R0)
  (3)  011222                                                                          L74:;;;;;;
 1993  011222                                    END ;OF FOR R2
  (6)  011222  005202                                                                          INC R2
  (5)  011224  020227  000004                                                                  CMP R2,#4
  (5)  011230  003730                                                                          BLE B16
  (5)  011232                                                                          E16:;;;;;;;
 1994  011232                                    IF APTCORE NE MJSIZE OR APTMOS NE MKSIZE
```

G13

MAIN SKELETON MACY11 30A(1052) 28-FEB-78 09:20 PAGE 6-17
CEMKAA.MA1 24-FEB-78 11:01 CHECK APT SIZING SEQ 0162

```
  (5)  011232  023737  002104  002046                                  CMP APTCORE,MJSIZE
  (5)  011240  001004                                                  BNE L101
  (6)  011242  023737  002106  002050                                  CMP APTMOS,MKSIZE
  (8)  011250  001401                                                  BEQ L102
  (4)  011252                                                     L101:;;;;;;
 1995  011252  104051                          ERROR        +51
 1996  011254                                END ;OF IF APTCORE
  (3)  011254                                                     L102:;;;;;;
 1997  011254                              END ;OF IF APTFLAG
  (3)  011254                                                     L73:;;;;;;
```

```
 1999  011254                          LOOP:   NEWTST  <<DIAGNOSTIC MODE DISPATCH ROUTINE>>
  (3)                                  ;*****************************************************************************
  (3)                                  ;*TEST 5        DIAGNOSTIC MODE DISPATCH ROUTINE
  (2)                                  ;*****************************************************************************
  (2)   011254  000004                 TSTS:   SCOPE
 2000   011256  005037  001704                  CLR     CONTFLAG
 2001   011262  017700  170712                  MOV     @SWR,R0         ;GET SWITCHES
 2002   011266  042700  177761                  BIC     #^C16,R0        ;MASK TO ONLY MODE BITS
 2003   011272  004770  011302                  CALL    @DISPTBL(R0)    ;DISPATCH TO ROUTINE THROUGH NEXT TABLE
 2004   011276  000137  011322                  JMP     MEMDONE         ;GO TO NEXT TEST
 2005   011302  011754                 DISPTBL:BAFPAF  ;MODE 0;BANKS FORWARD, PATTERNS FORWARD
 2006   011304  012070                         BAFPAR  ;MODE 1;BANKS FORWARD, PATTERNS REVERSE
 2007   011306  012204                         BAWPAF  ;MODE 2;BANKS WORST FIRST, PATTERNS FORWARD
 2008   011310  012342                         BAWPAR  ;MODE 3;BANKS WORST FIRST, PATTERNS REVERSE
 2009   011312  012500                         PAFBAF  ;MODE 4;PATTERNS FORWARD, BANKS FORWARD
 2010   011314  012636                         PAFBAW  ;MODE 5;PATTERNS FORWARD, BANKS WORST FIRST
 2011   011316  013016                         PARBAF  ;MODE 6;PATTERNS REVERSE, BANKS FORWARD
 2012   011320  013154                         PARBAW  ;MODE 7;PATTERNS REVERSE, BANKS WORST FIRST
 2013                                         ;**NOTE** MARGINS FOLLOW PATTERNS (FORWARD IS (0,2,3,4,5))
 2014                                         ;(REVERSE IS (5,4,3,2,0))
```

```
                                MEMDONE:NEWTST<<UNIQUE BANK TEST>>
                                ;*****************************************************************************
                                ;*TEST 6         UNIQUE BANK TEST
                                ;*****************************************************************************
                                TST6:   SCOPE
                                        SET     HEADER
                                        MOV     #-1,HEADER
                                        SET     MUT
                                        MOV     #-1,MUT
                                        CALL    MT0027
                                        SET     HEADER
                                        MOV     #-1,HEADER
                                        CLR     MUT

                                        NEWTST  <<FORCE ADDRESS PARITY ERRORS>>
                                ;*****************************************************************************
                                ;*TEST 7         FORCE ADDRESS PARITY ERRORS
                                ;*****************************************************************************
                                TST7:   SCOPE
                                        MOV     #FIRST,R0
                                        MOV     #4,NOPAR                ;INDICATE PARITY ACTION
                                        BMOV    ADDPAR
                                            JSR R5,BLOCK1
                                            ADDPAR
                                        FOR BANK := #0 TO #167
                                                                                        CLR BANK
                                                                                B17:::::::
                                          CALL EXBANK
                                          IF ACFLAG NE #0
                                                                                        TST ACFLAG
                                                                                        BEQ L103
                                            CLR PARCNT
                                            CALL SUPD01
                                            IF PARCNT NE #1
                                                                                        CMP PARCNT,#1
                                                                                        BEQ L104
                                              ERROR +35
                                            END ;OF IF PARCNT
                                                                                L104:::::::
                                          END ;OF IF ACFLAG
                                                                                L103:::::::
                                        END ;OF FOR BANK
                                                                                        INC BANK
                                                                                        CMP BANK,#167
                                                                                        BLE B17
                                                                                E17:::::::
                                        CLR     NOPAR                   ;INDICATE PARITY ACTION
                                        BR      SUBAAU

                                        ;THESE 3 INSTRUCTIONS ARE MOVED TO THE USER PAR'S IN ORDER TO
                                        ;CAUSE ABORTS FROM THE BANK UNDER TEST VICE THE PROGRAM SPACE
                                ADDPAR: MOV     #BIT1,MAINT             ;V177640        ;CREATE ADDRESS PARITY ERROR
                                        TST     (R0)                    ;V177646        ;THIS SHOULD ABORT & INCREMENT P
                                        RETURN                          ;V177650
                                FLUSH:  SUBTST  <<FLUSH OUT DBE'S>>
                                ;*****************************************************************************
```

```
MAIN SKELETON   MACY11 30A(1052)  28-FEB-78  09:20  PAGE 8-1                                                    SEQ 0165
CEMKAA.MAI     24-FEB-78 11:01          FLUSH OUT DBE'S

    (3)                                 ;*SUBTEST       FLUSH OUT DBE'S
    (2)                                 ;*****************************************************************************
  2057  011526  004737  021206                  CALL    MT0030
```

```
                                        .SBTTL  END OF PASS ROUTINE
                                ;*******************************************************************************
                                ;*INCREMENT THE PASS NUMBER ($PASS)
                                ;*INDICATE END-OF-PROGRAM AFTER EACH PASSES THRU THE PROGRAM
                                ;*TYPE "END PASS #XXXXX" (WHERE XXXXX IS A DECIMAL NUMBER)
                                ;*IF THERES A MONITOR GO TO IT
                                ;*IF THERE ISN'T JUMP TO LOOP
011532  005037  001412          $EOP:   CLR     SBEFLAG
011536  005037  002126                  CLR     FSINFLAG
011542  013737  001406  001410          MOV     $ERTTL,LASTERROR
011550  005237  060172                  INC     $PASS           ;;INCREMENT THE PASS NUMBER
011554  042737  100000  060172          BIC     #100000,$PASS   ;;DON'T ALLOW A NEG. NUMBER
011562                                  TYPE    MSG077          ;;TYPE "END PASS #"
 (1)    011562  104401  071723          TYPEIT  ,MSG077
011566                                  IF #SW11 SET.IN @SWR OR QVFLAG NE #0
 (5)    011566  032777  004000  170404                                          BIT #SW11,@SWR
 (5)    011574  001003                                                          BNE L110
 (6)    011576  005737  002016                                                  TST QVFLAG
 (8)    011602  001404                                                          BEQ L111
 (4)    011604                                                          L110:;;;;;;
011604                                    TYPE  MSG035          ;QV
 (1)    011604  104401  070265          TYPEIT  ,MSG035
011610  005037  002016                    CLR   QVFLAG
011614                                  END ;OF IF SW11
 (3)    011614                                                          L111:;;;;;;
011614                                  TYPDEC  $PASS
 (1)    011614  013746  060172          MOV     $PASS,-(SP)     ;;SAVE $PASS FOR TYPEOUT
 (1)    011620  104405                  TYPDS                   ;;GO TYPE--DECIMAL ASCII WITH SIGN
011622  013700  000042                  MOV     42,R0           ;;GET MONITOR ADDRESS
011626  001430                          BEQ     $DOAGAIN        ;;BRANCH IF NO MONITOR
011630  022700  001400          $ZAP42: CMP     #STACK,R0       ;ARE WE UNDER RT11
011634  001425                          BEQ     $DOAGAIN        ;YES - BRANCH
                                        ;WE ARE UNDER (HEAVEN HELP US) XXDP!
011636                                  PUSH    R0
 (2)    011636  010046                                                  MOV R0,-(SP)
011640  004737  037156                  CALL    SHUTUP
011644                                  POP     R0
 (2)    011644  012600                                                  MOV (SP)+,R0
011646  000005                          RESET                   ;;CLEAR THE WORLD
011650  004710                  $ENDAD: CALL    (R0)            ;;GO TO MONITOR
011652  000240                          NOP                     ;;SAVE ROOM
011654  000240                          NOP                     ;;FOR
011656  000240                          NOP                     ;;ACT11
011660                          $DOAGN: ;UNDO SHUTUP STUFF
                                        ;       RESTORE STACK
                                        ;       ENERGIZE UNIBUS MAP & 22 BIT ADDRESSING
                                        ;       ENERGIZE MEMORY MANAGEMENT
                                        ;       PUT LOADERS BACK HOME
011660  012706  001400                  MOV     #STACK,SP
011664  052737  000060  172516          BIS     #BIT5!BIT4,MMR3
011672  104420                          ENERGIZE
011674  013700  002114                  MOV     LOADHOME,R0             ;DESTINATION BANK
011700  012701  000001                  MOV     #1,R1                   ;SOURCE BANK
011704  004737  035570                  CALL    BANKMOV
011710                          $DOAGAIN:IF $USWR EQ $PASS      ;IF APT LIMITING # OF PASSES
 (4)    011710  023737  060210  060172                                  CMP $USWR,$PASS
```

```
  (6)  011716  001014                                                                  BNE L112
 2103  011720  012701  000050          APTHANG:  MOV    #50,R1
 2104  011724  077001                  2$:       SOB    R0,2$
 2105  011726  062737  000001  060174            ADD    #1,$DEVCT
 2106  011734  005537  060176                    ADC    $UNIT
 2107  011740  077107                            SOB    R1,2$
 2108  011742  005237  060172                    INC    $PASS
 2109  011746  000764                            BR     APTHANG
 2110  011750                                  END ;OF IF $USWR
  (3)  011750                                                                          L112:::::::
 2111  011750  000137  011254                  JMP     LOOP            ;RETURN
```

```
                                        .TITLE  MTEST MODES

                                BAFPAF: SUBTST  <<BANKS FORWARD,PATTERNS FORWARD      **RECURSIVE**>>
                                ;*********************************************************************
                                ;*SUBTEST        BANKS FORWARD,PATTERNS FORWARD      **RECURSIVE**
                                ;*********************************************************************
011754  005037  001470                  CLR     BANK            ;SET BANK TO 0
                                        ;START OF BANK LOOP
011760  004737  036052          1$:     CALL    EXBANK          ;EXAMINE BANK
011764  005737  001506                  TST     ACFLAG          ;CAN WE ACCESS THIS BANK?
011770  001417                          BEQ     4$              ;NO - GO TO BANK LOOP TERMINATION
011772  005737  001514                  TST     RRFLAG          ;RELOCATION REQUIRED?
011776  001014                          BNE     4$              ;YES - GO TO BANK LOOP TERMINATION
012000  005037  001502                  CLR     PATTERN         ;SET PATTERN TO 0
                                        ;START OF PATTERN LOOP
012004  005037  001474          2$:     CLR     MARGIN          ;SET MARGIN TO 0
                                        ;START OF MARGIN LOOP
012010  004737  013334          3$:     CALL    MTEST           ;GO TEST CORRECT MEMORY
                                        ;TERMINATION OF MARGIN LOOP
012014  004737  036412                  CALL    INCMAR          ;NEXT HIGHER MARGIN
012020  001373                          BNE     3$              ;IF NOT DONE - LOOP ON THIS MARGIN
                                        ;TERMINATION OF PATTERN LOOP
012022  004737  036352                  CALL    INCPAT          ;GO SEE IF THIS IS THE LAST PATTERN
012026  001366                          BNE     2$              ;NO - LOOP ON THIS PATTERN
                                        ;TERMINATION OF BANK LOOP
012030  004737  036376          4$:     CALL    INCBNK          ;NEXT HIGHER BANK
012034  001351                          BNE     1$              ;IF NOT DONE - LOOP ON THIS BANK
                                        ;END OF LOOPS
012036  005737  001516                  TST     RLFLAG          ;HAVE WE BEEN RELOCATED?
012042  001401                          BEQ     5$              ;NO - SKIP
012044  000207                          RETURN                  ;YES - RETURN
012046  004737  034720          5$:     CALL    RELOCATE        ;MOVE & MAP PROGRAM
012052                                  ON.ERROR THEN $RETURN
012052  103001                                                                  BCC L113
012054  000207                                                                  RTS PC
012056                                                                  L113:;;;;;;
                                        ;**NOTE** RECURSIVE CALL
012056  004737  011754                  CALL    BAFPAF          ;CALL SELF
012062  004737  035340                  CALL    UNRELOCATE      ;UNMOVE & UNMAP PROGRAM
012066  000207                          RETURN
```

```
                                BAFPAR: SUBTST  <<BANKS FORWARD,PATTERNS REVERSE      **RECURSIVE**>>
                                ;*********************************************************************
                                ;*SUBTEST       BANKS FORWARD,PATTERNS REVERSE      **RECURSIVE**
                                ;*********************************************************************
2150  012070  005037  001470          CLR     BANK            ;SET BANK TO 0
2151                                  ;START OF BANK LOOP
2152  012074  004737  036052    1$:   CALL    EXBNK           ;EXAMINE BANK
2153  012100  005737  001506          TST     ACFLAG          ;CAN WE ACCESS THIS BANK?
2154  012104  001417                  BEQ     4$              ;NO - GO TO BANK LOOP TERMINATION
2155  012106  005737  001514          TST     RRFLAG          ;RELOCATION REQUIRED?
2156  012112  001014                  BNE     4$              ;YES - GO TO BANK LOOP TERMINATION
2157  012114  004737  036366          CALL    SETPAT          ;SET HIGH PATTERN FOR CORRECT MEMORY
2158                                  ;START OF PATTERN LOOP
2159  012120  004737  036616    2$:   CALL    SETMAR          ;SET HIGH MARGINS (IF NOT INHIBITED)
2160                                  ;START OF MARGIN LOOP
2161  012124  004737  013334    3$:   CALL    MTEST           ;GO TEST CORRECT MEMORY
2162                                  ;TERMINATION OF MARGIN LOOP
2163  012130  004737  036556          CALL    DECMAR          ;NEXT LOWER MARGIN
2164  012134  100373                  BPL     3$              ;IF NOT DONE - LOOP ON THIS MARGIN
2165                                  ;TERMINATION OF PATTERN LOOP
2166  012136  005337  001502          DEC     PATTERN         ;IS THIS THE LAST PATTERN?
2167  012142  100366                  BPL     2$              ;NO - LOOP ON THIS PATTERN
2168                                  ;TERMINATION OF BANK LOOP
2169  012144  004737  036376    4$:   CALL    INCBNK          ;NEXT HIGHER BANK
2170  012150  001351                  BNE     1$              ;IF NOT DONE - LOOP ON THIS BANK
2171                                  ;END OF LOOPS
2172  012152  005737  001516          TST     RLFLAG          ;HAVE WE BEEN RELOCATED?
2173  012156  001401                  BEQ     5$              ;NO - SKIP
2174  012160  000207                  RETURN                  ;YES - RETURN
2175  012162  004737  034720    5$:   CALL    RELOCATE        ;MOVE & MAP PROGRAM
2176  012166                        ON.ERROR THEN $RETURN
  (8) 012166  103001                                                          BCC L114
  (5) 012170  000207                                                          RTS PC
  (4) 012172                                                        L114:;;;;;;
2177                                  ;**NOTE** RECURSIVE CALL
2178  012172  004737  012070          CALL    BAFPAR          ;CALL SELF
2179  012176  004737  035340          CALL    UNRELOCATE      ;UNMOVE & UNMAP PROGRAM
2180  012202  000207                  RETURN
```

```
                                BAWPAF: SUBTST  <<BANKS WORST FIRST,PATTERNS FORWARD  **RECURSIVE**>
                                ;*****************************************************************************
                                ;*SUBTEST       BANKS WORST FIRST,PATTERNS FORWARD  **RECURSIVE**
                                ;*****************************************************************************
2183  012204  005037  001470          CLR     BANK                    ;SET BANK TO 0
2184                                  ;START OF BANK LOOP
2185  012210  004737  036052    1$:    CALL    EXBNK                   ;EXAMINE BANK
2186  012214  005737  001506          TST     ACFLAG                  ;CAN WE ACCESS THIS BANK?
2187  012220  001422                  BEQ     4$                      ;NO - GO TO BANK LOOP TERMINATION
2188  012222  005737  001520          TST     BMFLAG                  ;IS THIS BAD MEMORY (WORST FIRST)?
2189  012226  001417                  BEQ     4$                      ;NO - GO TO BANK LOOP TERMINATION
2190  012230  005737  001514          TST     RRFLAG                  ;RELOCATION REQUIRED?
2191  012234  001014                  BNE     4$                      ;YES - GO TO BANK LOOP TERMINATION
2192  012236  005037  001502          CLR     PATTERN                 ;SET PATTERN TO 0
2193                                  ;START OF PATTERN LOOP
2194  012242  005037  001474    2$:    CLR     MARGIN                  ;SET MARGIN TO 0
2195                                  ;START OF MARGIN LOOP
2196  012246  004737  013334    3$:    CALL    MTEST                   ;GO TEST CORRECT MEMORY
2197                                  ;TERMINATION OF MARGIN LOOP
2198  012252  004737  036412          CALL    INCMAR                  ;NEXT HIGHER MARGIN
2199  012256  001373                  BNE     3$                      ;IF NOT DONE - LOOP ON THIS MARGIN
2200                                  ;TERMINATION OF PATTERN LOOP
2201  012260  004737  036352          CALL    INCPAT                  ;GO SEE IF THIS IS THE LAST PATTERN
2202  012264  001366                  BNE     2$                      ;NO - LOOP ON THIS PATTERN
2203                                  ;TERMINATION OF BANK LOOP
2204  012266  004737  036376    4$:    CALL    INCBNK                  ;NEXT HIGHER BANK
2205  012272  001346                  BNE     1$                      ;IF NOT DONE - LOOP ON THIS BANK
2206                                  ;END OF LOOPS
2207  012274  005137  002140          COM     WORST                   ;IS THIS AN EVEN NUMBERED PASS?
2208  012300  001003                  BNE     5$                      ;YES - SKIP
2209                                  ;**NOTE** RECURSIVE CALL
2210  012302  004737  012204          CALL    BAWPAF                  ;CALL SELF
2211  012306  000207                  RETURN
2212  012310  005737  001516    5$:    TST     RLFLAG                  ;HAVE WE BEEN RELOCATED?
2213  012314  001401                  BEQ     6$                      ;NO - SKIP
2214  012316  000207                  RETURN                          ;YES - RETURN
2215  012320  004737  034720    6$:    CALL    RELOCATE                ;MOVE & MAP PROGRAM
2216  012324                          ON.ERROR THEN $RETURN
  (8) 012324  103001                                                  BCC L115
  (5) 012326  000207                                                  RTS PC
  (4) 012330                                                        L115:::::::
2217                                  ;**NOTE** RECURSIVE CALL
2218  012330  004737  012204          CALL    BAWPAF                  ;CALL SELF
2219  012334  004737  035340          CALL    UNRELOCATE              ;UNMOVE & UNMAP PROGRAM
2220  012340  000207                  RETURN
```

```
 2222  012342                          BAWPAR: SUBTST  <<BANKS WORST FIRST,PATTERNS REVERSE  **RECURSIVE**>>
   (3)                               ;*****************************************************************************
   (3)                               ;*SUBTEST       BANKS WORST FIRST,PATTERNS REVERSE  **RECURSIVE**
   (2)                               ;*****************************************************************************
 2223  012342  005037  001470               CLR     BANK            ;SET BANK TO 0
 2224                                       ;START OF BANK LOOP
 2225  012346  004737  036052         1$:   CALL    EXBNK           ;EXAMINE BANK
 2226  012352  005737  001506               TST     ACFLAG          ;CAN WE ACCESS THIS BANK?
 2227  012356  001422                       BEQ     4$              ;NO - GO TO BANK LOOP TERMINATION
 2228  012360  005737  001520               TST     BMFLAG          ;IS THIS BAD MEMORY (WORST FIRST)
 2229  012364  001417                       BEQ     4$              ;NO - GO TO BANK LOOP TERMINATION
 2230  012366  005737  001514               TST     RRFLAG          ;RELOCATION REQUIRED?
 2231  012372  001014                       BNE     4$              ;YES - GO TO BANK LOOP TERMINATION
 2232  012374  004737  036366               CALL    SETPAT          ;SET HIGH PATTERN FOR CORRECT MEMORY
 2233                                       ;START OF PATTERN LOOP
 2234  012400  004737  036616         2$:   CALL    SETMAR          ;SET HIGH MARGINS (IF NOT INHIBITED)
 2235                                       ;START OF MARGIN LOOP
 2236  012404  004737  013334         3$:   CALL    MTEST           ;GO TEST CORRECT MEMORY
 2237                                       ;TERMINATION OF MARGIN LOOP
 2238  012410  004737  036556               CALL    DECMAR          ;NEXT LOWER MARGIN
 2239  012414  100373                       BPL     3$              ;IF NOT DONE - LOOP ON THIS MARGIN
 2240                                       ;TERMINATION OF PATTERN LOOP
 2241  012416  005337  001502               DEC     PATTERN         ;IS THIS THE LAST PATTERN?
 2242  012422  100366                       BPL     2$              ;NO - LOOP ON THIS PATTERN
 2243                                       ;TERMINATION OF BANK LOOP
 2244  012424  004737  036376         4$:   CALL    INCBNK          ;NEXT HIGHER BANK
 2245  012430  001346                       BNE     1$              ;IF NOT DONE - LOOP ON THIS BANK
 2246                                       ;END OF LOOPS
 2247  012432  005137  002140               COM     WORST           ;IS THIS AN EVEN NUMBERED PASS?
 2248  012436  001003                       BNE     5$              ;YES - SKIP
 2249                                       ;**NOTE** RECURSIVE CALL
 2250  012440  004737  012342               CALL    BAWPAR          ;CALL SELF
 2251  012444  000207                       RETURN
 2252  012446  005737  001516         5$:   TST     RLFLAG          ;HAVE WE BEEN RELOCATED?
 2253  012452  001401                       BEQ     6$              ;NO - SKIP
 2254  012454  000207                       RETURN                  ;YES - RETURN
 2255  012456  004737  034720         6$:   CALL    RELOCATE        ;MOVE & MAP PROGRAM
 2256  012462                               ON.ERROR THEN $RETURN
   (8)  012462  103001                                                             BCC L116
   (5)  012464  000207                                                             RTS PC
   (4)  012466                                                           L116:;;;;;;
 2257                                       ;**NOTE** RECURSIVE CALL
 2258  012466  004737  012342               CALL    BAWPAR          ;CALL SELF
 2259  012472  004737  035340               CALL    UNRELOCATE      ;UNMOVE & UNMAP PROGRAM
 2260  012476  000207                       RETURN
```

```
012500                          PAFBAF: SUBTST  <<PATTERNS FORWARD,BANKS FORWARD      **RECURSIVE**>>
                                ;*****************************************************************************
                                ;*SUBTEST        PATTERNS FORWARD,BANKS FORWARD      **RECURSIVE**
                                ;*****************************************************************************
012500  005037  001502                  CLR     PATTERN         ;SET PATTERN TO 0
                                        ;START OF PATTERN LOOP
012504  005037  001470          1$:     CLR     BANK            ;SET BANK TO 0
                                        ;START OF BANK LOOP
012510  004737  036052          2$:     CALL    EXBANK          ;EXAMINE BANK
012514  004737  036334                  CALL    BANKOK          ;CORRECT MEMORY FOR THIS BANK?
012520  001015                          BNE     4$              ;NO - GO TO BANK LOOP TERMINATOR
012522  005737  001506                  TST     ACFLAG          ;CAN WE ACCESS THIS BANK?
012526  001412                          BEQ     4$              ;NO - GO TO BANK LOOP TERMINATION
012530  005737  001514                  TST     RRFLAG          ;RELOCATION REQUIRED?
012534  001007                          BNE     4$              ;YES - GO TO BANK LOOP TERMINATION
012536  005037  001474                  CLR     MARGIN          ;SET MARGIN TO 0
                                        ;START OF MARGIN LOOP
012542  004737  013334          3$:     CALL    MTEST           ;GO TEST CORRECT MEMORY
                                        ;TERMINATION OF MARGIN LOOP
012546  004737  036412                  CALL    INCMAR          ;NEXT HIGHER MARGIN
012552  001373                          BNE     3$              ;IF NOT DONE - LOOP ON THIS MARGIN
                                        ;TERMINATION OF BANK LOOP
012554  004737  036376          4$:     CALL    INCBNK          ;NEXT HIGHER BANK
012560  001353                          BNE     2$              ;IF NOT DONE - LOOP ON THIS BANK
                                        ;TERMINATION OF PATTERN LOOP
012562  004737  036352                  CALL    INCRPT          ;NEXT HIGHER PATTERN
012566  001346                          BNE     1$              ;OK - LOOP; ELSE CONTINUE
                                        ;END OF LOOPS
012570  005137  001522                  COM     TMFLAG          ;COMPLEMENT TYPE OF MEMORY
                                                                ;IS THIS AN EVEN NUMBER PASS?
012574  001403                          BEQ     5$              ;YES - SKIP
                                        ;**NOTE** RECURSIVE CALL
012576  004737  012500                  CALL    PAFBAF          ;CALL SELF
012602  000207                          RETURN
012604  005737  001516          5$:     TST     RLFLAG          ;HAVE WE BEEN RELOCATED?
012610  001401                          BEQ     6$              ;NO - SKIP
012612  000207                          RETURN                  ;YES - RETURN
012614  004737  034720          6$:     CALL    RELOCATE        ;MOVE & MAP PROGRAM
012620                                  ON.ERROR THEN $RETURN
012620  103001                                                                          BCC L117
012622  000207                                                                          RTS PC
012624                                                                          L117:;;;;;;
                                        ;**NOTE** RECURSIVE CALL
012624  004737  012500                  CALL    PAFBAF          ;CALL SELF
012630  004737  035340                  CALL    UNRELOCATE      ;UNMOVE & UNMAP PROGRAM
012634  000207                          RETURN
```

```
                                PAFBAW: SUBTST  <<PATTERNS FORWARD,BANKS WORST FIRST  **RECURSIVE**>>
                                ;*****************************************************************************
                                ;*SUBTEST       PATTERNS FORWARD,BANKS WORST FIRST  **RECURSIVE**
                                ;*****************************************************************************
2305  012636  005037  001502          CLR     PATTERN             ;SET PATTERN TO 0
2306                                  ;START OF PATTERN LOOP
2307  012642  005037  001470    1$:   CLR     BANK                ;SET BANK TO 0
2308                                  ;START OF BANK LOOP
2309  012646  004737  036052    2$:   CALL    EXBANK              ;EXAMINE BANK
2310  012652  004737  036334          CALL    BANKOK              ;CORRECT MEMORY FOR THIS BANK?
2311  012656  001020                  BNE     4$                  ;NO - GO TO BANK LOOP TERMINATOR
2312  012660  005737  001506          TST     ACFLAG              ;CAN WE ACCESS THIS BANK?
2313  012664  001415                  BEQ     4$                  ;NO - GO TO BANK LOOP TERMINATION
2314  012666  005737  001520          TST     BMFLAG              ;IS THIS BAD MEMORY (WORST FIRST)
2315  012672  001412                  BEQ     4$                  ;NO - GO TO BANK LOOP TERMINATION
2316  012674  005737  001514          TST     RRFLAG              ;RELOCATION REQUIRED?
2317  012700  001007                  BNE     4$                  ;YES - GO TO BANK LOOP TERMINATION
2318  012702  005037  001474          CLR     MARGIN              ;SET MARGIN TO 0
2319                                  ;START OF MARGIN LOOP
2320  012706  004737  013334    3$:   CALL    MTEST               ;GO TEST CORRECT MEMORY
2321                                  ;TERMINATION OF MARGIN LOOP
2322  012712  004737  036412          CALL    INCMAR              ;NEXT HIGHER MARGIN
2323  012716  001373                  BNE     3$                  ;IF NOT DONE - LOOP ON THIS MARGIN
2324                                  ;TERMINATION OF BANK LOOP
2325  012720  004737  036376    4$:   CALL    INCBNK              ;NEXT HIGHER BANK
2326  012724  001350                  BNE     2$                  ;IF NOT DONE - LOOP ON THIS BANK
2327                                  ;TERMINATION OF PATTERN LOOP
2328  012726  004737  036352          CALL    INCRPT              ;NEXT HIGHER PATTERN
2329  012732  001343                  BNE     1$                  ;OK - LOOP; ELSE CONTINUE
2330                                  ;END OF LOOPS
2331  012734  005137  001522          COM     TMFLAG              ;COMPLEMENT TYPE OF MEMORY
2332                                                              ;IS THIS AN EVEN NUMBER PASS?
2333  012740  001403                  BEQ     5$                  ;YES - SKIP
2334                                  ;**NOTE** RECURSIVE CALL
2335  012742  004737  012636          CALL    PAFBAW              ;CALL SELF
2336  012746  000207                  RETURN
2337  012750  005137  002140    5$:   COM     WORST               ;4TH PASS?
2338  012754  001003                  BNE     6$                  ;YES - SKIP
2339                                  ;**NOTE** RECURSIVE CALL
2340  012756  004737  012636          CALL    PAFBAW              ;CALL SELF
2341  012762  000207                  RETURN
2342  012764  005737  001516    6$:   TST     RLFLAG              ;HAVE WE BEEN RELOCATED?
2343  012770  001401                  BEQ     7$                  ;NO - SKIP
2344  012772  000207                  RETURN                      ;YES - RETURN
2345  012774  004737  034720    7$:   CALL    RELOCATE            ;MOVE & MAP PROGRAM
2346  013000                        ON.ERROR THEN $RETURN
  (8) 013000  103001                                                          BCC L120
  (5) 013002  000207                                                          RTS PC
  (4) 013004                                                        L120:::::::
2347                                  ;**NOTE** RECURSIVE CALL
2348  013004  004737  012636          CALL    PAFBAW              ;CALL SELF
2349  013010  004737  035340          CALL    UNRELOCATE          ;UNMOVE & UNMAP PROGRAM
2350  013014  000207                  RETURN
```

```
 2352  013016                          PARBAF: SUBTST  ((PATTERNS REVERSE,BANKS FORWARD      **RECURSIVE**))
  (3)                                  ;*********************************************************************
  (3)                                  ;*SUBTEST       PATTERNS REVERSE,BANKS FORWARD      **RECURSIVE**
  (2)                                  ;*********************************************************************
 2353  013016  004737  036366                  CALL    HIPAT                   ;SET HIGHTEST PATTERNS
 2354                                           ;START OF PATTERN LOOP
 2355  013022  005037  001470          1$:     CLR     BANK                    ;SET BANK TO 0
 2356                                           ;START OF BANK LOOP
 2357  013026  004737  036052          2$:     CALL    EXBANK                  ;EXAMINE BANK
 2358  013032  004737  036334                  CALL    BANKOK                  ;CORRECT MEMORY FOR THIS BANK?
 2359  013036  001015                          BNE     4$                      ;NO - GO TO BANK LOOP TERMINATOR
 2360  013040  005737  001506                  TST     ACFLAG                  ;CAN WE ACCESS THIS BANK?
 2361  013044  001412                          BEQ     4$                      ;NO - GO TO BANK LOOP TERMINATION
 2362  013046  005737  001514                  TST     RRFLAG                  ;RELOCATION REQUIRED?
 2363  013052  001007                          BNE     4$                      ;YES - GO TO BANK LOOP TERMINATION
 2364  013054  004737  036616                  CALL    SETMAR                  ;SET HIGH MARGINS (IF NOT INHIBITED)
 2365                                           ;START OF MARGIN LOOP
 2366  013060  004737  013334          3$:     CALL    MTEST                   ;GO TEST CORRECT MEMORY
 2367                                           ;TERMINATION OF MARGIN LOOP
 2368  013064  004737  036556                  CALL    DECMAR                  ;NEXT LOWER MARGIN
 2369  013070  100373                          BPL     3$                      ;IF NOT DONE - LOOP ON THIS MARGIN
 2370                                           ;TERMINATION OF BANK LOOP
 2371  013072  004737  036376          4$:     CALL    INCBNK                  ;NEXT HIGHER BANK
 2372  013076  001353                          BNE     2$                      ;IF NOT DONE - LOOP ON THIS BANK
 2373                                           ;TERMINATION OF PATTERN LOOP
 2374  013100  005337  001502                  DEC     PATTERN                 ;NEXT LOWER PATTERN
 2375  013104  100346                          BPL     1$                      ;OK - LOOP; ELSE CONTINUE
 2376                                           ;END OF LOOPS
 2377  013106  005137  001522                  COM     TMFLAG                  ;COMPLEMENT TYPE OF MEMORY
 2378                                                                           ;IS THIS AN EVEN NUMBER PASS?
 2379  013112  001403                          BEQ     5$                      ;YES - SKIP
 2380                                           ;**NOTE** RECURSIVE CALL
 2381  013114  004737  013016                  CALL    PARBAF                  ;CALL SELF
 2382  013120  000207                          RETURN
 2383  013122  005737  001516          5$:     TST     RLFLAG                  ;HAVE WE BEEN RELOCATED?
 2384  013126  001401                          BEQ     6$                      ;NO - SKIP
 2385  013130  000207                          RETURN                          ;YES - RETURN
 2386  013132  004737  034720          6$:     CALL    RELOCATE                ;MOVE & MAP PROGRAM
 2387  013136                                  ON.ERROR THEN $RETURN
   (8) 013136  103001                                                                  BCC L121
   (5) 013140  000207                                                                  RTS PC
   (4) 013142                                                                  L121:::::::
 2388                                           ;**NOTE** RECURSIVE CALL
 2389  013142  004737  013016                  CALL    PARBAF                  ;CALL SELF
 2390  013146  004737  035340                  CALL    UNRELOCATE              ;UNMOVE & UNMAP PROGRAM
 2391  013152  000207                          RETURN
```

```
 2393  013154                          PARBAW: SUBTST  <<PATTERNS REVERSE,BANKS WORST FIRST  **RECURSIVE**>>
   (3)                                ;*********************************************************************
   (3)                                ;*SUBTEST       PATTERNS REVERSE,BANKS WORST FIRST  **RECURSIVE**
   (2)                                ;*********************************************************************
 2394  013154  004737  036366                  CALL    HIPAT           ;SET HIGHTEST PATTERN
 2395                                          ;START OF PATTERN LOOP
 2396  013160  005037  001470          1$:     CLR     BANK            ;SET BANK TO 0
 2397                                          ;START OF BANK LOOP
 2398  013164  004737  036052          2$:     CALL    EXBANK          ;EXAMINE BANK
 2399  013170  004737  036334                  CALL    BANKOK          ;CORRECT MEMORY FOR THIS BANK?
 2400  013174  001020                          BNE     4$              ;NO - GO TO BANK LOOP TERMINATOR
 2401  013176  005737  001506                  TST     ACFLAG          ;CAN WE ACCESS THIS BANK?
 2402  013202  001415                          BEQ     4$              ;NO - GO TO BANK LOOP TERMINATION
 2403  013204  005737  001520                  TST     BMFLAG          ;IS THIS BAD MEMORY (WORST FIRST)
 2404  013210  001412                          BEQ     4$              ;NO - GO TO BANK LOOP TERMINATION
 2405  013212  005737  001514                  TST     RRFLAG          ;RELOCATION REQUIRED?
 2406  013216  001007                          BNE     4$              ;YES - GO TO BANK LOOP TERMINATION
 2407  013220  004737  036616                  CALL    SETMAR          ;SET HIGH MARGINS (IF NOT INHIBITED)
 2408                                          ;START OF MARGIN LOOP
 2409  013224  004737  013334          3$:     CALL    MTEST           ;GO TEST CORRECT MEMORY
 2410                                          ;TERMINATION OF MARGIN LOOP
 2411  013230  004737  036556                  CALL    DECMAR          ;NEXT LOWER MARGIN
 2412  013234  100373                          BPL     3$              ;IF NOT DONE - LOOP ON THIS MARGIN
 2413                                          ;TERMINATION OF BANK LOOP
 2414  013236  004737  036376          4$:     CALL    INCBNK          ;NEXT HIGHER BANK
 2415  013242  001350                          BNE     2$              ;IF NOT DONE - LOOP ON THIS BANK
 2416                                          ;TERMINATION OF PATTERN LOOP
 2417  013244  005337  001502                  DEC     PATTERN         ;NEXT LOWER PATTERN
 2418  013250  100343                          BPL     1$              ;OK - LOOP; ELSE CONTINUE
 2419                                          ;END OF LOOPS
 2420  013252  005137  001522                  COM     TMFLAG          ;COMPLEMENT TYPE OF MEMORY
 2421                                                                  ;IS THIS AN EVEN NUMBER PASS?
 2422  013256  001403                          BEQ     5$              ;YES - SKIP
 2423                                          ;**NOTE** RECURSIVE CALL
 2424  013260  004737  013154                  CALL    PARBAW          ;CALL SELF
 2425  013264  000207                          RETURN
 2426  013266  005137  002140          5$:     COM     WORST           ;4TH PASS?
 2427  013272  001003                          BNE     6$              ;YES - SKIP
 2428                                          ;**NOTE** RECURSIVE CALL
 2429  013274  004737  013154                  CALL    PARBAW          ;CALL SELF
 2430  013300  000207                          RETURN
 2431  013302  005737  001516          6$:     TST     RLFLAG          ;HAVE WE BEEN RELOCATED?
 2432  013306  001401                          BEQ     7$              ;NO - SKIP
 2433  013310  000207                          RETURN                  ;YES - RETURN
 2434  013312  004737  034720          7$:     CALL    RELOCATE        ;MOVE & MAP PROGRAM
 2435  013316                                  ON.ERROR THEN $RETURN
   (8)  013316  103001                                                                   BCC L122
   (5)  013320  000207                                                                   RTS PC
   (4)  013322                                                                 L122::::::
 2436                                          ;**NOTE** RECURSIVE CALL
 2437  013322  004737  013154                  CALL    PARBAW          ;CALL SELF
 2438  013326  004737  035340                  CALL    UNRELOCATE      ;UNMOVE & UNMAP PROGRAM
 2439  013332  000207                          RETURN
```

H14


```
                                MTEST:  SUBTST  <<SUBR  SETUP MEMORY TEST>>
                                ;*****************************************************************************
                                ;*SUBTEST        SUBR    SETUP MEMORY TEST
                                ;*****************************************************************************
2443  013334  013700  001474          MOV     MARGIN,R0
2444  013340  006300                   ASL     R0
2445  013342  010037  177750          MOV     R0,MAINT                ;SET MARGINS
2446  013346                          SET     HEADER                  ;INITIALIZE HEADER MESSAGE TYPEOUT
 (1)  013346  012737  177777  002146  MOV     #-1,HEADER
2447  013354                          SET     MUT                     ;INDICATE THERE IS A MEMORY UNDER TEST
 (1)  013354  012737  177777  001500  MOV     #-1,MUT
2448  013362  005037  001734          CLR     PASFLG
2449  013366  005737  001510          TST     MKFLAG                  ;MK11?
2450  013372  001410                   BEQ     2$                      ;NO - SKIP
2451                                  ;IF CONTFLAG EQ #0 THEN CALL    MKCONTROL
2452  013374  005737  001704          TST     CONTFLAG
2453  013400  001002                   BNE     1$
2454  013402  004737  013434          CALL    MKCONTROL
2455  013406  004737  015774  1$:     CALL    MKTEST                  ;YES - DO MK11 TESTS
2456  013412  000402                   BR      3$
2457  013414  004737  016156  2$:     CALL    MJTEST                  ;DO MJ11 TESTS
2458  013420  005037  001500  3$:     CLR     MUT                     ;NOW - NO MEMORY UNDER TEST
2459  013424                          SET     HEADER                  ;ALLOW HEADERS NORMAL
 (1)  013424  012737  177777  002146  MOV     #-1,HEADER
2460  013432  000207                   RETURN
```

```
 2462  013434                          MKCONTROL:SUBTST        <<SUBR  TEST MK11 CONTROLLER LOGIC DISPATCH>>
  (3)                                  ;*****************************************************************************
  (3)                                  ;*SUBTEST        SUBR    TEST MK11 CONTROLLER LOGIC DISPATCH
  (2)                                  ;*****************************************************************************
 2463  013434                                  PUSH    BANK
  (2)  013434  013746  001470                                                                  MOV BANK,-(SP)
 2464  013440  112737  000200  001774          MOVB    #200,CSRINDEX           ;SET BIT FOR CONTROLLER
 2465  013446  112737  000001  001775          MOVB    #1,CSROUDEX             ;SET BIT FOR CONTROLLER
 2466  013454                                  FOR CSRLOOP := #'0 TO #'7
  (4)  013454  012737  000060  001776                                                          MOV #'0,CSRLOOP
  (4)  013462                                                                          B20:;;;;;;
 2467  013462                                    IFB CSRINDEX SET.IN MKCSRS
  (5)  013462  133737  001774  001706                                                          BITB CSRINDEX,MKCSRS
  (7)  013470  001536                                                                          BEQ L123
 2468  013472  004737  014034                      CALL        ISECCON
 2469  013476                                      FOR SIDE := #'0 TO #'1
  (4)  013476  012737  000060  002000                                                          MOV #'0,SIDE
  (4)  013504                                                                          B21:;;;;;;
 2470  013504  004737  014344                        CALL      INTKRAP
 2471  013510                                        BEGIN CSRSTUFF
  (3)  013510                                                                          B22:;;;;;;
 2472  013510  005037  002002                        CLR       SUCCESS
 2473  013514                                        FOR BANK := CSRLBANK DOWNTO CSRFBANK
  (4)  013514  013737  001770  001470                                                          MOV CSRLBANK,BANK
  (4)  013522                                                                          B23:;;;;;;
 2474  013522  004737  036052                          CALL    EXBANK
 2475  013526                                          IF ACFLAG NE #0 AND RRFLAG EQ #0
  (4)  013526  005737  001506                                                                  TST ACFLAG
  (6)  013532  001466                                                                          BEQ L124
  (6)  013534  005737  001514                                                                  TST RRFLAG
  (8)  013540  001063                                                                          BNE L124
 2476  013542                                            MAP   BANK
  (3)  013542  010346                                                                          MOV R3,-(SP)
  (1)  013544  013703  001470                  MOV     BANK,R3
  (1)  013550  004737  034510                  CALL    MAPPER
  (3)  013554  012603                                                                          MOV (SP)+,R3
 2477  013556  000241                                    CLC
 2478  013560                                            IF SPECIAL NE #0 THEN $CALL SIDEOK
  (4)  013560  005737  002010                                                                  TST SPECIAL
  (6)  013564  001402                                                                          BEQ L125
  (4)  013566  004737  015640                                                                  JSR PC,SIDEOK
  (3)  013572                                                                          L125:;;;;;;
 2479  013572                                            ON.NOERROR
  (8)  013572  103446                                                                          BCS L126
 2480  013574                                              FOR TESTADD := CSRFIRST TO #LAST BY CSRINC
  (3)  013574  013737  001710  002070                                                          MOV CSRFIRST,TESTADD
  (3)  013602                                                                          B24:;;;;;;
 2481  013602  004737  014130                                CALL      SBETEST
 2482  013606                                                ON.NOERROR
  (8)  013606  103431                                                                          BCS L127
 2483  013610  104424                                          CACHOFF
 2484  013612  005037  001464                                  CLR     NOPAR           ;INDICATE PARITY ACTION
 2485  013616                                                  FOR I := #0 TO #37
  (4)  013616  005037  002130                                                                  CLR I
  (4)  013622                                                                          B25:;;;;;;
 2486  013622                                                    SET   HEADER
```

```
  (1)  013622  012737  177777  002146          MOV     #-1,HEADER
2487   013630  005037  001734                          CLR   PASFLG
2488   013634                                          LET R0 := I
  (3)  013634  013700  002130                          CALL CSRCASE
2489   013640  004737  014666                        END ;OF FOR I
2490   013644
  (6)  013644  005237  002130
  (5)  013650  023727  002130  000037
  (5)  013656  003761
  (5)  013660
2491   013660  104423                              CACHON
2492   013662                                      SET     SUCCESS
  (1)  013662  012737  177777  002002          MOV     #-1,SUCCESS
2493   013670                                      LEAVE CSRSTUFF
  (3)  013670  000415
2494   013672                                    END ;OF ON.NOERROR
  (3)  013672
2495   013672                                  END ;OF FOR TESTADD
  (6)  013672  063737  001772  002070
  (5)  013700  023727  002070  157776
  (5)  013706  003735
  (5)  013710
2496   013710                                    END ;OF ON.NOERROR
  (3)  013710
2497   013710                                  END ;OF IF
  (3)  013710
2498   013710                                END ;OF FOR BANK
  (6)  013710  005337  001470
  (5)  013714  023737  001470  001766
  (5)  013722  002277
  (5)  013724
2499   013724                                END CSRSTUFF
  (3)  013724
2500   013724                                IF SUCCESS EQ #0
  (4)  013724  005737  002002
  (6)  013730  001010
2501   013732  113737  001776  070204            MOVB    CSRLOOP,MSGA34          ;SETUP CSR NUMBER
2502   013740  113737  002000  070221            MOVB    SIDE,MSGB34             ;SETUP SIDE (0 OR 1)
2503   013746                                    TYPE    MSG034
  (1)  013746  104401  070150              TYPEIT  ,MSG034
2504   013752                                  END ;OF IF SUCCESS
  (3)  013752
2505   013752                                END ;OF FOR SIDE
  (6)  013752  005237  002000
  (5)  013756  023727  002000  000061
  (5)  013764  003647
  (5)  013766
2506   013766                                  END ;OF IFB CSRINDEX
  (3)  013766
2507   013766  000241                          CLC
2508   013770  106037  001774                  RORB  CSRINDEX                  ;CHANGE TO NEXT CONTROLLER
2509   013774  106337  001775                  ASLB  CSROUDEX                  ;CHANGE TO NEXT CONTROLLER
2510   014000                                END ;OF FOR CSRLOOP
  (6)  014000  005237  001776
  (5)  014004  023727  001776  000067
  (5)  014012  003623
```

```
                MOV I,R0

                INC I
                CMP I,#37
                BLE B25
E25:::::::

                BR E22

L127:::::::

                ADD CSRINC,TESTADD
                CMP TESTADD,#LAST
                BLE B24
E24:::::::

L126:::::::

L124:::::::

                DEC BANK
                CMP BANK,CSRFBANK
                BGE B23
E23:::::::

E22:::::::

                TST SUCCESS
                BNE L130

L130:::::::

                INC SIDE
                CMP SIDE,#'1
                BLE B21
E21:::::::

L123:::::::

                INC CSRLOOP
                CMP CSRLOOP,#'7
                BLE B20
```

K14

MTEST MODES MACY11 30A(1052) 28-FEB-78 09:20 PAGE 12-3
CEMKAA.MA1 24-FEB-78 11:01 SUBR TEST MK11 CONTROLLER LOGIC DISPATCH SEQ 0179

```
  (5)  014014                                                                  E20::::::::
 2511  014014                                  SET     CONTFLAG
  (1)  014014  012737  177777  001704          MOV     #-1,CONTFLAG
 2512  014022                                  POP     BANK
  (2)  014022  012637  001470                                                  MOV (SP)+,BANK
 2513  014026  004737  036052                  CALL    EXBANK
 2514  014032  000207                          RETURN
```

```
2516  014034                          ISECCON:SUBTST  <<IS ECC ON?>>
 (3)                                  ;*****************************************************************************
 (3)                                  ;*SUBTEST       IS ECC ON?
 (2)                                  ;*****************************************************************************
2517                                          ;SETUP CSR NUMBER
2518  014034  005000                         CLR     R0                      ;START A ZERO
2519  014036  113702  001775                 MOVB    CSRINDEX,R2             ;GET CSR INDEX #2
2520  014042  106002                 1$:     RORB    R2                      ;GET CSR BIT IN CARRY (MAYBE)
2521  014044  103403                         BCS     2$                      ;I GOT IT - SKIP
2522  014046  062700  000004                 ADD     #4,R0                   ;BUMP CSR ADDRESS LSB'S BY 4
2523  014052  000773                         BR      1$                      ;LOOP TILL I FIND THE CSR
2524  014054  010037  001634         2$:     MOV     R0,CSRNO                ;UPDATE CSR NUMBER
2525  014060  104503                         CLR1CSR
2526  014062  104426                         READCSR
2527  014064                                  IF #BIT1 SET.IN CSR
 (4)  014064  032737  000002  001526                                                    BIT #BIT1,CSR
 (6)  014072  001413                                                                    BEQ L131
2528  014074                                    TYPE  MSG078                  ;WARNING!  ECC IS DISABLED ON BOX
 (1)  014074  104401  071737
                                          TYPEIT ,MSG078
2529  014100  113737  001776  067051        MOVB  CSRLOOP,MSG015
2530  014106                                TYPE  MSG015
 (1)  014106  104401  067051
                                          TYPEIT ,MSG015
2531  014112                                SET   NOECCFLAG
 (1   014112  012737  177777  002056       MOV   #-1,NOECCFLAG
2532  014120                                ELSE
 (3)  014120  000402                                                                    BR L132
 (3)  014122                                                                  L131:::::::
2533  014122  005037  002056                 CLR   NOECCFLAG
2534  014126                                END :OF IF #BIT1
 (3)  014126                                                                  L132:::::::
2535  014126  000207                       RETURN
```

```
2537  014130                        SBETEST:SUBTST  <<CHECK FOR SBE FREE LOCATIONS>>
 (3)                                ;*****************************************************************************
 (3)                                ;*SUBTEST       CHECK FOR SBE FREE LOCATIONS
 (2)                                ;*****************************************************************************
2538                                        .ENABL  LSB
2539  014130  013701  002070               MOV     TESTADD,R1
2540  014134                              SUPERVISOR
 (1)  014134  052737  040000  177776      BIS     #BIT14,PSW              ;DO IT TO IT!
2541  014142  104424                      CACHOFF
2542  014144  104471                      ECC1DIS
2543  014146  005011                      CLR     (R1)
2544  014150  005061  000002              CLR     2(R1)
2545  014154  005711                      TST     (R1)
2546  014156  001061                      BNE     1$
2547  014160  005761  000002              TST     2(R1)
2548  014164  001056                      BNE     1$
2549
2550  014166  104503                      CLR1CSR
2551  014170  005011                      CLR     (R1)
2552  014172  005061  000002              CLR     2(R1)
2553  014176  005711                      TST     (R1)
2554  014200  001050                      BNE     1$
2555  014202  005761  000002              TST     2(R1)
2556  014206  001045                      BNE     1$
2557
2558  014210  104510                      TSTREAD                         ;TEST FOR ERRORS WITHOUT FETCHES
2559  014212                              IF #BIT15!BIT4 SET.IN CSR THEN GOTO 1$
 (5)  014212  032737  100020  001526                                                BIT #BIT15!BIT4,CSR
 (5)  014220  001040                                                                BNE 1$
2560
2561  014222  104471                      ECC1DIS
2562  014224  005111                      COM     (R1)
2563  014226  005161  000002              COM     2(R1)
2564  014232  023711  002154              CMP     ONES,(R1)
2565  014236  001031                      BNE     1$
2566  014240  023761  002154  000002      CMP     ONES,2(R1)
2567  014246  001025                      BNE     1$
2568
2569  014250  104503                      CLR1CSR
2570  014252  005011                      CLR     (R1)
2571  014254  012761  100000  000002      MOV     #BIT15,2(R1)
2572  014262  005711                      TST     (R1)
2573  014264  001016                      BNE     1$
2574  014266  022761  100000  000002      CMP     #BIT15,2(R1)
2575  014274  001012                      BNE     1$
2576
2577  014276  104510                      TSTREAD                         ;TEST FOR ERRORS WITHOUT FETCHES
2578  014300                              IF #BIT15!BIT4 SET.IN CSR THEN GOTO 1$
 (5)  014300  032737  100020  001526                                                BIT #BIT15!BIT4,CSR
 (5)  014306  001005                                                                BNE 1$
2579
2580  014310  104417                      KERNEL
2581  014312  104473                      ECC1INIT
2582  014314  104423                      CACHON
2583  014316                              $RETURN NOERROR
 (2)  014316  000241                                                                CLC
```

N14


```
  (2)  014320  000207                                                    RTS PC
 2584
 2585  014322  104503                  1$:     CLR1CSR
 2586  014324  005011                          CLR     (R1)
 2587  014326  005061  000002                  CLR     2(R1)
 2588  014332  104417                          KERNEL
 2589  014334  104473                          ECC1INIT
 2590  014336  104423                          CACHON
 2591  014340                                  $RETURN ERROR
  (2)  014340  000261                                                    SEC
  (2)  014342  000207                                                    RTS PC
 2592                                          .DSABL  LSB
```

```
2594  014344                          INTKRAP:SUBTST  <<SETUP INTERLEAVE INFO>>
 (3)                                  ;*****************************************************************************
 (3)                                  ;*SUBTEST       SETUP INTERLEAVE INFO
 (2)                                  ;*****************************************************************************
2595  014344                                PUSH    R0                                              MOV R0,-(SP)
 (2)  014344  010046
2596  014346  004737  014472                CALL    GETSIZE
2597  014352  013737  001636  001766       MOV     FIRSTB,CSRFBANK
2598  014360  013737  001640  001770       MOV     LASTB,CSRLBANK
2599  014366  005037  002010               CLR     SPECIAL
2600  014372  104426                       READCSR                 ;READ CSR
2601  014374  013700  001530               MOV     CSR+2,R0
2602  014400  072027  177765               ASH     #-11.,R0
2603  014404  042700  177760               BIC     #^C17,R0        ;R0 = (0..17) = BITS 14,13,12,11 OF CSR+2
2604  014410                               CASE R0                 ;IN THE CASE THAT R0 = 0 CALL NEXT; IF R0 = 1 CALL NEXT+
 (2)  014410  010046                                                                             MOV R0,-(SP)
 (2)  014412  006316                                                                             ASL @SP
 (3)  014414  004737  014460                                                                     JSR PC,L133
2605  014420  015006                         INT..0
2606  014422  015032                         INT..1
2607  014424  015076                         INT..2
2608  014426  015122                         INT..3
2609  014430  015166                         INT..4
2610  014432  015212                         INT..5
2611  014434  015256                         INT..6
2612  014436  015302                         INT..7
2613  014440  015346                         INT.10
2614  014442  015346                         INT.11
2615  014444  015370                         INT.12
2616  014446  015414                         INT.13
2617  014450  015460                         INT.14
2618  014452  015504                         INT.15
2619  014454  015550                         INT.16
2620  014456  015574                         INT.17
2621  014460                               END                     ;OF CASE
 (3)  014460                                                                  L133:::::::
 (2)  014460  062616                                                                             ADD (SP)+,@SP
 (2)  014462  013646                                                                             MOV @(SP)+,-(SP)
 (2)  014464  004736                                                                             JSR PC,@(SP)+
2622  014466                                POP     R0                                            MOV (SP)+,R0
 (2)  014466  012600
2623  014470  000207                        RETURN
```

```
 2625  014472                          GETSIZE:SUBTST  ((SUBR  GET SIZE FROM CSR))
  (3)                                  ;*****************************************************************************
  (3)                                  ;*SUBTEST       SUBR    GET SIZE FROM CSR
  (2)                                  ;*****************************************************************************
 2626                                         ;SETUP FIRSTB AND LASTB
 2627  014472                                 PUSH    R0
  (2)  014472  010046                                                                    MOV R0,-(SP)
 2628  014474  052737  000010  001526         BIS     #BIT3,CSR                ;GET SET TO READ CSR
 2629  014502  042737  000006  001526         BIC     #BIT2!BIT1,CSR
 2630  014510                                 SET     INTOVR          ;OVERIDE INTERLEAVE TO PRESERVE READING
  (1)  014510  012737  177777  002042         MOV     #-1,INTOVR
 2631  014516  104425                         LOADCSR
 2632  014520  005037  002042                 CLR     INTOVR
 2633  014524  104426                         READCSR
 2634  014526  042737  000010  001526         BIC     #BIT3,CSR                ;RESTORE THIS FROM "SIZE" MODE
 2635  014534  113700  001527                 MOVB    CSR+1,R0                 ;GET SIZE
 2636  014540  042700  000200                 BIC     #BIT7,R0
 2637  014544  006300                         ASL     R0                       ;R0 = # OF BANKS
 2638  014546  110037  001640                 MOVB    R0,LASTB
 2639  014552  013700  001530                 MOV     CSR+2,R0
 2640                                         ;CORRECT LAST BANK FOR EXTERNAL INTERLEAVE
 2641  014556                                 IF #BIT12 SET.IN R0 THEN LET LASTB := LASTB L.SHIFT 1
  (4)  014556  032700  010000                                                            BIT #BIT12,R0
  (6)  014562  001402                                                                    BEQ L134
  (7)  014564  006337  001640                                                            ASL LASTB
  (3)  014570                                                                          L134:::::::
 2642  014570                                 IF #BIT13 SET.IN R0 THEN LET LASTB := LASTB L.SHIFT 1
  (4)  014570  032700  020000                                                            BIT #BIT13,R0
  (6)  014574  001402                                                                    BEQ L135
  (7)  014576  006337  001640                                                            ASL LASTB
  (3)  014602                                                                          L135:::::::
 2643  014602  042700  177000                 BIC     #177000,R0
 2644  014606  006300                         ASL     R0
 2645  014610  010037  001636                 MOV     R0,FIRSTB
 2646  014614  063737  001636  001640         ADD     FIRSTB,LASTB
 2647  014622  005337  001640                 DEC     LASTB
 2648  014626                                 IF FIRSTB HI #167 THEN LET FIRSTB := #167
  (4)  014626  023727  001636  000167                                                    CMP FIRSTB,#167
  (6)  014634  101403                                                                    BLOS L136
  (5)  014636  012737  000167  001636                                                    MOV #167,FIRSTB
  (3)  014644                                                                          L136:::::::
 2649  014644                                 IF LASTB HI #167 THEN LET LASTB := #167
  (4)  014644  023727  001640  000167                                                    CMP LASTB,#167
  (6)  014652  101403                                                                    BLOS L137
  (5)  014654  012737  000167  001640                                                    MOV #167,LASTB
  (3)  014662                                                                          L137:::::::
 2650  014662                                 POP     R0
  (2)  014662  012600                                                                    MOV (SP)+,R0
 2651  014664  000207                         RETURN
```

```
                                CSRCASE:SUBTST  <<CSR PATTERN CASE STATEMENT>>
                                ;*****************************************************************************
                                ;*SUBTEST       CSR PATTERN CASE STATEMENT
                                ;*****************************************************************************
                                        CASE R0
014666                                                                          MOV R0,-(SP)
014666  010046                                                                 ASL @SP
014670  006316                                                                 JSR PC,L140
014672  004737  014776
                                ;WARNING IF YOU CHANGE THIS TABLE ALSO
                                ;CHANGE "$DDW0" - "$DDW5" (THE PATTERN BIT MAP)

                                ;PAT  TIME              DISCRIPTION
014676  016760                  MKCSRT: MT0006          ;<1 SEC         INITIAL DATA TEST
014700  017056                          MT0010          ;<1 SEC         BYTE ADDRESSING TEST
014702  020442                          MT0025          ;<1 SEC         INTERRUPT ENABLE TEST
014704  017112                          MT0011          ;<2 SEC         CREATE SINGLE BIT ERROR TEST
014706  017170                          MT0012          ;<1 SEC         WRITE BYTE CLEARS SBE TEST
014710  017246                          MT0013          ; 1 SEC         CREATE DOUBLE BIT ERROR TEST
014712  017322                          MT0014          ; 1 SEC         WRITE INHIBIT DURING DATIP WITH DBE
014714  017400                          MT0015          ; 1 SEC         WRITE INHIBIT OF BYTE WITH DBE
014716  017456                          MT0016          ;<1 SEC         WRITE INHIBIT OF WORD WITH DBE
014720  021532                          MT0999          ; 0 SEC         NULL TEST
014722  021532                          MT0999          ; 0 SEC         NULL TEST
014724  021532                          MT0999          ; 0 SEC         NULL TEST
014726  021532                          MT0999          ; 0 SEC         NULL TEST
014730  021532                          MT0999          ; 0 SEC         NULL TEST
014732  021532                          MT0999          ; 0 SEC         NULL TEST
014734  021532                          MT0999          ; 0 SEC         NULL TEST
014736  021532                          MT0999          ; 0 SEC         NULL TEST
014740  021532                          MT0999          ; 0 SEC         NULL TEST
014742  021532                          MT0999          ; 0 SEC         NULL TEST
014744  021532                          MT0999          ; 0 SEC         NULL TEST
014746  021532                          MT0999          ; 0 SEC         NULL TEST
014750  021532                          MT0999          ; 0 SEC         NULL TEST
014752  021532                          MT0999          ; 0 SEC         NULL TEST
014754  021532                          MT0999          ; 0 SEC         NULL TEST
014756  021532                          MT0999          ; 0 SEC         NULL TEST
014760  021532                          MT0999          ; 0 SEC         NULL TEST
014762  021532                          MT0999          ; 0 SEC         NULL TEST
014764  021532                          MT0999          ; 0 SEC         NULL TEST
014766  021532                          MT0999          ; 0 SEC         NULL TEST
014770  021532                          MT0999          ; 0 SEC         NULL TEST
014772  021532                          MT0999          ; 0 SEC         NULL TEST
014774  021532                          MT0999          ; 0 SEC         NULL TEST
014776                          END ;OF CASE R0
014776                                                                  L140::::;;;
014776  062616                                                                  ADD (SP)+,@SP
015000  013646                                                                  MOV @(SP)+,-(SP)
015002  004736                                                                  JSR PC,@(SP)+
015004  000207                  RETURN
```

```
                        SUBTST  <<DETERMINE ADDRESSING ACCORDING TO INTERLEAVE>>
                ;*****************************************************************************
                ;*SUBTEST       DETERMINE ADDRESSING ACCORDING TO INTERLEAVE
                ;*****************************************************************************
                        ;CSRFBANK := THE FIRST BANK OF THIS CONTROLLER OF THIS CSR
                        ;CSRLBANK := THE LAST BANK OF THIS CONTROLLER OF THIS CSR
                        ;CSRFIRST := THE FIRST ADDRESS OF THIS CONTROLLER OF THIS CSR
                        ;CSRINC := THE NUMBER TO ADD TO THE ADDRESS PAIR
                        ;       TO RETURN TO THIS CONTROLLER OF THIS CSR
                        ;SPECIAL := A FLAG SET IF THIS BOX IS NOT INTERNALLY INTERLEAVED

                        ;NO INTERNAL INTERLEAVE, NO EXTERNAL INTERLEAVE
015006          INT..0: LET CSRFIRST := #FIRST                          MOV #FIRST,CSRFIRST
015014                  SET     SPECIAL
015014  012737  177777  002010  MOV     #-1,SPECIAL
                        LET CSRINC := #4                                MOV #4,CSRINC
015030  000207          RETURN

                        ;INTERNAL INTERLEAVE, NO EXTERNAL INTERLEAVE
015032          INT..1: LET CSRFBANK := FIRSTB                          MOV FIRSTB,CSRFBANK
015040                  IFB SIDE EQ #'0                                 CMPB SIDE,#'0
015046  001004                                                          BNE L141
                          LET CSRFIRST := #FIRST                        MOV #FIRST,CSRFIRST
015056                  ELSE                                            BR L142
015060                                                          L141:::::::
                          LET CSRFIRST := #FIRST+4                      MOV #FIRST+4,CSRFIRST
015066                  END ;OF IFB
015066                                                          L142:::::::
                        LET CSRINC := #10
015066  012737  000010  001772  MOV     #10,CSRINC
015074  000207          RETURN

                        ;NO INTERNAL INTERLEAVE, 2-1 EXTERNAL INTERLEAVE (WITH EVEN DOUBLE WORDS SELECTE
015076          INT..2: LET CSRFIRST := #FIRST                          MOV #FIRST,CSRFIRST
015104                  SET     SPECIAL
015104  012737  177777  002010  MOV     #-1,SPECIAL
015112                  LET CSRINC := #10
015112  012737  000010  001772  MOV     #10,CSRINC
015120  000207          RETURN

                        ;INTERNAL INTERLEAVE, 2-1 EXTERNAL INTERLEAVE (WITH EVEN DOUBLE WORDS SELECTED?)
015122          INT..3: LET CSRFBANK := FIRSTB                          MOV FIRSTB,CSRFBANK
015130                  IFB SIDE EQ #'0                                 CMPB SIDE,#'0
015136  001004                                                          BNE L143
                          LET CSRFIRST := #FIRST                        MOV #FIRST,CSRFIRST
```

F15

MTEST MODES MACY11 30A(1052) 28-FEB-78 09:20 PAGE 13-2
CEMKAA.MA1 24-FEB-78 11:01 DETERMINE ADDRESSING ACCORDING TO INTERLEAVE SEQ 0187

```
015146                                  ELSE
015146  000403                                                                  BR L144
015150                                                                  L143:;;;;;;
015150                                    LET CSRFIRST := #FIRST+10
015150  012737  060010  001710                                                  MOV #FIRST+10,CSRFIRST
015156                                  END ;OF IFB
015156                                                                  L144:;;;;;;
015156                                  LET CSRINC := #20
015156  012737  000020  001772                                                  MOV #20,CSRINC
015164  000207                          RETURN

                                        ;NO INTERNAL INTERLEAVE, 4-1 EXTERNAL INTERLEAVE (WITH A03=0,A02=0)
015166                          INT..4: LET CSRFIRST := #FIRST
015166  012737  060000  001710                                                  MOV #FIRST,CSRFIRST
015174                                  SET     SPECIAL
015174  012737  177777  002010          MOV     #-1,SPECIAL
015202                                  LET CSRINC := #20
015202  012737  000020  001772                                                  MOV #20,CSRINC
015210  000207                          RETURN

                                        ;INTERNAL INTERLEAVE, 4-1 EXTERNAL INTERLEAVE (WITH A03=0,A02=0)
015212                          INT..5: LET CSRFBANK := FIRSTB
015212  013737  001636  001766                                                  MOV FIRSTB,CSRFBANK
015220                                  IFB SIDE EQ #'0
015220  123727  002000  000060                                                  CMPB SIDE,#'0
015226  001004                                                                  BNE L145
015230                                    LET CSRFIRST := #FIRST
015230  012737  060000  001710                                                  MOV #FIRST,CSRFIRST
015236                                  ELSE
015236  000403                                                                  BR L146
015240                                                                  L145:;;;;;;
015240                                    LET CSRFIRST := #FIRST+20
015240  012737  060020  001710                                                  MOV #FIRST+20,CSRFIRST
015246                                  END ;OF IFB
015246                                                                  L146:;;;;;;
015246                                  LET CSRINC := #40
015246  012737  000040  001772                                                  MOV #40,CSRINC
015254  000207                          RETURN

                                        ;NO INTERNAL INTERLEAVE, 4-1 EXTERNAL INTERLEAVE (WITH A03=1,A02=0)
015256                          INT..6: LET CSRFIRST := #FIRST+10
015256  012737  060010  001710                                                  MOV #FIRST+10,CSRFIRST
015264                                  SET     SPECIAL
015264  012737  177777  002010          MOV     #-1,SPECIAL
015272                                  LET CSRINC := #20
015272  012737  000020  001772                                                  MOV #20,CSRINC
015300  000207                          RETURN

                                        ;INTERNAL INTERLEAVE, 4-1 EXTERNAL INTERLEAVE (WITH A03=1,A02=0)
015302                          INT..7: LET CSRFBANK := FIRSTB
015302  013737  001636  001766                                                  MOV FIRSTB,CSRFBANK
015310                                  IFB SIDE EQ #'0
015310  123727  002000  000060                                                  CMPB SIDE,#'0
015316  001004                                                                  BNE L147
015320                                    LET CSRFIRST := #FIRST+10
015320  012737  060010  001710                                                  MOV #FIRST+10,CSRFIRST
```

```
015326                                  ELSE
015326  000403                                                                  BR L150
015330                                                                  L147:::::::
015330                                    LET CSRFIRST := #FIRST+30
015330  012737  060030  001710                                                  MOV #FIRST+30,CSRFIRST
015336                                  END ;OF IFB
015336                                                                  L150:::::::
015336                                  LET CSRINC := #40
015336  012737  000040  001772                                                  MOV #40,CSRINC
015344  000207                          RETURN

                                        ;IMPOSSIBLE CONDITION
                                        ;NO INTERNAL INTERLEAVE, NO EXTERNAL INTERLEAVE (WITH ODD DOUBLE WORDS SELECTED?
015346                          INT.10: ;FALL THRU TO INT.11

                                        ;IMPOSSIBLE CONDITION
                                        ;INTERNAL INTERLEAVE, NO EXTERNAL INTERLEAVE (WITH ODD DOUBLE WORDS SELECTED?)
015346                          INT.11: IF FSINFLAG IS FALSE
015346  005737  002126                                                          TST FSINFLAG
015352  001005                                                                  BNE L151
015354                                    SET   FSINFLAG
015354  012737  177777  002126          MOV     #-1,FSINFLAG
015362                                    TYPE  MSG097
015362  104401  072437                  TYPEIT  ,MSG097
015366                                  END ;OF IF FSINFLAG
015366                                                                  L151:::::::
015366                                  GOTO    INT..1
015366  000621                                                                  BR INT..1

                                        ;NO INTERNAL INTERLEAVE, 2-1 EXTERNAL INTERLEAVE (WITH ODD DOUBLE WORDS SELECTED
015370                          INT.12: LET CSRFIRST := #FIRST+4
015370  012737  060004  001710                                                  MOV #FIRST+4,CSRFIRST
015376                                  SET     SPECIAL
015376  012737  177777  002010          MOV     #-1,SPECIAL
015404                                  LET CSRINC := #10
015404  012737  000010  001772                                                  MOV #10,CSRINC
015412  000207                          RETURN

                                        ;INTERNAL INTERLEAVE, 2-1 EXTERNAL INTERLEAVE (WITH ODD DOUBLE WORDS SELECTED?)
015414                          INT.13: LET CSRFBANK := FIRSTB
015414  013737  001636  001766                                                  MOV FIRSTB,CSRFBANK
015422                                  IFB SIDE EQ #'0
015422  123727  002000  000060                                                  CMPB SIDE,#'0
015430  001004                                                                  BNE L152
015432                                    LET CSRFIRST := #FIRST+4
015432  012737  060004  001710                                                  MOV #FIRST+4,CSRFIRST
015440                                  ELSE
015440  000403                                                                  BR L153
015442                                                                  L152:::::::
015442                                    LET CSRFIRST := #FIRST+14
015442  012737  060014  001710                                                  MOV #FIRST+14,CSRFIRST
015450                                  END ;OF IFB
015450                                                                  L153:::::::
015450                                  LET CSRINC := #20
015450  012737  000020  001772                                                  MOV #20,CSRINC
015456  000207                          RETURN
```

```
                                                ;NO INTERNAL INTERLEAVE, 4-1 EXTERNAL INTERLEAVE (WITH A03=0,A02=1)
015460                                  INT.14: LET CSRFIRST := #FIRST+4
(3) 015460  012737  060004  001710                                                                  MOV #FIRST+4,CSRFIRST
015466                                          SET     SPECIAL
(1) 015466  012737  177777  002010              MOV     #-1,SPECIAL
015474                                          LET CSRINC := #20
(3) 015474  012737  000020  001772                                                                  MOV #20,CSRINC
015502  000207                                  RETURN

                                                ;INTERNAL INTERLEAVE, 4-1 EXTERNAL INTERLEAVE (WITH A03=0,A02=1)
015504                                  INT.15: LET CSRFBANK := FIRSTB
(3) 015504  013737  001636  001766                                                                  MOV FIRSTB,CSRFBANK
015512                                          IFB SIDE EQ #'0
(5) 015512  123727  002000  000060                                                                  CMPB SIDE,#'0
(7) 015520  001004                                                                                  BNE L154
015522                                            LET CSRFIRST := #FIRST+4
(3) 015522  012737  060004  001710                                                                  MOV #FIRST+4,CSRFIRST
015530                                          ELSE
(3) 015530  000403                                                                                  BR L155
(3) 015532                                                                                  L154:;;;;;;
015532                                            LET CSRFIRST := #FIRST+24
(3) 015532  012737  060024  001710                                                                  MOV #FIRST+24,CSRFIRST
015540                                          END ;OF IFB
(3) 015540                                                                                  L155:;;;;;;
015540                                          LET CSRINC := #40
(3) 015540  012737  000040  001772                                                                  MOV #40,CSRINC
015546  000207                                  RETURN

                                                ;NO INTERNAL INTERLEAVE, 4-1 EXTERNAL INTERLEAVE (WITH A03=1,A02=1)
015550                                  INT.16: LET CSRFIRST := #FIRST+14
(3) 015550  012737  060014  001710                                                                  MOV #FIRST+14,CSRFIRST
015556                                          SET     SPECIAL
(1) 015556  012737  177777  002010              MOV     #-1,SPECIAL
015564                                          LET CSRINC := #20
(3) 015564  012737  000020  001772                                                                  MOV #20,CSRINC
015572  000207                                  RETURN

                                                ;INTERNAL INTERLEAVE, 4-1 EXTERNAL INTERLEAVE (WITH A03=1,A02=1)
015574                                  INT.17: LET CSRFBANK := FIRSTB
(3) 015574  013737  001636  001766                                                                  MOV FIRSTB,CSRFBANK
015602                                          IFB SIDE EQ #'0
(5) 015602  123727  002000  000060                                                                  CMPB SIDE,#'0
(7) 015610  001004                                                                                  BNE L156
015612                                            LET CSRFIRST := #FIRST+14
(3) 015612  012737  060014  001710                                                                  MOV #FIRST+14,CSRFIRST
015620                                          ELSE
(3) 015620  000403                                                                                  BR L157
(3) 015622                                                                                  L156:;;;;;;
015622                                            LET CSRFIRST := #FIRST+34
(3) 015622  012737  060034  001710                                                                  MOV #FIRST+34,CSRFIRST
015630                                          END ;OF IFB
(3) 015630                                                                                  L157:;;;;;;
015630                                          LET CSRINC := #40
(3) 015630  012737  000040  001772                                                                  MOV #40,CSRINC
015636  000207                                  RETURN
```

```
2826
```

```
 2828  015640                          SIDEOK: SUBTST  ;;DETERMINE IF BANK BELONGS TO CORRECT SIDE>>
  (3)                                  ;*****************************************************************************
  (3)                                  ;*SUBTEST       DETERMINE IF BANK BELONGS TO CORRECT SIDE
  (2)                                  ;*****************************************************************************
 2829                                          ;RETURN ERROR IF NOT!
 2830
 2831  015640  005037  001762                 CLR     CHECK
 2832  015644  013702  001710                 MOV     CSRFIRST,R2
 2833  015650  104475                         CB1CSR
 2834  015652                                  SUPERVISOR
  (1)  015652  052737  040000  177776         BIS     #BIT14,PSW              ;DO IT TO IT!
 2835  015660  104424                         CACHOFF
 2836  015662  012722  000000                 MOV     #0,(R2)+
 2837  015666  012712  000000                 MOV     #0,(R2)
 2838  015672  104503                         CLR1CSR
 2839  015674  005712                         TST     (R2)
 2840  015676  104426                         READCSR
 2841  015700  013701  001530                 MOV     CSR+2,R1
 2842
 2843                                          ;ELIMINATE DBE
 2844  015704  104471                         ECC1DIS
 2845  015706  013712  002006                 MOV     ZEROS,(R2)
 2846  015712  013742  002006                 MOV     ZEROS,-(R2)
 2847  015716  104503                         CLR1CSR
 2848  015720  104417                         KERNEL
 2849  015722  104423                         CACHON
 2850
 2851  015724                                  IF #BIT9 OFF.IN R1 AND SIDE EQ #'0 THEN $RETURN NOERROR
  (4)  015724  032701  001000                                                         BIT #BIT9,R1
  (6)  015730  001006                                                                 BNE L160
  (6)  015732  023727  002000  000060                                                 CMP SIDE,#'0
  (8)  015740  001002                                                                 BNE L160
  (6)  015742  000241                                                                 CLC
  (6)  015744  000207                                                                 RTS PC
  (5)  015746                                                                 L160:;;;;;;
 2852  015746                                  IF #BIT9 SET.IN R1 AND SIDE EQ #'1 THEN $RETURN NOERROR
  (4)  015746  032701  001000                                                         BIT #BIT9,R1
  (6)  015752  001406                                                                 BEQ L161
  (6)  015754  023727  002000  000061                                                 CMP SIDE,#'1
  (8)  015762  001002                                                                 BNE L161
  (6)  015764  000241                                                                 CLC
  (6)  015766  000207                                                                 RTS PC
  (5)  015770                                                                 L161:;;;;;;;
 2853  015770                                  $RETURN ERROR
  (2)  015770  000261                                                                 SEC
  (2)  015772  000207                                                                 RTS PC
```

```
015774                                  MKTEST: SUBTST  <<SUBR  MK11 TEST DISPATCH>>
                                        ;*******************************************************************
                                        ;*SUBTEST        SUBR    MK11 TEST DISPATCH
                                        ;*******************************************************************
015774                                          IF #SW0 SET.IN @SWR OR ACTFLAG NE #0
015774  032777  000001  164176                                                                  BIT #SW0,@SWR
016002  001003                                                                                  BNE L162
016004  005737  002020                                                                          TST ACTFLAG
016010  001402                                                                                  BEQ L163
016012                                                                                  L162:;;;;;;
016012  104470                                    ECCDIS
016014                                          ELSE
016014  000401                                                                                  BR L164
016016                                                                                  L163:;;;;;;
016016  104502                                    CLRCSR
016020                                          END ;OF IF
016020                                                                                  L164:;;;;;;
016020  012737  000002  001464                  MOV     #2,NOPAR                ;INDICATE PARITY ACTION
016026  012737  000002  001764                  MOV     #2,PCBUMP               ;TRAPS ADD 2 TO PC
016034  013700  001502                          MOV     PATTERN,R0              ;GET PATTERN NUMBER
016040  006300                                  ASL     R0                      ;MAKE IT A WORD ADDRESS
016042  004770  016056                          CALL    @MKPAT(R0)              ;INDEX OFF TABLE
016046  104472                                  ECCINIT
016050  005037  001464                          CLR     NOPAR                   ;INDICATE PARITY ACTION
016054  000207                                  RETURN

                                                ;WARNING IF YOU CHANGE THIS TABLE ALSO
                                                ;CHANGE "$DDW0" - "$DDW5" (THE PATTERN BIT MAP)

                                                ;PAT    TIME            DISCRIPTION
016056  017534                          MKPAT:  MT0017  ;<1 SEC         ;HOLDING 1'S & 0'S TEST
016060  017014                                  MT0007  ;<1 SEC         ;ADDRESS BIT TEST
016062  016346                                  MT0001  ;<1 SEC         ;ADDRESS TEST
016064  016406                                  MT0002  ;<1 SEC         ;COMPLEMENT ADDRESS TEST
016066  016616                                  MT0004  ; 1 SEC         ;ROTATING ZEROS TEST
016070  016700                                  MT0005  ; 1 SEC         ;ROTATING ONES TEST
016072  020212                                  MT0021  ; 1 SEC         ;MARCHING 0'S & 1'S TEST
016074  017556                                  MT0020  ;<1 SEC         ;MARCHING 1'S & 0'S IN CHECK BITS
016076  020334                                  MT0022  ; 9 SEC         ;REFRESH TEST
016100  020510                                  MT0026  ;<1 SEC         ;RANDOM DATA TEST
016102  020356                                  MT0023  ;69 SEC         ;SHIFTING DIAGONAL TEST
016104  020404                                  MT0024  ;58 SEC         ;FAST GALLOPING PATTERN TEST
016106  021430                                  MT0031  ;?? SEC         ;SOB-A-LONG TEST
016110  021532                                  MT0999  ; 0 SEC         ;NULL TEST
016112  021532                                  MT0999  ; 0 SEC         ;NULL TEST
016114  021532                                  MT0999  ; 0 SEC         ;NULL TEST
016116  021532                                  MT0999  ; 0 SEC         ;NULL TEST
016120  021532                                  MT0999  ; 0 SEC         ;NULL TEST
016122  021532                                  MT0999  ; 0 SEC         ;NULL TEST
016124  021532                                  MT0999  ; 0 SEC         ;NULL TEST
016126  021532                                  MT0999  ; 0 SEC         ;NULL TEST
016130  021532                                  MT0999  ; 0 SEC         ;NULL TEST
016132  021532                                  MT0999  ; 0 SEC         ;NULL TEST
016134  021532                                  MT0999  ; 0 SEC         ;NULL TEST
016136  021532                                  MT0999  ; 0 SEC         ;NULL TEST
016140  021532                                  MT0999  ; 0 SEC         ;NULL TEST
```

```
 2901  016142  021532                    MT0999  ; 0 SEC        ;NULL TEST
 2902  016144  021532                    MT0999  ; 0 SEC        ;NULL TEST
 2903  016146  021532                    MT0999  ; 0 SEC        ;NULL TEST
 2904  016150  021532                    MT0999  ; 0 SEC        ;NULL TEST
 2905  016152  021532                    MT0999  ; 0 SEC        ;NULL TEST
 2906  016154  021532                    MT0999  ; 0 SEC        ;NULL TEST
```

```
                                MJTEST: SUBTST  <<SUBR  MJ11 TEST DISPATCH>>
                                ;*****************************************************************************
                                ;*SUBTEST        SUBR    MJ11 TEST DISPATCH
                                ;*****************************************************************************
2909  016156  012737  000002  001464          MOV     #2,NOPAR                ;INDICATE PARITY ACTION
2910  016164  012737  000002  001764          MOV     #2,PCBUMP               ;TRAPS ADD 2 TO PC
2911  016172  013700  001502                  MOV     PATTERN,R0              ;GET PATTERN NUMBER
2912  016176  006300                          ASL     R0                      ;MAKE IT A WORD ADDRESS
2913  016200  004770  016212                  CALL    @MJPAT(R0)              ;INDEX OFF TABLE
2914  016204  005037  001464                  CLR     NOPAR                   ;INDICATE PARITY ACTION
2915  016210  000207                          RETURN
2916
2917                                          ;WARNING IF YOU CHANGE THIS TABLE ALSO
2918                                          ;CHANGE "$DDW0" - "$DDW5" (THE PATTERN BIT MAP)
2919
2920                                          ;PAT    TIME            DISCRIPTION
2921  016212  016346                  MJPAT:  MT0001  ;<1 SEC         ;ADDRESS TEST
2922  016214  016406                          MT0002  ;<1 SEC         ;COMPLEMENT ADDRESS TEST
2923  016216  016450                          MT0003  ; 1 SEC         ;3 XOR 9 WORST CASE NOISE TEST
2924  016220  016616                          MT0004  ; 1 SEC         ;ROTATING ZEROS TEST
2925  016222  016700                          MT0005  ; 1 SEC         ;ROTATING ONES TEST
2926  016224  020510                          MT0026  ;<1 SEC         ;RANDOM DATA TEST
2927  016226  021430                          MT0031  ;?? SEC         ;SOB-A-LONG TEST
2928  016230  021532                          MT0999  ; 0 SEC         ;NULL TEST
2929  016232  021532                          MT0999  ; 0 SEC         ;NULL TEST
2930  016234  021532                          MT0999  ; 0 SEC         ;NULL TEST
2931  016236  021532                          MT0999  ; 0 SEC         ;NULL TEST
2932  016240  021532                          MT0999  ; 0 SEC         ;NULL TEST
2933  016242  021532                          MT0999  ; 0 SEC         ;NULL TEST
2934  016244  021532                          MT0999  ; 0 SEC         ;NULL TEST
2935  016246  021532                          MT0999  ; 0 SEC         ;NULL TEST
2936  016250  021532                          MT0999  ; 0 SEC         ;NULL TEST
2937  016252  021532                          MT0999  ; 0 SEC         ;NULL TEST
2938  016254  021532                          MT0999  ; 0 SEC         ;NULL TEST
2939  016256  021532                          MT0999  ; 0 SEC         ;NULL TEST
2940  016260  021532                          MT0999  ; 0 SEC         ;NULL TEST
2941  016262  021532                          MT0999  ; 0 SEC         ;NULL TEST
2942  016264  021532                          MT0999  ; 0 SEC         ;NULL TEST
2943  016266  021532                          MT0999  ; 0 SEC         ;NULL TEST
2944  016270  021532                          MT0999  ; 0 SEC         ;NULL TEST
2945  016272  021532                          MT0999  ; 0 SEC         ;NULL TEST
2946  016274  021532                          MT0999  ; 0 SEC         ;NULL TEST
2947  016276  021532                          MT0999  ; 0 SEC         ;NULL TEST
2948  016300  021532                          MT0999  ; 0 SEC         ;NULL TEST
2949  016302  021532                          MT0999  ; 0 SEC         ;NULL TEST
2950  016304  021532                          MT0999  ; 0 SEC         ;NULL TEST
2951  016306  021532                          MT0999  ; 0 SEC         ;NULL TEST
2952  016310  021532                          MT0999  ; 0 SEC         ;NULL TEST
```

```
                                .TITLE  PATTERNS

                                .SBTTL  MEMORY TEST SETUP ROUTINES
                        MT0000: SUBTST  <<MT0000        SETUP DATA PATTERN TEST>>
                        ;*************************************************************************
                        ;*SUBTEST       MT0000  SETUP DATA PATTERN TEST
                        ;*************************************************************************
016312 005037 001744            CLR     REALPAT         ;SETUP PATTERN NUMBER FOR TYPEOUT & DISPLAY
016316 012700 060000            MOV     #FIRST,R0
016322 012701 040000            MOV     #SIZE,R1
016326 004737 030764            CALL    REGCOPY
016332                          BMOV    MTP000
016332 004537 037306                JSR R5,BLOCK1
016336 022102                      MTP000
016340 004737 021546            CALL    SUPD01                  ;DO IT IN SUPERVISOR MODE
016344 000207                   RETURN
016346                  MT0001: SUBTST  <<MT0001        SETUP ADDRESS TEST>>
                        ;*************************************************************************
                        ;*SUBTEST       MT0001  SETUP ADDRESS TEST
                        ;*************************************************************************
016346 012737 000001 001744     MOV     #1,REALPAT              ;SETUP PATTERN NUMBER FOR TYPEOUT & DISPLAY
016354 012700 060000            MOV     #FIRST,R0
016360 012701 040000            MOV     #SIZE,R1
016364 005002                   CLR     R2
016366 004737 030764            CALL    REGCOPY
016372                          BMOV    MTP001
016372 004537 037306                JSR R5,BLOCK1
016376 022126                      MTP001
016400 004737 021546            CALL    SUPD01                  ;DO IT IN SUPERVISOR MODE
016404 000207                   RETURN
016406                  MT0002: SUBTST  <<MT0002        SETUP COMPLEMENT ADDRESS TEST>>
                        ;*************************************************************************
                        ;*SUBTEST       MT0002  SETUP COMPLEMENT ADDRESS TEST
                        ;*************************************************************************
016406 012737 000002 001744     MOV     #2,REALPAT              ;SETUP PATTERN NUMBER FOR TYPEOUT & DISPLAY
016414 012700 160000            MOV     #LAST+2,R0
016420 012701 040000            MOV     #SIZE,R1
016424 012702 100001            MOV     #100001,R2
016430 004737 030764            CALL    REGCOPY
016434                          BMOV    MTP002
016434 004537 037306                JSR R5,BLOCK1
016440 022160                      MTP002
016442 004737 021546            CALL    SUPD01
016446 000207                   RETURN
```

```
 2984  016450                          MT0003: SUBTST  <<MT0003        SETUP 3 XOR 9 WORST CASE NOISE TEST>>
  (3)                                  ;*****************************************************************************
  (3)                                  ;*SUBTEST        MT0003  SETUP 3 XOR 9 WORST CASE NOISE TEST
  (2)                                  ;*****************************************************************************
 2985  016450  012737  000003  001744          MOV     #3,REALPAT              ;SETUP PATTERN NUMBER FOR TYPEOUT & DISPLAY
 2986  016456  005037  001764          CLR     PCBUMP                  ;TRAPS DO NOT ADD TO PC
 2987  016462  004737  030774  1$:     CALL    FLIPWARN                ;SETUP WARNING CONSTANTS & R2
 2988  016466  012701  060000  2$:     MOV     #FIRST,R1               ;R1 <-- STARTING ADDRESS
 2989  016472  005000                  CLR     R0                      ;INTERLEAVE ADDRESS CONSTANT
 2990  016474  160001                  SUB     R0,R1                   ;CORRECT FOR ADD BEFORE USE MODE
 2991  016476  012703  020000          MOV     #SIZE/2,R3              ;INTERLEAVE SIZE (8K,4K,2K,OR 1K (DOUBLE WORDS))
 2992  016502  072327  177770          ASH     #-8.,R3                 ;R3 <-- R3 / 256.
 2993  016506  012702  000004          MOV     #4,R2                   ;SMALL LOOP SIZE
 2994  016512  012705  000100          MOV     #64.,R5                 ;MEDIUM LOOP SIZE
 2995  016516                          BMOV    MTPA03,UDPDR7,17.
  (1)  016516  004537  037342              JSR R5,BLOCK3
  (1)  016522  000021                      17.
  (1)  016524  177636                      UDPDR7
  (1)  016526  022212                      MTPA03
 2996  016530  104415                  SAVREG
 2997  016532  012737  177636  001732          MOV     #FASTCITY-2,SUPDOADD
 2998  016540  004737  021724          CALL    SUPD03
 2999  016544                          BMOV    MTPB03
  (1)  016544  004537  037306              JSR R5,BLOCK1
  (1)  016550  022254                      MTPB03
 3000  016552                          BMOV    MTPC03,KDPAR0,8.
  (1)  016552  004537  037342              JSR R5,BLOCK3
  (1)  016556  000010                      8.
  (1)  016560  172360                      KDPAR0
  (1)  016562  022314                      MTPC03
 3001  016564                          BMOV    MTPD03,SDPAR0,8.
  (1)  016564  004537  037342              JSR R5,BLOCK3
  (1)  016570  000010                      8.
  (1)  016572  172260                      SDPAR0
  (1)  016574  022334                      MTPD03
 3002  016576  104416                  RESREG
 3003  016600  004737  021562          CALL    SUPD02
 3004  016604  022737  000003  002160          CMP     #3,FLIPLOC              ;DONE WITH 4 PATTERNS
 3005                                                                          ; [(0,177777);(177777,0);(401,177777);(177777,40
 3006  016612  001323                  BNE     1$                      ;NO - LOOP
 3007  016614  000207                  RETURN
 3008
 3009  016616                          MT0004: SUBTST  <<MT0004        SETUP ROTATING ZEROS TEST>>
  (3)                                  ;*****************************************************************************
  (3)                                  ;*SUBTEST        MT0004  SETUP ROTATING ZEROS TEST
  (2)                                  ;*****************************************************************************
 3010  016616  012737  000004  001744          MOV     #4,REALPAT              ;SETUP PATTERN NUMBER FOR TYPEOUT & DISPLAY
 3011  016624  012737  000004  001764          MOV     #4,PCBUMP               ;TRAPS ADD 4 TO PC
 3012  016632  013702  002154          MOV     ONES,R2
 3013  016636  004737  031124          CALL    BACKGND                 ;WRITE BACKGROUND OF ONES
 3014  016642  012700  060000          MOV     #FIRST,R0
 3015  016646  012701  040000          MOV     #SIZE,R1
 3016  016652                          BMOV    MTPA04
  (1)  016652  004537  037306              JSR R5,BLOCK1
  (1)  016656  022354                      MTPA04
 3017  016660                          BMOV    MTPB04,KDPAR0,8.
```

```
  (1)  016660  004537  037342                    JSR R5,BLOCK3
  (1)  016664  000010                            8.
  (1)  016666  172360                            KDPAR0
  (1)  016670  022410                            MTPB04
 3018  016672  004737  021562                    CALL    SUPD02
 3019  016676  000207                            RETURN
 3020  016700                            MT0005: SUBTST  <<MT0005        SETUP ROTATING ONES TEST>>
  (3)                                    ;*******************************************************************
  (3)                                    ;*SUBTEST       MT0005  SETUP ROTATING ONES TEST
  (2)                                    ;*******************************************************************
 3021  016700  012737  000005  001744            MOV     #5,REALPAT              ;SETUP PATTERN NUMBER FOR TYPEOUT & DISPLAY
 3022  016706  012737  000004  001764            MOV     #4,PCBUMP               ;TRAPS ADD 4 TO PC
 3023  016714  005002                            CLR     R2
 3024  016716  004737  031124                    CALL    BACKGND                 ;WRITE BACKGROUND OF ZEROS
 3025  016722  012700  060000                    MOV     #FIRST,R0
 3026  016726  012701  040000                    MOV     #SIZE,R1
 3027  016732                                    BMOV    MTP005
  (1)  016732  004537  037306                    JSR R5,BLOCK1
  (1)  016736  022430                            MTP005
 3028  016740                                    BMOV    MTPB04,KDPAR0,8.
  (1)  016740  004537  037342                    JSR R5,BLOCK3
  (1)  016744  000010                            8.
  (1)  016746  172360                            KDPAR0
  (1)  016750  022410                            MTPB04
 3029  016752  004737  021562                    CALL    SUPD02
 3030  016756  000207                            RETURN
 3031  016760                            MT0006: SUBTST  <<MT0006        SETUP INITIAL DATA TEST>>
  (3)                                    ;*******************************************************************
  (3)                                    ;*SUBTEST       MT0006  SETUP INITIAL DATA TEST
  (2)                                    ;*******************************************************************
 3032  016760  012737  000006  001744            MOV     #6,REALPAT              ;SETUP PATTERN NUMBER FOR TYPEOUT & DISPLAY
 3033  016766  012737  000004  001764            MOV     #4,PCBUMP               ;TRAPS ADD 4 TO PC
 3034  016774  013701  002070                    MOV     TESTADD,R1
 3035  017000  012737  022464  001732            MOV     #MTP006,SUPDOADD
 3036  017006  004737  021724                    CALL    SUPD03                  ;DO IT IN SUPERVISOR MODE
 3037  017012  000207                            RETURN
 3038  017014                            MT0007: SUBTST  <<MT0007        SETUP ADDRESS BIT TEST>>
  (3)                                    ;*******************************************************************
  (3)                                    ;*SUBTEST       MT0007  SETUP ADDRESS BIT TEST
  (2)                                    ;*******************************************************************
 3039  017014  012737  000007  001744            MOV     #7,REALPAT              ;SETUP PATTERN NUMBER FOR TYPEOUT & DISPLAY
 3040  017022  005002                            CLR     R2
 3041  017024  004737  031124                    CALL    BACKGND                 ;OF ZEROS
 3042  017030  012701  060000                    MOV     #FIRST,R1
 3043  017034  012702  000001                    MOV     #1,R2
 3044  017040  050201                            BIS     R2,R1
 3045  017042  012737  022654  001732            MOV     #MTP007,SUPDOADD
 3046  017050  004737  021724                    CALL    SUPD03                  ;DO IT IN SUPERVISOR MODE
 3047  017054  000207                            RETURN
```

```
 3049  017056                          MT0010: SUBTST  <<MT0010        SETUP BYTE ADDRESSING TEST>>
  (3)                                  ;*****************************************************************************
  (3)                                  ;*SUBTEST        MT0010  SETUP BYTE ADDRESSING TEST
  (2)                                  ;*****************************************************************************
 3050  017056  012737  000010  001744          MOV     #10,REALPAT             ;SETUP PATTERN NUMBER FOR TYPEOUT & DISPLAY
 3051  017064  012737  000004  001764          MOV     #4,PCBUMP               ;TRAPS ADD 4 TO PC
 3052  017072  013704  002070                  MOV     TESTADD,R4
 3053  017076  012737  022754  001732          MOV     #MTP010,SUPDOADD
 3054  017104  004737  021724                  CALL    SUPD03                  ;DO IT IN SUPERVISOR MODE
 3055  017110  000207                          RETURN
 3056  017112                          MT0011: SUBTST  <<MT0011        SETUP CREATE SINGLE BIT ERROR TEST>>
  (3)                                  ;*****************************************************************************
  (3)                                  ;*SUBTEST        MT0011  SETUP CREATE SINGLE BIT ERROR TEST
  (2)                                  ;*****************************************************************************
 3057  017112                                  IF NOECCFLAG NE #0 THEN $RETURN
  (4)  017112  005737  002056                                                    TST NOECCFLAG
  (6)  017116  001401                                                            BEQ L165
  (4)  017120  000207                                                            RTS PC
  (3)  017122                                                                 L165:;;;;;;
 3058  017122                                  IF ACTFLAG NE #0 OR APTFLAG NE #0
  (5)  017122  005737  002020                                                    TST ACTFLAG
  (5)  017126  001003                                                            BNE L166
  (6)  017130  005737  002022                                                    TST APTFLAG
  (8)  017134  001404                                                            BEQ L167
  (4)  017136                                                                 L166:;;;;;;
 3059  017136                                    IF $PASS NE #0 THEN $RETURN
  (4)  017136  005737  060172                                                    TST $PASS
  (6)  017142  001401                                                            BEQ L170
  (4)  017144  000207                                                            RTS PC
  (3)  017146                                                                 L170:;;;;;;
 3060  017146                                  END ;OF IF ACTFLAG
  (3)  017146                                                                 L167:;;;;;;
 3061  017146  012737  000011  001744          MOV     #11,REALPAT             ;SETUP PATTERN NUMBER FOR TYPEOUT & DISPLAY
 3062  017154  012737  023062  001732          MOV     #MTP011,SUPDOADD
 3063  017162  004737  021724                  CALL    SUPD03                  ;DO IT IN SUPERVISOR MODE
 3064  017166  000207                          RETURN
 3065  017170                          MT0012: SUBTST  <<MT0012        SETUP WRITE BYTE CLEARS SBE TEST>>
  (3)                                  ;*****************************************************************************
  (3)                                  ;*SUBTEST        MT0012  SETUP WRITE BYTE CLEARS SBE TEST
  (2)                                  ;*****************************************************************************
 3066  017170                                  IF NOECCFLAG NE #0 THEN $RETURN
  (4)  017170  005737  002056                                                    TST NOECCFLAG
  (6)  017174  001401                                                            BEQ L171
  (4)  017176  000207                                                            RTS PC
  (3)  017200                                                                 L171:;;;;;;
 3067  017200                                  IF ACTFLAG NE #0 OR APTFLAG NE #0
  (5)  017200  005737  002020                                                    TST ACTFLAG
  (5)  017204  001003                                                            BNE L172
  (6)  017206  005737  002022                                                    TST APTFLAG
  (8)  017212  001404                                                            BEQ L173
  (4)  017214                                                                 L172:;;;;;;
 3068  017214                                    IF $PASS NE #0 THEN $RETURN
  (4)  017214  005737  060172                                                    TST $PASS
  (6)  017220  001401                                                            BEQ L174
  (4)  017222  000207                                                            RTS PC
  (3)  017224                                                                 L174:;;;;;;
```

```
                                        END ;OF IF ACTFLAG
                                                                                L173:;;;;;;;
3069  017224
 (3)  017224
3070  017224  012737  000012  001744    MOV     #12,REALPAT             ;SETUP PATTERN NUMBER FOR TYPEOUT & DISPLAY
3071  017232  012737  023472  001732    MOV     #MTP012,SUPDOADD
3072  017240  004737  021724            CALL    SUPD03                  ;DO IT IN SUPERVISOR MODE
3073  017244  000207                    RETURN
3074  017246                    MT0013: SUBTST  <<MT0013        SETUP CREATE DOUBLE BIT ERROR TEST>>
 (3)                                    ;*****************************************************************************
 (3)                                    ;*SUBTEST       MT0013  SETUP CREATE DOUBLE BIT ERROR TEST
 (2)                                    ;*****************************************************************************
3075  017246                            IF ACTFLAG NE #0 OR APTFLAG NE #0
 (5)  017246  005737  002020                                                            TST ACTFLAG
 (5)  017252  001003                                                                    BNE L175
 (6)  017254  005737  002022                                                            TST APTFLAG
 (8)  017260  001404                                                                    BEQ L176
 (4)  017262                                                                    L175:;;;;;;;
3076  017262                              IF $PASS NE #0 THEN $RETURN
 (4)  017262  005737  060172                                                            TST $PASS
 (6)  017266  001401                                                                    BEQ L177
 (4)  017270  000207                                                                    RTS PC
 (3)  017272                                                                    L177:;;;;;;;
3077  017272                                  END ;OF IF ACTFLAG
 (3)  017272                                                                    L176:;;;;;;;
3078  017272  012737  000013  001744    MOV     #13,REALPAT             ;SETUP PATTERN NUMBER FOR TYPEOUT & DISPLAY
3079  017300  012737  024050  001732    MOV     #MTP013,SUPDOADD
3080  017306  012737  000003  001464    MOV     #3,NOPAR                ;INDICATE PARITY ACRION
3081  017314  004737  021724            CALL    SUPD03                  ;DO IT IN SUPERVISOR MODE
3082  017320  000207                    RETURN
3083  017322                    MT0014: SUBTST  <<MT0014        SETUP WRITE INHIBIT DURING DATIP WITH DBE>>
 (3)                                    ;*****************************************************************************
 (3)                                    ;*SUBTEST       MT0014  SETUP WRITE INHIBIT DURING DATIP WITH DBE
 (2)                                    ;*****************************************************************************
3084  017322                            IF NOECCFLAG NE #0 THEN $RETURN
 (4)  017322  005737  002056                                                            TST NOECCFLAG
 (6)  017326  001401                                                                    BEQ L200
 (4)  017330  000207                                                                    RTS PC
 (3)  017332                                                                    L200:;;;;;;;
3085  017332                            IF ACTFLAG NE #0 OR APTFLAG NE #0
 (5)  017332  005737  002020                                                            TST ACTFLAG
 (5)  017336  001003                                                                    BNE L201
 (6)  017340  005737  002022                                                            TST APTFLAG
 (8)  017344  001404                                                                    BEQ L202
 (4)  017346                                                                    L201:;;;;;;;
3086  017346                              IF $PASS NE #0 THEN $RETURN
 (4)  017346  005737  060172                                                            TST $PASS
 (6)  017352  001401                                                                    BEQ L203
 (4)  017354  000207                                                                    RTS PC
 (3)  017356                                                                    L203:;;;;;;;
3087  017356                                  END ;OF IF ACTFLAG
 (3)  017356                                                                    L202:;;;;;;;
3088  017356  012737  000014  001744    MOV     #14,REALPAT             ;SETUP PATTERN NUMBER FOR TYPEOUT & DISPLAY
3089  017364  012737  024422  001732    MOV     #MTP014,SUPDOADD
3090  017372  004737  021724            CALL    SUPD03                  ;DO IT IN SUPERVISOR MODE
3091  017376  000207                    RETURN
```

```
                                MT0014  SETUP WRITE INHIBIT DURING DATIP WITH DBE                                    SEQ 0200

                        MT0015: SUBTST  <<MT0015        SETUP WRITE INHIBIT OF BYTE WITH DBE>>
                        ;*****************************************************************************
                        ;*SUBTEST       MT0015  SETUP WRITE INHIBIT OF BYTE WITH DBE
                        ;*****************************************************************************
                                IF NOECCFLAG NE #0 THEN $RETURN
017400  005737  002056                                                          TST NOECCFLAG
017404  001401                                                                  BEQ L204
017406  000207                                                                  RTS PC
                                                                        L204:::::::
                                IF ACTFLAG NE #0 OR APTFLAG NE #0
017410  005737  002020                                                          TST ACTFLAG
017414  001003                                                                  BNE L205
017416  005737  002022                                                          TST APTFLAG
017422  001404                                                                  BEQ L206
                                                                        L205:::::::
                                  IF $PASS NE #0 THEN $RETURN
017424  005737  060172                                                          TST $PASS
017430  001401                                                                  BEQ L207
017432  000207                                                                  RTS PC
                                                                        L207:::::::
                                END ;OF IF ACTFLAG
                                                                        L206:::::::
017434  012737  000015  001744        MOV     #15,REALPAT             ;SETUP PATTERN NUMBER FOR TYPEOUT & DISPLAY
017442  012737  025016  001732        MOV     #MTP015,SUPDOADD
017450  004737  021724                CALL    SUPD03                  ;DO IT IN SUPERVISOR MODE
017454  000207                        RETURN
017456                  MT0016: SUBTST  <<MT0016        SETUP WRITE INHIBIT OF WORD WITH DBE>>
                        ;*****************************************************************************
                        ;*SUBTEST       MT0016  SETUP WRITE INHIBIT OF WORD WITH DBE
                        ;*****************************************************************************
                                IF NOECCFLAG NE #0 THEN $RETURN
017456  005737  002056                                                          TST NOECCFLAG
017462  001401                                                                  BEQ L210
017464  000207                                                                  RTS PC
                                                                        L210:::::::
                                IF ACTFLAG NE #0 OR APTFLAG NE #0
017466  005737  002020                                                          TST ACTFLAG
017472  001003                                                                  BNE L211
017474  005737  002022                                                          TST APTFLAG
017500  001404                                                                  BEQ L212
                                                                        L211:::::::
                                  IF $PASS NE #0 THEN $RETURN
017502  005737  060172                                                          TST $PASS
017506  001401                                                                  BEQ L213
017510  000207                                                                  RTS PC
                                                                        L213:::::::
                                END ;OF IF ACTFLAG
                                                                        L212:::::::
017512  012737  000016  001744        MOV     #16,REALPAT             ;SETUP PATTERN NUMBER FOR TYPEOUT & DISPLAY
017520  012737  025404  001732        MOV     #MTP016,SUPDOADD
017526  004737  021724                CALL    SUPD03                  ;DO IT IN SUPERVISOR MODE
017532  000207                        RETURN
017534                  MT0017: SUBTST  <<MT0017        SETUP HOLDING 1'S & 0'S>>
                        ;*****************************************************************************
                        ;*SUBTEST       MT0017  SETUP HOLDING 1'S & 0'S
                        ;*****************************************************************************
```

```
017534  012737  000017  001744          MOV     #17,REALPAT                 ;SETUP PATTERN NUMBER FOR TYPEOUT & DISPLAY
017542  012737  026006  001732          MOV     #MTP017,SUPDOADD
017550  004737  021724                  CALL    SUPD03                      ;DO IT IN SUPERVISOR MODE
017554  000207                          RETURN
017556                          MT0020: SUBTST  <<MT0020        SETUP MARCHING 0'S & 1'S IN CHECKBITS TEST>>
                                ;*****************************************************************************
                                ;*SUBTEST        MT0020  SETUP MARCHING 0'S & 1'S IN CHECKBITS TEST
                                ;*****************************************************************************
017556                                  IF ACTFLAG NE #0 OR APTFLAG NE #0
017556  005737  002020                                                                  TST ACTFLAG
017562  001003                                                                          BNE L214
017564  005737  002022                                                                  TST APTFLAG
017570  001404                                                                          BEQ L215
017572                                                                          L214:;;;;;;
017572                                    IF $PASS NE #0 THEN $RETURN
017572  005737  060172                                                                  TST $PASS
017576  001401                                                                          BEQ L216
017600  000207                                                                          RTS PC
017602                                                                          L216:;;;;;;
017602                                  END ;OF IF ACTFLAG
017602                                                                          L215:;;;;;;
017602  012737  000020  001744          MOV     #20,REALPAT                 ;SETUP PATTERN NUMBER FOR TYPEOUT & DISPLAY
017610  005037  001464                  CLR     NOPAR               ;INDICATE PARITY ACTION
017614  012737  026276  001732          MOV     #MTPA20,SUPDOADD
017622  013700  001706                  MOV     MKCSRS,R0
017626                                  FOR MTV020 := #0 TO #34 BY #4
017626  005037  017720                                                                  CLR MTV020
017632                                                                          B26:;;;;;;
017632  013737  017720  001634            MOV   MTV020,CSRNO
017640  106300                            ASLB  R0
017642                                    ON.ERROR
017642  103016                                                                          BCC L217
017644  112737  000060  002000              MOVB        #'0,SIDE
017652  004737  014344                      CALL        INTKRAP
017656                                      IF SPECIAL EQ #0
017656  005737  002010                                                                  TST SPECIAL
017662  001002                                                                          BNE L220
017664  006237  001772                        ASR       CSRINC
017670                                      END ;OF IF SPECIAL
017670                                                                          L220:;;;;;;
                                            ;NOW CSRFIRST & CSRINC ARE SETUP
017670                                      PUSH        R0
017670  010046                                                                          MOV R0,-(SP)
017672  004737  017722                      CALL        MT020Z
017676                                      POP         R0
017676  012600                                                                          MOV (SP)+,R0
017700                                    END ;OF ON.ERROR
017700                                                                          L217:;;;;;;
017700                                  END ;OF FOR CSRNO
017700  062737  000004  017720                                                          ADD #4,MTV020
017706  023727  017720  000034                                                          CMP MTV020,#34
017714  003746                                                                          BLE B26
017716                                                                          E26:;;;;;;;
017716  000207                          RETURN
017720  000000                  MTV020: 0                           ;VARIABLE FOR PAT 20
```

```
 3142  017722  013700  001710          MT020Z: MOV     CSRFIRST,R0
 3143  017726  010001                          MOV     R0,R1
 3144  017730  013702  001772                  MOV     CSRINC,R2
 3145  017734  005003                          CLR     R3
 3146  017736  012704  100000                  MOV     #BIT15,R4
 3147  017742  012705  160000                  MOV     #LAST+2,R5
 3148  017746  104503                          CLR1CSR
 3149  017750                                  BMOV    MTP20A
   (1) 017750  004537  037306                      JSR R5,BLOCK1
   (1) 017754  026064                              MTP20A
 3150  017756  004737  021546                  CALL    SUPD01
 3151  017762                                  IF #SW11 SET.IN @SWR OR QVFLAG IS TRUE
   (5) 017762  032777  004000  162210                                          BIT #SW11,@SWR
   (5) 017770  001003                                                          BNE L221
   (7) 017772  005737  002016                                                  TST QVFLAG
   (9) 017776  001401                                                          BEQ L222
   (4) 020000                                                          L221:::::::
 3152  020000                                    GOTO  MT020Y
   (2) 020000  000434                                                        BR MT020Y
 3153  020002                                  END ;OF IF #SW11
   (3) 020002                                                          L222:::::::
 3154  020002                                  BMOV    MTP20B
   (1) 020002  004537  037306                      JSR R5,BLOCK1
   (1) 020006  026102                              MTP20B
 3155  020010  004737  021562                  CALL    SUPD02
 3156  020014                                  IF #SW0 SET.IN @SWR OR ACTFLAG IS TRUE
   (5) 020014  032777  000001  162156                                          BIT #SW0,@SWR
   (5) 020022  001003                                                          BNE L223
   (7) 020024  005737  002020                                                  TST ACTFLAG
   (9) 020030  001402                                                          BEQ L224
   (4) 020032                                                          L223:::::::
 3157  020032  104477                            WAS1SBE
 3158  020034                                    ON.ERROR THEN GOTO MT020X
   (7) 020034  103463                                                        BCS MT020X
 3159  020036                                  END ;OF IF #SW0
   (3) 020036                                                          L224:::::::
 3160  020036                                  BMOV    MTP20C
   (1) 020036  004537  037306                      JSR R5,BLOCK1
   (1) 020042  026134                              MTP20C
 3161  020044  004737  021562                  CALL    SUPD02
 3162  020050                                  IF #SW0 SET.IN @SWR OR ACTFLAG IS TRUE
   (5) 020050  032777  000001  162122                                          BIT #SW0,@SWR
   (5) 020056  001003                                                          BNE L225
   (7) 020060  005737  002020                                                  TST ACTFLAG
   (9) 020064  001402                                                          BEQ L226
   (4) 020066                                                          L225:::::::
 3163  020066  104477                            WAS1SBE
 3164  020070                                    ON.ERROR THEN GOTO MT020X
   (7) 020070  103445                                                        BCS MT020X
 3165  020072                                  END ;OF IF #SW0
   (3) 020072                                                          L226:::::::
 3166  020072  013701  001710          MT020Y: MOV     CSRFIRST,R1
 3167  020076                                  BMOV    MTP20D
   (1) 020076  004537  037306                      JSR R5,BLOCK1
   (1) 020102  026166                              MTP20D
 3168  020104  004737  021562                  CALL    SUPD02
```

```
3169  020110                                  IF #SW0 SET.IN @SWR OR ACTFLAG IS TRUE
 (5)  020110  032777  000001  162062                                                  BIT #SW0,@SWR
 (5)  020116  001003                                                                  BNE L227
 (7)  020120  005737  002020                                                          TST ACTFLAG
 (9)  020124  001402                                                                  BEQ L230
 (4)  020126                                                                      L227:;;;;;;
3170  020126  104477                            WAS1SBE
3171  020130                                    ON.ERROR THEN GOTO MT020X
 (7)  020130  103425                                                                  BCS MT020X
3172  020132                                  END ;OF IF #SW0
 (3)  020132                                                                      L230:;;;;;;
3173  020132                                  BMOV    MTP20E
 (1)  020132  004537  037306                      JSR R5,BLOCK1
 (1)  020136  026220                              MTP20E
3174  020140  004737  021562                  CALL    SUPD02
3175  020144                                  IF #SW0 SET.IN @SWR OR ACTFLAG IS TRUE
 (5)  020144  032777  000001  162026                                                  BIT #SW0,@SWR
 (5)  020152  001003                                                                  BNE L231
 (7)  020154  005737  002020                                                          TST ACTFLAG
 (9)  020160  001402                                                                  BEQ L232
 (4)  020162                                                                      L231:;;;;;;
3176  020162  104477                            WAS1SBE
3177  020164                                    ON.ERROR THEN GOTO MT020X
 (7)  020164  103407                                                                  BCS MT020X
3178  020166                                  END ;OF IF #SW0
 (3)  020166                                                                      L232:;;;;;;
3179  020166                                  BMOV    MTP20F
 (1)  020166  004537  037306                      JSR R5,BLOCK1
 (1)  020172  026252                              MTP20F
3180  020174  004737  021562                  CALL    SUPD02
3181  020200  104503                          CLR1CSR
3182  020202  000207                          RETURN
3183
3184  020204  004737  021740          MT020X: CALL    SUPD04
3185  020210  000207                          RETURN
```

```
3187  020212                          MT0021: SUBTST  <<MT0021        SETUP MARCHING 0'S & 1'S TEST>>
 (3)                                  ;*****************************************************************************
 (3)                                  ;*SUBTEST        MT0021  SETUP MARCHING 0'S & 1'S TEST
 (2)                                  ;*****************************************************************************
3188  020212                                  SET NOSCOPE
 (1)  020212  012737  177777  002124        MOV     #-1,NOSCOPE
3189  020220  012737  000021  001744        MOV     #21,REALPAT             ;SETUP PATTERN NUMBER FOR TYPEOUT & DISPLAY
3190  020226  013702  002174                MOV     BAKPAT,R2
3191  020232  004737  031124                CALL    BACKGND
3192  020236  010203                        MOV     R2,R3
3193  020240  000303                        SWAB    R3
3194  020242  012701  160000                MOV     #LAST+2,R1
3195  020246  010105                        MOV     R1,R5
3196  020250  012704  060000                MOV     #FIRST,R4
3197  020254                                BMOV    MTPA21
 (1)  020254  004537  037306                    JSR R5,BLOCK1
 (1)  020260  026724                        MTPA21
3198  020262  004737  021546                CALL    SUPD01
3199
3200  020266                                BMOV    MTPB21
 (1)  020266  004537  037306                    JSR R5,BLOCK1
 (1)  020272  026754                        MTPB21
3201  020274  004737  021562                CALL    SUPD02
3202
3203  020300  010401                        MOV     R4,R1
3204  020302                                BMOV    MTPC21
 (1)  020302  004537  037306                    JSR R5,BLOCK1
 (1)  020306  027010                        MTPC21
3205  020310  004737  021562                CALL    SUPD02
3206
3207  020314                                BMOV    MTPD21
 (1)  020314  004537  037306                    JSR R5,BLOCK1
 (1)  020320  027044                        MTPD21
3208  020322  004737  021562                CALL    SUPD02
3209  020326  005037  002124                CLR     NOSCOPE
3210  020332  000207                        RETURN
3211
3212  020334                          MT0022: SUBTST  <<MT0022        SETUP REFRESH TEST>>
 (3)                                  ;*****************************************************************************
 (3)                                  ;*SUBTEST        MT0022  SETUP REFRESH TEST
 (2)                                  ;*****************************************************************************
3213  020334  012737  000022  001744        MOV     #22,REALPAT             ;SETUP PATTERN NUMBER FOR TYPEOUT & DISPLAY
3214  020342  012737  027074  001732        MOV     #MTP022,SUPDOADD
3215  020350  004737  021724                CALL    SUPD03                  ;DO IT IN SUPERVISOR MODE
3216  020354  000207                        RETURN
3217  020356                          MT0023: SUBTST  <<MT0023        SETUP SHIFTING DIAGONAL PATTERN TEST>>
 (3)                                  ;*****************************************************************************
 (3)                                  ;*SUBTEST        MT0023  SETUP SHIFTING DIAGONAL PATTERN TEST
 (2)                                  ;*****************************************************************************
3218  020356  012737  000023  001744        MOV     #23,REALPAT             ;SETUP PATTERN NUMBER FOR TYPEOUT & DISPLAY
3219  020364  012704  060000                MOV     #FIRST,R4
3220  020370  012737  027542  001732        MOV     #MTP023,SUPDOADD
3221  020376  004737  021724                CALL    SUPD03                  ;DO IT IN SUPERVISOR MODE
3222  020402  000207                        RETURN
```

```
                                MT0024: SUBTST  <<MT0024         SETUP FAST GALLOPING PATTERN TEST>>
                                ;*****************************************************************************
                                ;*SUBTEST        MT0024  SETUP FAST GALLOPING PATTERN TEST
                                ;*****************************************************************************
3225  020404  012737  000024  001744          MOV     #24,REALPAT             ;SETUP PATTERN NUMBER FOR TYPEOUT & DISPLAY
3226  020412  012737  000004  001764          MOV     #4,PCBUMP               ;TRAPS ADD 4 TO PC
3227  020420  012702  060000                  MOV     #FIRST,R2
3228  020424  010203                          MOV     R2,R3
3229  020426  012737  027750  001732          MOV     #MTP024,SUPDOADD
3230  020434  004737  021724                  CALL    SUPD03                  ;DO IT IN SUPERVISOR MODE
3231  020440  000207                          RETURN
3232  020442                        MT0025: SUBTST  <<MT0025         SETUP INTERRUPT ENABLE TEST>>
                                ;*****************************************************************************
                                ;*SUBTEST        MT0025  SETUP INTERRUPT ENABLE TEST
                                ;*****************************************************************************
3233  020442                          IF ACTFLAG NE #0 OR APTFLAG NE #0
(5)   020442  005737  002020                                                  TST ACTFLAG
(5)   020446  001003                                                          BNE L233
(6)   020450  005737  002022                                                  TST APTFLAG
(8)   020454  001404                                                          BEQ L234
(4)   020456                                                        L233:;;;;;;;
3234  020456                            IF $PASS NE #0 THEN $RETURN
(4)   020456  005737  060172                                                  TST $PASS
(6)   020462  001401                                                          BEQ L235
(4)   020464  000207                                                          RTS PC
(3)   020466                                                        L235:;;;;;;;
3235  020466                          END ;OF IF ACTFLAG
(3)   020466                                                        L234:;;;;;;;
3236  020466  012737  000025  001744          MOV     #25,REALPAT             ;SETUP PATTERN NUMBER FOR TYPEOUT & DISPLAY
3237  020474  012737  030144  001732          MOV     #MTP025,SUPDOADD
3238  020502  004737  021724                  CALL    SUPD03                  ;DO IT IN SUPERVISOR MODE
3239  020506  000207                          RETURN
3240  020510                        MT0026: SUBTST  <<MT0026         SETUP RANDOM DATA TEST>>
                                ;*****************************************************************************
                                ;*SUBTEST        MT0026  SETUP RANDOM DATA TEST
                                ;*****************************************************************************
3241  020510  012737  000026  001744          MOV     #26,REALPAT
3242  020516  005037  001764                  CLR     PCBUMP                  ;TRAPS DO NOT ADD TO THE PC
3243  020522  013703  002144                  MOV     SEEDLO,R3               ;INITIALIZE RANDOM NUMBERS
3244  020526  013702  002142                  MOV     SEEDHI,R2
3245  020532  010305                          MOV     R3,R5
3246  020534  010204                          MOV     R2,R4
3247  020536  012701  060000                  MOV     #FIRST,R1
3248  020542  012700  020000                  MOV     #SIZE/2,R0
3249  020546  104415                          SAVREG
3250  020550                                  BMOV    MTPA26                  ;WRITE ROUTINE TO FAST MEMORY
(1)   020550  004537  037306                      JSR R5,BLOCK1
(1)   020554  030570                              MTPA26
3251  020556                                  BMOV    MTPC26,KDPAR0,8.        ;RANDOM SUBPROGRAM TO FAST MEMORY
(1)   020556  004537  037342                      JSR R5,BLOCK3
(1)   020562  000010                              8.
(1)   020564  172360                              KDPAR0
(1)   020566  030640                              MTPC26
3252  020570                                  BMOV    MTPD26,SDPAR0,8.        ;RANDOM SUBSUBPROGRAM TO FAST MEMORY
(1)   020570  004537  037342                      JSR R5,BLOCK3
(1)   020574  000010                              8.
```

```
    (1)  020576  172260                        SDPARD
    (1)  020600  030660                        MTPD26
3253     020602  004737  021546          CALL    SUPDO1          ;WRITE RANDOM DATA
3254     020606  005037  030620          CLR     RANODD          ;FOR ERROR REPORTING
3255     020612                          BMOV    MTPB26          ;READ ROUTINE TO FAST MEMORY
    (1)  020612  004537  037306                JSR R5,BLOCK1
    (1)  020616  030604                        MTPB26
3256     020620  104416                  RESREG
3257     020622  004737  021546          CALL    SUPDO1          ;READ RANDOM DATA
3258     020626  010337  002144          MOV     R3,SEEDLO       ;UPDATE FOR NEW RANDOM NUMBERS
3259     020632  010237  002142          MOV     R2,SEEDHI
3260     020636  000207                  RETURN
```

```
3262  020640                        MT0027: SUBTST  <<MT0027        UNIQUE BANK TEST>>
 (3)                                ;*****************************************************************************
 (3)                                ;*SUBTEST       MT0027  UNIQUE BANK TEST
 (2)                                ;*****************************************************************************
3263  020640  012737  000027  001744          MOV     #27,REALPAT                 ;SETUP PATTERN NUMBER FOR TYPEOUT & DISPLAY
3264  020646  005037  001474                  CLR     MARGIN
3265  020652  005037  177750                  CLR     MAINT                       ;CLEAR MARGINS
3266  020656  104502                          CLRCSR
3267  020660                                  BMOV    MTPOOO
 (1)  020660  004537  037306                      JSR R5,BLOCK1
 (1)  020664  022102                              MTPOOO
3268  020666  012737  000207  177644  WARN7:  MOV     #207,UIPAR2                 ;PUT "RETURN" INSTRUCTION AFTER WRITE ROUTINE
3269  020674  012737  177646  001732          MOV     #UIPAR3,SUPDOADD
3270  020702                                  SET NOFSMODE
 (1)  020702  012737  177777  002110          MOV     #-1,NOFSMODE
3271  020710                                  FOR I := #1 TO #2
 (4)  020710  012737  000001  002130                                                      MOV #1,I
 (4)  020716                                                                      B27:;;;;;;;
3272  020716                                    FOR BANK := #0 TO #167
 (4)  020716  005037  001470                                                      CLR BANK
 (4)  020722                                                                      B30:;;;;;;
3273  020722  004737  036052                    CALL EXBANK
3274  020726                                    IF ACFLAG NE #0 AND RRFLAG EQ #0
 (4)  020726  005737  001506                                                      TST ACFLAG
 (5)  020732  001427                                                              BEQ L236
 (6)  020734  005737  001514                                                      TST RRFLAG
 (8)  020740  001024                                                              BNE L236
3275  020742                                      LET R2 := BANK
 (3)  020742  013702  001470                                                      MOV BANK,R2
3276  020746  012700  060000                      MOV       #FIRST,R0
3277  020752  010004                            MOV       R0,R4
3278  020754  012701  040000                    MOV       #SIZE,R1
3279  020760  010103                            MOV       R1,R3
3280  020762                                    IF I EQ #1
 (4)  020762  023727  002130  000001                                                CMP I,#1
 (6)  020770  001002                                                              BNE L237
3281  020772  004737  021546                      CALL    SUPD01
3282  020776                                    END ;OF IF
 (3)  020776                                                                    L237:;;;;;;
3283  020776                                    IF I EQ #2
 (4)  020776  023727  002130  000002                                                CMP I,#2
 (6)  021004  001002                                                              BNE L240
3284  021006  004737  021724                      CALL    SUPD03
3285  021012                                    END ;OF IF
 (3)  021012                                                                    L240:;;;;;;
3286  021012                                  END ;OF IF
 (3)  021012                                                                    L236:;;;;;;
3287  021012                                END ;OF FOR BANK
 (6)  021012  005237  001470                                                      INC BANK
 (5)  021016  023727  001470  000167                                              CMP BANK,#167
 (5)  021024  003736                                                              BLE B30
 (5)  021026                                                                    E30:;;;;;;;
3288  021026                              END ;OF FOR I
 (6)  021026  005237  002130                                                      INC I
 (5)  021032  023727  002130  000002                                              CMP I,#2
 (5)  021040  003726                                                              BLE B27
```

```
  (5)  021042
3289   021042
  (4)  021042  012737  000001  002130          FOR I := #1 TO #2
  (4)  021050
3290   021050
  (4)  021050  012737  000167  001470            FOR BANK := #167 TO #0
  (4)  021056
3291   021056  004737  036052                      CALL EXBANK
3292   021062                                      IF ACFLAG NE #0 AND RRFLAG EQ #0
  (4)  021062  005737  001506
  (6)  021066  001430
  (6)  021070  005737  001514
  (8)  021074  001025
3293   021076                                        LET R2 := BANK
  (3)  021076  013702  001470
3294   021102  005102                              COM       R2
3295   021104  012700  060000                      MOV       #FIRST,R0
3296   021110  010004                              MOV       R0,R4
3297   021112  012701  040000                      MOV       #SIZE,R1
3298   021116  010103                              MOV       R1,R3
3299   021120                                      IF I EQ #1
  (4)  021120  023727  002130  000001
  (6)  021126  001002
3300   021130  004737  021546                        CALL    SUPD01
3301   021134                                      END ;OF IF
  (3)  021134
3302   021134                                      IF I EQ #2
  (4)  021134  023727  002130  000002
  (6)  021142  001002
3303   021144  004737  021724                        CALL    SUPD03
3304   021150                                      END ;OF IF
  (3)  021150
3305   021150                                    END ;OF IF
  (3)  021150
3306   021150                                  END ;OF FOR BANK
  (6)  021150  005237  001470
  (5)  021154  023727  001470  000000
  (5)  021162  003735
  (5)  021164
3307   021164                                END ;OF FOR I
  (6)  021164  005237  002130
  (5)  021170  023727  002130  000002
  (5)  021176  003724
  (5)  021200
3308   021200  005037  002110                CLR       NOFSMODE
3309   021204  000207                        RETURN
```

```
E27:;;;;;;;;
        MOV #1,I
B31:;;;;;;;
        MOV #167,BANK
B32:;;;;;;;

        TST ACFLAG
        BEQ L241
        TST RRFLAG
        BNE L241

        MOV BANK,R2



        CMP I,#1
        BNE L242

L242:;;;;;;;
        CMP I,#2
        BNE L243

L243:;;;;;;;
L241:;;;;;;;
        INC BANK
        CMP BANK,#0
        BLE B32
E32:;;;;;;;;
        INC I
        CMP I,#2
        BLE B31
E31:;;;;;;;;
```

```
3311  021206                          MT0030: SUBTST  <<MT0030        SETUP FLUSH OUT DBE'S TEST>>
 (3)                                  ;*****************************************************************************
 (3)                                  ;*SUBTEST       MT0030  SETUP FLUSH OUT DBE'S TEST
 (2)                                  ;*****************************************************************************
3312  021206  005037  001734                  CLR     PASFLG
3313  021212  012737  000030  001744  MTA030: MOV     #30,REALPAT             ;SETUP PATTERN NUMBER FOR TYPEOUT & DISPLAY
3314  021220  005037  001474                  CLR     MARGIN
3315  021224  005037  177750                  CLR     MAINT                   ;CLEAR MARGINS
3316  021230  012737  000001  001464          MOV     #1,NOPAR                ;INDICATE COUNT PARITY ERRORS
3317  021236                                  BMOV    MTP030
 (1)  021236  004537  037306                      JSR R5,BLOCK1
 (1)  021242  030676                             MTP030
3318  021244  104470                          ECCDIS
3319  021246                                  SET     NOFSMODE
 (1)  021246  012737  177777  002110          MOV     #-1,NOFSMODE
3320  021254                                  FOR BANK := #0 TO #167
 (4)  021254  005037  001470                                                    CLR BANK
 (4)  021260                                                            B33:::::::
3321  021260  004737  036052                    CALL  EXBANK
3322  021264                                    IF MKFLAG NE #0
 (4)  021264  005737  001510                                                    TST MKFLAG
 (6)  021270  001414                                                            BEQ L244
3323  021272                                      IF ACFLAG NE #0 AND RRFLAG EQ #0
 (4)  021272  005737  001506                                                    TST ACFLAG
 (6)  021276  001411                                                            BEQ L245
 (6)  021300  005737  001514                                                    TST RRFLAG
 (8)  021304  001006                                                            BNE L245
3324  021306  012701  040000                      MOV       #SIZE,R1
3325  021312  012700  060000                      MOV       #FIRST,R0
3326  021316  004737  021546                      CALL      SUPD01
3327  021322                                    END ;OF IF ACFLAG
 (3)  021322                                                            L245:::::::
3328  021322                                    END ;OF IF MKFLAG
 (3)  021322                                                            L244:::::::
3329  021322                                  END ;OF FOR
 (6)  021322  005237  001470                                                    INC BANK
 (5)  021326  023727  001470  000167                                            CMP BANK,#167
 (5)  021334  003751                                                            BLE B33
 (5)  021336                                                            E33:::::::
3330  021336                                  IF PASFLG EQ #0
 (4)  021336  005737  001734                                                    TST PASFLG
 (6)  021342  001026                                                            BNE L246
3331  021344                                    SET PASFLG
 (1)  021344  012737  177777  001734          MOV     #-1,PASFLG
3332  021352  104502                            CLRCSR
3333  021354  004737  034720                    CALL  RELOCATE
3334  021360                                    ON.ERROR
 (8)  021360  103004                                                            BCC L247
3335  021362  104472                              ECCINIT
3336  021364  005037  002110                      CLR NOFSMODE
3337  021370  000207                              RETURN
3338  021372                                    END ;OF ON.ERROR
 (3)  021372                                                            L247:::::::
3339  021372  013727  001756  001470            MOV   NEWBANK,BANK
3340  021400  004737  036052                    CALL  EXBANK
3341  021404  004737  021212                    CALL  MTA030
```

```
 3342  021410  104472                    ECCINIT
 3343  021412  004737  035340            CALL  UNRELOCATE
 3344  021416  000207                    RETURN
 3345  021420                          END ;OF IF PASFLG
   (3) 021420                                                  L246::::::
 3346  021420  104472                  ECCINIT
 3347  021422  005037  002110          CLR     NOFSMODE
 3348  021426  000207                  RETURN
```

```
                                MT0031: SUBTST  <<MT0031        SETUP SOB-A-LONG TEST>>
                                ;*****************************************************************************
                                ;*SUBTEST        MT0031  SETUP SOB-A-LONG TEST
                                ;*****************************************************************************
3351  021430  012737  000031  001744          MOV     #31,REALPAT             ;SETUP PATTERN NUMBER FOR TYPEOUT & DISPLAY
3352  021436  005037  001464                  CLR     NOPAR                   ;SETUP PARITY ACTION
3353  021442                                  MAP     BANK                    ;MAP FIRST SO BLOCK MOVE WORKS
(3)   021442  010346                                                                    MOV R3,-(SP)
(1)   021444  013703  001470                  MOV     BANK,R3
(1)   021450  004737  034510                  CALL    MAPPER
(3)   021454  012603                                                                   MOV (SP)+,R3
3354  021456                                  SUPERVISOR
(1)   021456  052737  040000  177776          BIS     #BIT14,PSW              ;DO IT TO IT!
3355  021464                                  BMOV    MTP031,FIRST,SOBLENGTH/2
(1)   021464  004537  037342                      JSR R5,BLOCK3
(1)   021470  000027                              SOBLENGTH/2
(1)   021472  060000                              FIRST
(1)   021474  030706                              MTP031
3356  021476  104417                          KERNEL
3357  021500  013702  002136                  MOV     SOBK,R2
3358  021504  010200                          MOV     R2,R0
3359  021506  012701  100776                  MOV     #100776,R1              ;COMPLEMENT OF INSTRUCTION "SOB R0,.DOT"
3360  021512  012705  060056                  MOV     #FIRST+SOBLENGTH,R5
3361  021516  012737  060002  001732          MOV     #FIRST+2,SUPDOADD
3362  021524  004737  021740                  CALL    SUPD04
3363  021530  000207                          RETURN
3364  021532                                MT0999: SUBTST  <<MT0999        SETUP NULL TEST>>
                                ;*****************************************************************************
                                ;*SUBTEST        MT0999  SETUP NULL TEST
                                ;*****************************************************************************
3365  021532  005037  001744                  CLR     REALPAT
3366  021536                                  SET     NULLFLAG
(1)   021536  012737  177777  002014          MOV     #-1,NULLFLAG
3367  021544  000207                          RETURN
```

```
PATTERNS        MACY11 30A(1052)  28-FEB-78  09:20  PAGE 18                                                    SEQ 0212
CEMKAA.MA1    24-FEB-78 11:01          MT0999  SETUP NULL TEST

  3370  021546                          SUPD01: SUBTST  <<SUBR  EXECUTE PATTERN IN SUPERVISOR>>
   (3)                                  ;*****************************************************************************
   (3)                                  ;*SUBTEST        SUBR    EXECUTE PATTERN IN SUPERVISOR
   (2)                                  ;*****************************************************************************
  3371  021546                                  MAP     BANK
   (3)  021546  010346                                                                    MOV R3,-(SP)
   (1)  021550  013703  001470                    MOV     BANK,R3
   (1)  021554  004737  034510                    CALL    MAPPER
   (3)  021560  012603                                                                    MOV (SP)+,R3
  3372  021562  004737  050650          SUPD02: CALL    GETDIS
  3373  021566                                  PUSH    $LPERR,$LPADR
   (2)  021566  013746  002172                                                            MOV $LPERR,-(SP)
   (3)  021572  013746  002170                                                            MOV $LPADR,-(SP)
  3374  021576  010037  001644                    MOV     R0,SUPDR0
  3375  021602  012700  001646                    MOV     #SUPDR1,R0
  3376  021606  010120                            MOV     R1,(R0)+
  3377  021610  010220                            MOV     R2,(R0)+
  3378  021612  010320                            MOV     R3,(R0)+
  3379  021614  010420                            MOV     R4,(R0)+
  3380  021616  010520                            MOV     R5,(R0)+
  3381  021620  010620                            MOV     SP,(R0)+
  3382  021622  013700  001644                    MOV     SUPDR0,R0
  3383  021626  012737  021642  002170            MOV     #TAG4$,$LPADR
  3384  021634  013737  002170  002172            MOV     $LPADR,$LPERR
  3385  021642  012700  001662          TAG4$:  MOV     #SUPDR6+2,R0
  3386  021646  014006                            MOV     -(R0),SP
  3387  021650  014005                            MOV     -(R0),R5
  3388  021652  014004                            MOV     -(R0),R4
  3389  021654  014003                            MOV     -(R0),R3
  3390  021656  014002                            MOV     -(R0),R2
  3391  021660  014001                            MOV     -(R0),R1
  3392  021662  014000                            MOV     -(R0),R0
  3393  021664                                  SUPERVISOR
   (1)  021664  052737  040000  177776            BIS     #BIT14,PSW              ;DO IT TO IT!
  3394  021672  012706  000700                    MOV     #SUPSTK,SSP
  3395  021676  104424                            CACHOFF
  3396  021700  004737  177640                    CALL    FASTCITY
  3397  021704  104423                            CACHON
  3398  021706  104417                            KERNEL
  3399  021710  000004                            SCOPE
  3400  021712                                  POP     $LPADR,$LPERR
   (2)  021712  012637  002170                                                            MOV (SP)+,$LPADR
   (3)  021716  012637  002172                                                            MOV (SP)+,$LPERR
  3401  021722  000207                    RETURN
```

```
3404  021724                          SUPDO3: MAP     BANK
 (3)  021724  010346                                                                  MOV R3,-(SP)
 (1)  021726  013703  001470                  MOV     BANK,R3
 (1)  021732  004737  034510                  CALL    MAPPER
 (3)  021736  012603                                                                  MOV (SP)+,R3
3405  021740  004737  050650          SUPDO4: CALL    GETDIS
3406  021744                                  PUSH    $LPERR,$LPADR
 (2)  021744  013746  002172                                                          MOV $LPERR,-(SP)
 (3)  021750  013746  002170                                                          MOV $LPADR,-(SP)
3407  021754  010037  001644                  MOV     R0,SUPDR0
3408  021760  012700  001646                  MOV     #SUPDR1,R0
3409  021764  010120                          MOV     R1,(R0)+
3410  021766  010220                          MOV     R2,(R0)+
3411  021770  010320                          MOV     R3,(R0)+
3412  021772  010420                          MOV     R4,(R0)+
3413  021774  010520                          MOV     R5,(R0)+
3414  021776  010620                          MOV     SP,(R0)+
3415  022000  013700  001644                  MOV     SUPDR0,R0
3416  022004  012737  022020  002170          MOV     #TBG4$,$LPADR
3417  022012  013737  002170  002172          MOV     $LPADR,$LPERR
3418  022020  012700  001662          TBG4$:  MOV     #SUPDR6+2,R0
3419  022024  014006                          MOV     -(R0),SP
3420  022026  014005                          MOV     -(R0),R5
3421  022030  014004                          MOV     -(R0),R4
3422  022032  014003                          MOV     -(R0),R3
3423  022034  014002                          MOV     -(R0),R2
3424  022036  014001                          MOV     -(R0),R1
3425  022040  014000                          MOV     -(R0),R0
3426  022042                                  SUPERVISOR
 (1)  022042  052737  040000  177776          BIS     #BIT14,PSW              ;DO IT TO IT!
3427  022050  012706  000700                  MOV     #SUPSTK,SSP
3428  022054  104424                          CACHOFF
3429  022056  004777  157650                  CALL    @SUPDOADD
3430  022062  104423                          CACHON
3431  022064  104417                          KERNEL
3432  022066  000004                          SCOPE
3433  022070                                  POP     $LPADR,$LPERR
 (2)  022070  012637  002170                                                          MOV (SP)+,$LPADR
 (3)  022074  012637  002172                                                          MOV (SP)+,$LPERR
3434  022100  000207                          RETURN
```

```
                        .SBTTL  MEMORY TEST PATTERN ROUTINES
                ;*******************************************************************************
                ; PATTERN REGISTER CONVENTIONS
                ;       R0      FIRST ADDRESS OF PATTERN (FIRST,LAST+2,ETC)
                ;       R1      NUMBER OF ADDRESSES IN PATTERN (SIZE)
                ;       R2      DATA FOR PATTERN (ONES,52525,ETC)
                ;       R3      COPY OF R1 (IF NECESSARY)
                ;       R4      COPY OF R0 (IF NECESSARY)
                ;       R5      COPY OF R2 (IF NECESSARY)
                ;*******************************************************************************
022102          MTP000: SUBTST  <<MTP000        BASIC DATA TEST>>
                ;*******************************************************************************
                ;*SUBTEST       MTP000  BASIC DATA TEST
                ;*******************************************************************************
022102  010220  1$:     MOV     R2,(R0)+        ;V177640
022104  077102          SOB     R1,MTP000       ;V177642
022106  000240          NOP                     ;V177644
022110  012401  2$:     MOV     (R4)+,R1        ;V177646
022112  020102          CMP     R1,R2           ;V177650
022114  001402          BEQ     3$              ;V177652
022116  104430          PERR02                  ;V177654
022120  000240          NOP                     ;V177656
022122  077306  3$:     SOB     R3,2$           ;V177660
022124  000207          RETURN                  ;V177662
022126          MTP001: SUBTST  <<MTP001        ADDRESS TEST>>
                ;*******************************************************************************
                ;*SUBTEST       MTP001  ADDRESS TEST
                ;*******************************************************************************
022126  010220  3$:     MOV     R2,(R0)+        ;V177640
022130  062702  000002          ADD     #2,R2   ;V177642
022134  077104          SOB     R1,3$           ;V177646
022136  000240          NOP                     ;V177650
022140  012400  1$:     MOV     (R4)+,R0        ;V177652
022142  020005          CMP     R0,R5           ;V177654
022144  001401          BEQ     2$              ;V177656
022146  104427          PERR01                  ;V177660
022150  062705  000002  2$:     ADD     #2,R5   ;V177662
022154  077307          SOB     R3,1$           ;V177666
022156  000207          RETURN                  ;V177672
022160          MTP002: SUBTST  <<MTP002        COMPLEMENT ADDRESS TEST (WRITE DOWN, READ UP)>>
                ;*******************************************************************************
                ;*SUBTEST       MTP002  COMPLEMENT ADDRESS TEST (WRITE DOWN, READ UP)
                ;*******************************************************************************
022160  010244  3$:     MOV     R2,-(R4)        ;V177640
022162  062702  000002          ADD     #2,R2   ;V177642
022166  077104          SOB     R1,3$           ;V177646
022170  000240          NOP                     ;V177650
022172  162702  000002  1$:     SUB     #2,R2   ;V177652
022176  012401          MOV     (R4)+,R1        ;V177656
022200  020102          CMP     R1,R2           ;V177660
022202  001401          BEQ     2$              ;V177662
022204  104430          PERR02                  ;V177664
022206  077307  2$:     SOB     R3,1$           ;V177666
022210  000207          RETURN                  ;V177670
```

```
                                MTPA03: SUBTST  <<MTPA03        3 XOR 9 WORST CASE NOISE TEST (WRITE)>>
                                ;*****************************************************************************
                                ;*SUBTEST        MTPA03  3 XOR 9 WORST CASE NOISE TEST (WRITE)
                                ;*****************************************************************************
                                        ;R0 = INTERLEAVE FACTOR
                                        ;R1 = ADDRESS
                                        ;R2 = SMALL LOOP CONSTANT
                                        ;R3 = NUM OF ADD TO TEST (LARGE LOOP)
                                        ;R4 = GOOD DATA
                                        ;R5 = MEDIUM LOOP CONSTANT
                                        .ENABL  LSB
022212  060001          1$:    ADD     R0,R1           ;V177636 ADD INTERLEAVE FACTOR
022214  010421                 MOV     R4,(R1)+        ;V177640
022216  010421                 MOV     R4,(R1)+        ;V177642
022220  077204                 SOB     R2,1$           ;V177644
022222  005104                 COM     R4              ;V177646
022224  052704                 BIS     (PC)+,R4        ;V177650
022226  000401         WARN2:  401                     ;V177652 WARNING LOCATION IS MODIFIED BEFORE LOADING
022230  012702  000004         MOV     #4,R2           ;V177654
022234  077512                 SOB     R5,1$           ;V177660
022236  005104                 COM     R4              ;V177662
022240  052704                 BIS     (PC)+,R4        ;V177664
022242  000401         WARN3:  401                     ;V177666 WARNING LOCATION IS MODIFIED BEFORE LOADING
022244  012705  000100         MOV     #64.,R5         ;V177670
022250  077320                 SOB     R3,1$           ;V177674
022252  000207                 RETURN                  ;V177676
                                        .DSABL  LSB

022254                          MTPB03: SUBTST  <<MTPB03        3 XOR 9 WORST CASE NOISE TEST (READ)>>
                                ;*****************************************************************************
                                ;*SUBTEST        MTPB03  3 XOR 9 WORST CASE NOISE TEST (READ)
                                ;*****************************************************************************
                                        .ENABL  LSB
022254  000137  172360  1$:    JMP     KDPAR0          ;V177640        GO TO V172360
022260  077203                 SOB     R2,1$           ;V177644
022262  005104                 COM     R4              ;V177646
022264  052704                 BIS     (PC)+,R4        ;V177650
022266  000401         WARN4:  401                     ;V177652 WARNING LOCATION IS MODIFIED BEFORE LOADING
022270  012702  000004         MOV     #4,R2           ;V177654
022274  077511                 SOB     R5,1$           ;V177660
022276  005104                 COM     R4              ;V177662
022300  052704                 BIS     (PC)+,R4        ;V177664
022302  000401         WARN5:  401                     ;V177666 WARNING LOCATION IS MODIFIED BEFORE LOADING
022304  012705  000100         MOV     #64.,R5         ;V177670
022310  077317                 SOB     R3,1$           ;V177674
022312  000207                 RETURN                  ;V177676
                                        .DSABL  LSB
```

J01


```
  3526  022314                          MTPC03: SUBTST  <<MTPC03         TEST DATA SUBPROGRAM>>
   (3)                                  ;*****************************************************************************
   (3)                                  ;*SUBTEST        MTPC03  TEST DATA SUBPROGRAM
   (2)                                  ;*****************************************************************************
  3527  022314  060001                          ADD     R0,R1           ;V172360
  3528  022316  020421                          CMP     R4,(R1)+        ;V172362
  3529  022320  001401                          BEQ     1$              ;V172364
  3530  022322  104431                          PERR03                  ;V172366
  3531  022324  005141                  1$:     COM     -(R1)           ;V172370
  3532  022326  005111                          COM     (R1)            ;V172372
  3533  022330  000137  172260                  JMP     SDPAR0  ;V174372        GO TO V172260
  3534
  3535  022334                          MTPD03: SUBTST  <<MTPD03         TEST DATA SUBSUBPROGRAM>>
   (3)                                  ;*****************************************************************************
   (3)                                  ;*SUBTEST        MTPD03  TEST DATA SUBSUBPROGRAM
   (2)                                  ;*****************************************************************************
  3536  022334  020421                          CMP     R4,(R1)+        ;V172260
  3537  022336  001401                          BEQ     1$              ;V172262
  3538  022340  104431                          PERR03                  ;V172264
  3539  022342  005127                  1$:     COM     (PC)+           ;V172266
  3540  022344  000000                          0                       ;V172270
  3541  022346  001033                          BNE     .+70            ;V172272        GO TO V172362
  3542  022350  000137  177644                  JMP     UIPAR2  ;V172274        GO TO V177644
```

```
3544  022354                        MTPA04: SUBTST  <<MTPA04         ROTATING ZEROS TEST>>
 (3)                                ;*****************************************************************************
 (3)                                ;*SUBTEST       MTPA04  ROTATING ZEROS TEST
 (2)                                ;*****************************************************************************
3545  022354  012705  000010        1$:     MOV     #8.,R5          ;V177640
3546  022360  010504                        MOV     R5,R4           ;V177644
3547  022362  000241                        CLC                     ;V177646
3548  022364  000137  172360                JMP     KDPARO  ;V177650
3549  022370  016004  177776                MOV     -2(R0),R4       ;V177654
3550  022374  103402                        BCS     2$              ;V177660
3551  022376  020204                        CMP     R2,R4           ;V177662
3552  022400  001401                        BEQ     3$              ;V177664
3553  022402  104432                2$:     PERR04                  ;V177666
3554  022404  077115                3$:     SOB     R1,1$           ;V177670
3555  022406  000207                        RETURN                  ;V177672
3556
3557  022410                        MTPB04: SUBTST  <<MTPB04         SUBR    ROTATING BIT>>
 (3)                                ;*****************************************************************************
 (3)                                ;*SUBTEST       MTPB04  SUBR    ROTATING BIT
 (2)                                ;*****************************************************************************
3558  022410  106110                1$:     ROLB    (R0)            ;V172360
3559  022412  077502                        SOB     R5,1$           ;V172362
3560  022414  106120                        ROLB    (R0)+           ;V172364
3561  022416  106110                2$:     ROLB    (R0)            ;V172366
3562  022420  077402                        SOB     R4,2$           ;V172370
3563  022422  106120                        ROLB    (R0)+           ;V172372
3564  022424  000137  177654                JMP     UIPAR6          ;V172374
3565
3566  022430                        MTPD05: SUBTST  <<MTPD05         ROTATION ONES TEST>>
 (3)                                ;*****************************************************************************
 (3)                                ;*SUBTEST       MTPD05  ROTATION ONES TEST
 (2)                                ;*****************************************************************************
3567  022430  012705  000010        1$:     MOV     #8.,R5          ;V177640
3568  022434  010504                        MOV     R5,R4           ;V177644
3569  022436  000261                        SEC                     ;V177646
3570  022440  000137  172360                JMP     KDPARO  ;V177650
3571  022444  016004  177776                MOV     -2(R0),R4       ;V177654
3572  022450  103002                        BCC     2$              ;V177660 IF THIS HAPPENS THE GOOD & BAD MATCH
3573  022452  020204                        CMP     R2,R4           ;V177662
3574  022454  001401                        BEQ     3$              ;V177664
3575  022456  104432                2$:     PERR04                  ;V177666
3576  022460  077115                3$:     SOB     R1,1$           ;V177670
3577  022462  000207                        RETURN                  ;V177672
```

```
3580  022464                          MTP006: SUBTST  <<MTP006         INITIAL DATA TEST>>
 (3)                                  ;*****************************************************************************
 (3)                                  ;*SUBTEST       MTP006  INITIAL DATA TEST
 (2)                                  ;*****************************************************************************
3581                                                  ;       THIS TEST CHECKS THE DI/DO LINES BY
3582                                                  ;       SHIFTING A 1 THROUGH THE WORD.
3583  022464  012737  000001  001712          MOV     #1,DATBUF          ;SET THE FIRST TEST BIT
3584  022472  005037  001714                  CLR     DATBUF+2           ;CLEAR 2ND WORD
3585  022476  013711  001712          1$:     MOV     DATBUF,(R1)        ;WRITE TEST WORD 1
3586  022502  013761  001714  000002          MOV     DATBUF+2,2(R1)     ;AND TEST WORD 2
3587  022510  011102                          MOV     (R1),R2
3588  022512  023702  001712                  CMP     DATBUF,R2          ;NOW READ THEM
3589  022516  001401                          BEQ     2$                 ;BR IF FIRST 16 OK
3590  022520  104433                          PERR07                     ;ERROR TRAP
3591
3592  022522  016102  000002          2$:     MOV     2(R1),R2
3593  022526  023702  001714                  CMP     DATBUF+2,R2        ;NOW READ SECOND WORD
3594  022532  001401                          BEQ     3$                 ;BR IF OK
3595  022534  104434                          PERR10                     ;ERROR TRAP
3596
3597  022536  005737  001714          3$:     TST     DATBUF+2           ;HAS LAST BIT BEEN TESTED ?
3598  022542  100405                          BMI     4$                 ;MINUS MEANS BIT 31
3599  022544                                  DLEFT   DATBUF             ;NO, SHIFT TEST BIT LEFT
 (1)  022544  006137  001712                  ROL     DATBUF
 (1)  022550  006137  001714                  ROL     DATBUF+2
3600  022554  000750                          BR      1$
3601                                                                     ;NOW GOING TO SHIFT A 0 IN DATA DIRECTION
3602  022556  012737  177776  001712  4$:     MOV     #177776,DATBUF     ;PUT A 0 IN BIT 0
3603  022564  012737  177777  001714          MOV     #-1, DATBUF+2      ;AND 1'S IN ALL OTHERS
3604  022572  013711  001712          5$:     MOV     DATBUF,(R1)        ;WRITE THE DATA
3605  022576  013761  001714  000002          MOV     DATBUF+2,2(R1)     ;2 WORDS WORTH
3606  022604  011102                          MOV     (R1),R2
3607  022606  023702  001712                  CMP     DATBUF,R2          ;NOW READ FIRST WORD
3608  022612  001401                          BEQ     6$                 ;BR IF OK
3609  022614  104433                          PERR07
3610
3611  022616  016102  000002          6$:     MOV     2(R1),R2
3612  022622  023702  001714                  CMP     DATBUF+2,R2        ;NOW, READ SECOND WORD
3613  022626  001401                          BEQ     7$                 ;BR IF OK
3614  022630  104434                          PERR10
3615
3616  022632  005737  001714          7$:     TST     DATBUF+2           ;TESTED BIT 31 YET?
3617  022636  100005                          BPL     8$                 ;BR IF YES, WE'RE DONE
3618  022640                                  DLEFT   DATBUF
 (1)  022640  006137  001712                  ROL     DATBUF
 (1)  022644  006137  001714                  ROL     DATBUF+2
3619  022650  000750                          BR      5$                 ;KEEP GOING
3620  022652  000207                  8$:     RETURN
```

```
 3623  022654                          MTP007: SUBTST  <<MTP007        ADDRESS BIT TEST>>
  (3)                                  ;*****************************************************************************
  (3)                                  ;*SUBTEST       MTP007  ADDRESS BIT TEST
  (2)                                  ;*****************************************************************************
 3624                                                  ;       THIS TEST CHECKS TO SEE THAT EACH ADDRESS
 3625                                                  ;       BIT IN EACH 16K BANK CAN BE ASSERTED UNIQUELY.
 3626                                                  ;       IT CHECKS FOR ADDRESS BITS THAT MAY BE STUCK
 3627                                                  ;       HIGH, STUCK LOW OR STUCK TOGETHER.
 3628  022654  111100                          MOVB    (R1),R0
 3629  022656  105700                          TSTB    R0              ;READ AND COMPARE FOR ZEROS
 3630  022660  001401                          BEQ     1$              ;BR IF OK
 3631  022662  104435                          PERR11
 3632
 3633  022664  105111                  1$:     COMB    (R1)            ;COMPLEMENT THE BYTE
 3634  022666  111100                          MOVB    (R1),R0
 3635  022670  105700                          TSTB    R0              ;READ FOR NON ZEROS
 3636  022672  001001                          BNE     2$              ;BR IF OK
 3637  022674  104436                          PERR12
 3638
 3639  022676  040201                  2$:     BIC     R2,R1           ;MASK OFF THE ASSERTED BIT
 3640  022700  006302                          ASL     R2              ;SHIFT R2 FOR NEXT BIT
 3641  022702  050201                          BIS     R2,R1           ;SET THE NEW BIT INTO R1
 3642  022704  011100                          MOV     (R1),R0
 3643  022706  005700                          TST     R0              ;READ THE NEW ADDRESS
 3644  022710  001401                          BEQ     3$              ;READ FOR ZEROS
 3645  022712  104437                          PERR13
 3646
 3647  022714  005111                  3$:     COM     (R1)            ;COMPL THE WORD
 3648  022716  011100                          MOV     (R1),R0
 3649  022720  005700                          TST     R0              ;READ IT AGAIN
 3650  022722  001001                          BNE     4$
 3651  022724  104440                          PERR14
 3652
 3653  022726  022702  100000          4$:     CMP     #100000,R2
 3654  022732  001407                          BEQ     5$
 3655  022734  022702  010000                  CMP     #10000,R2       ;CHECK FOR MSB IN 4K BANK
 3656  022740  001356                          BNE     2$              ;NOT LAST BIT, BRANCH
 3657  022742  006302                          ASL     R2
 3658  022744  012701  160000                  MOV     #160000,R1
 3659  022750  000752                          BR      2$
 3660  022752  000207                  5$:     RETURN
```

```
                        MTP010: SUBTST  <<MTP010         BYTE ADDRESSING TEST>>
                        ;*****************************************************************************
                        ;*SUBTEST       MTP010  BYTE ADDRESSING TEST
                        ;*****************************************************************************
                                ;TEST 3 THIS TEST CHECKS FOR PROPER
                                ;       BYTE ADDRESSING WITH ECC DISABLED
022754 010402                   MOV     R4,R2           ;R4 HAS LOWEST ADDRESS
022756 010403                   MOV     R4,R3           ;PUT IT IN R3 ALSO
022760 062702 000004            ADD     #4,R2           ;POINT R2 TO LAST BYTE +1
022764 012713 177777            MOV     #-1,(R3)        ;WRITE ALL ONES IN
022770 012763 177777 000002     MOV     #-1,2(R3)       ;THE 4 TEST BYTES
022776 105013           1$:     CLRB    (R3)            ;CLEAR A BYTE
023000 010401                   MOV     R4,R1           ;INITIALIZE R1 FOR EACH PASS
023002 020201           2$:     CMP     R2,R1           ;IF EQUAL, JUST READ LAST BYTE
023004 001420                   BEQ     6$              ;BR IF EQUAL
023006 020301                   CMP     R3,R1           ;IS THIS THE BYTE OF ZERO
023010 001007                   BNE     4$              ;BR IF NOT
023012 111100                   MOVB    (R1),R0
                                ;WARNING IF YOU OPTOMIZE CHANGE THE PCBUMP FOR THIS ERROR INCASE OF TRAPS
023014 022700 000000            CMP     #0,R0           ;IT IS, COMPARE FOR ZEROS
023020 001401                   BEQ     3$
023022 104435                   PERR11
023024 005201           3$:     INC     R1              ;NEXT BYTE
023026 000765                   BR      2$              ;RETURN
023030 111100           4$:     MOVB    (R1),R0
023032 122700 177777            CMPB    #-1,R0          ;ITS NOT THE BYTE OF 0'S, READ 1'S
023036 001401                   BEQ     5$
023040 104436                   PERR12
023042 005201           5$:     INC     R1              ;MOVE TO NEXT BYTE
023044 000756                   BR      2$
023046 112713 177777    6$:     MOVB    #-1,(R3)        ;RESTORE 1'S TO BYTE JUST TESTED
023052 005203                   INC     R3              ;INC TO NEXT BYTE
023054 020302                   CMP     R3,R2           ;WAS THAT JUST THE LAST ONE?
023056 001347                   BNE     1$              ;BR IF NO
023060 000207                   RETURN
```

```
 3698  023062                          MTP011: SUBTST  <<MPT011        SINGLE BIT ERROR TEST>>
  (3)                                  ;*********************************************************************
  (3)                                  ;*SUBTEST       MPT011  SINGLE BIT ERROR TEST
  (2)                                  ;*********************************************************************
 3699                                          ;(1)    CREATE A SINGLE BIT ERROR
 3700                                          ;
 3701                                          ;(2)    READ BACK SBE UNCORRECTED (WITH ECC DISABLE)
 3702                                          ;
 3703                                          ;(3)    ENABLE ECC & READ CORRECTED DATA
 3704                                          ;
 3705                                          ;(4)    CHECK THAT THE SBE FLAG WAS SET FORM THE LAST READ
 3706                                          ;
 3707                                          ;(5)    DO (1-4) FOR DATA CONSISTING OF 1 BIT SET IN EACH OF 32
 3708                                          ;       POSITIONS OF A DOUBLE WORD
 3709                                          ;       THEN DO IT AGAIN FOR 1 BIT CLEARED IN EACH OF 32 POSITIONS OF
 3710                                          ;       A DOUBLE WORD
 3711                                          ;       IE (64 TIMES)
 3712                                          ;
 3713                                          ;(6)    DO (1-5) FOR A SBE IN EACH OF 32 BIT POSITIONS
 3714                                          ;       IE (RUN TEST 64 * 32 = 2048 TIMES)
 3715  023062  104503                         CLR1CSR
 3716                                         ;BIG LOOP
 3717  023064  012737  000001  001712  1$:    MOV     #1,DATBUF       ;INITIAL DATA
 3718  023072  005037  001714                 CLR     DATBUF+2        ;32 BITS WORTH
 3719                                         ;MEDIUM LOOP
 3720  023076  012737  000001  001722  2$:    MOV     #1,SBEMSK       ;INITIAL ERROR MASK
 3721  023104  005037  001724                 CLR     SBEMSK+2        ;32 BITS WORTH
 3722                                         ;LITTLE LOOP
 3723  023110  013737  001712  001716  3$:    MOV     DATBUF,TSTDAT   ;
 3724  023116  013737  001714  001720         MOV     DATBUF+2,TSTDAT+2;TO SAVE ORIG DATA
 3725  023124  105737  001734                 TSTB    PASFLG  ;COMP DATA ON SECOND PASSONLY
 3726  023130  001404                         BEQ     4$              ;BR IF FIRST PASS
 3727  023132  005137  001716                 COM     TSTDAT          ;SECOND PASS, COMP BOTH WORDS
 3728  023136  005137  001720                 COM     TSTDAT+2
 3729  023142  013702  001716          4$:    MOV     TSTDAT,R2
 3730  023146  013703  001720                 MOV     TSTDAT+2,R3
 3731  023152  012737  001716  001760         MOV     #TSTDAT,SOURCE  ;SET UP ADDRESS FOR CHKGEN
 3732  023160  004737  034350                 CALL    CHKGEN          ;GEN CHECKBITS ON TSTDAT
 3733                                         ;*******************************
 3734                                         ;** CREATE A SINGLE BIT ERROR **
 3735                                         ;*******************************
 3736  023164  013701  001722                 MOV     SBEMSK,R1
 3737  023170  074137  001716                 XOR     R1,TSTDAT
 3738  023174  013701  001724                 MOV     SBEMSK+2,R1
 3739  023200  074137  001720                 XOR     R1,TSTDAT+2
 3740  023204  013701  002070          5$:    MOV     TESTADD,R1      ;FIRST TEST ADDRESS
 3741  023210  104471                         ECC1DIS                 ;FORCE VALIDATE WITH ECC DISABLED
 3742  023212  013721  001716                 MOV     TSTDAT,(R1)+    ;WRITE FIRST 16 BITS
 3743  023216  104475                         CB1CSR
 3744  023220  013711  001720                 MOV     TSTDAT+2,(R1)   ;WRITE SECOND 16 BITS AND
 3745                                                                 ;CHECK BITS. WE NOW HAVE CHECKBITS
 3746                                                                 ;GENERATED ON DATBUF AND DATA WITH
 3747                                                                 ;ONE BIT IN ERROR (AS PER SBEMSK).
 3748  023224  104471                         ECC1DIS
 3749  023226  014100                         MOV     -(R1),R0
 3750  023230  020037  001716                 CMP     R0,TSTDAT       ;READ THE LOW WORD (UNCORRECTED)
```

```
3751  023234  001401                      BEQ     6$              ;BR IF OK
3752  023236  104455                      PERR31
3753
3754  023240  062701  000002      6$:     ADD     #2,R1           ;POINT R1 TO SECOND 16 BITS
3755  023244  011100                      MOV     (R1),R0
3756  023246  020037  001720              CMP     R0,TSTDAT+2     ;READ THE HIGH WORD (UNCORRECTED)
3757  023252  001401                      BEQ     7$              ;BR IF OK
3758  023254  104455                      PERR31
3759
3760  023256  104503              7$:     CLR1CSR
3761  023260  014100                      MOV     -(R1),R0
3762  023262  020002                      CMP     R0,R2           ;SEE IF ITS BEEN CORRECTED
3763  023264  001401                      BEQ     8$              ;IT SHOULD HAVE BEEN
3764  023266  104456                      PERR32
3765
3766  023270  104510              8$:     TSTREAD
3767  023272  103411                      BCS     9$              ;BR IF IT IS SET
3768  023274                              SET     HEADER          ;ENABLE PRINTING OF ERROR HEADER INFO
 (1)  023274  012737  177777  002146      MOV     #-1,HEADER
3769  023302  010137  001422              MOV     R1,ADDRESS
3770  023306  104016                      ERROR   +16
3771  023310                              SET     HEADER          ;ENABLE PRINTING OF ERROR HEADER INFO
 (1)  023310  012737  177777  002146      MOV     #-1,HEADER
3772
3773  023316  104503              9$:     CLR1CSR
3774  023320  062701  000002              ADD     #2,R1           ;POINT TO HIGH WORD
3775  023324  011100                      MOV     (R1),R0
3776  023326  020003                      CMP     R0,R3           ;SEE IF ITS BEEN CORRECTED
3777  023330  001401                      BEQ     10$             ;BR IF OK
3778  023332  104456                      PERR32
3779
3780  023334  104510             10$:     TSTREAD
3781  023336  103411                      BCS     11$             ;BR IF YES
3782  023340                              SET     HEADER          ;ENABLE PRINTING OF ERROR HEADER INFO
 (1)  023340  012737  177777  002146      MOV     #-1,HEADER
3783  023346  010137  001422              MOV     R1,ADDRESS
3784  023352  104016                      ERROR   +16
3785  023354                              SET     HEADER          ;ENABLE PRINTING OF ERROR HEADER INFO
 (1)  023354  012737  177777  002146      MOV     #-1,HEADER
3786  023362  005737  001724     11$:     TST     SBEMSK+2        ;TEST FOR LAST MASK BIT
3787  023366  100405                      BMI     12$             ;MINUS MEANS BIT 31
3788  023370                              DLEFT   SBEMSK
 (1)  023370  006137  001722              ROL     SBEMSK
 (1)  023374  006137  001724              ROL     SBEMSK+2
3789  023400  000643                      BR      3$
3790  023402                     12$:     IF #SW11 SET.IN @SWR THEN GOTO 13$                      BIT #SW11,@SWR
 (5)  023402  032777  004000  156570                                                              BNE 13$
 (5)  023410  001013
3791  023412                              IF QVFLAG NE #0 THEN GOTO 13$                           TST QVFLAG
 (5)  023412  005737  002016                                                                      BNE 13$
 (5)  023416  001010
3792  023420  005737  001714              TST     DATBUF+2        ;LAST DATA BIT ?
3793  023424  100405                      BMI     13$             ;WHICH IS BIT 31
3794  023426                              DLEFT   DATBUF
 (1)  023426  006137  001712              ROL     DATBUF
 (1)  023432  006137  001714              ROL     DATBUF+2
```

```
 3795  023436  000617                    BR      2$
 3796  023440  105737  001734    13$:    TSTB    PASFLG  ;FIRST OR SECOND PASS ?
 3797  023444  001003                    BNE     14$             ;NON ZERO MEANS WE'RE DONE
 3798  023446  105237  001734            INCB    PASFLG  ;NOT DONE, GO DO SECOND PASS
 3799  023452  000604                    BR      1$
 3800  023454  104471            14$:    ECC1DIS                 ;CLEAR OUT ANY DBE'S OR SBE'S
 3801  023456  013701  002070            MOV     TESTADD,R1
 3802  023462  005021                    CLR     (R1)+
 3803  023464  005011                    CLR     (R1)
 3804  023466  104503                    CLR1CSR
 3805  023470  000207                    RETURN
```

```
 3808  023472                          MTP012: SUBTST  <<MTP012        WRITE BYTE CLEARS SBE TEST>>
  (3)                                  ;*****************************************************************************
  (3)                                  ;*SUBTEST        MTP012  WRITE BYTE CLEARS SBE TEST
  (2)                                  ;*****************************************************************************
 3809                                                  ;SINGLE BIT ERROR TEST TO INSURE THAT A WRITE
 3810                                                  ;BYTE CLEARS SINGLE BIT ERRORS.
 3811  023472  104503                          CLR1CSR
 3812  023474  012737  000001  001712          MOV     #1,DATBUF       ;INITIAL DATA
 3813  023502  005037  001714                  CLR     DATBUF+2        ;32 BITS WORTH
 3814  023506  012737  000001  001722  1$:     MOV     #1,SBEMSK       ;INITIAL ERROR MASK
 3815  023514  005037  001724                  CLR     SBEMSK+2        ;32 BITS WORTH
 3816  023520  013737  001712  001716  2$:     MOV     DATBUF,TSTDAT   ;SAVE ORIGINAL DATA
 3817  023526  013737  001714  001720          MOV     DATBUF+2,TSTDAT+2;BOTH WORDS
 3818  023534  012737  001716  001760          MOV     #TSTDAT,SOURCE  ;NEED ADDRESS FOR CHKGEN
 3819  023542  004737  034350                  CALL    CHKGEN          ;GENERATE CHECK BITS
 3820  023546  013701  001722                  MOV     SBEMSK,R1
 3821  023552  074137  001716                  XOR     R1,TSTDAT
 3822  023556  013701  001724                  MOV     SBEMSK+2,R1
 3823  023562  074137  001720                  XOR     R1,TSTDAT+2
 3824  023566  013704  002070          3$:     MOV     TESTADD,R4      ;FIRST TEST ADDRESS
 3825  023572  010401                          MOV     R4,R1           ;PUT IT IN R1 ALSO
 3826  023574  104471                          ECC1DIS
 3827  023576  013721  001716                  MOV     TSTDAT,(R1)+    ;WRITE 16 BITS
 3828  023602  104475                          CB1CSR
 3829  023604  013711  001720                  MOV     TSTDAT+2,(R1)   ;WRITE HIGH WORD+CHECKBITS
 3830  023610  104503                          CLR1CSR                 ;IT'S DANGEROUS IF WE DON'T
 3831  023612  012702  001722                  MOV     #SBEMSK,R2      ;ADDRESS OF ERROR MASK
 3832  023616  162701  000002                  SUB     #2,R1           ;ADJUST ADDRESS
 3833  023622  112711  177777          4$:     MOVB    #-1,(R1)        ;WRITE A BYTE OF 1'S
 3834  023626  132712  177777                  BITB    #-1,(R2)        ;DID THIS BYTE HAVE THE BAD BIT IN IT?
 3835  023632  001420                          BEQ     6$              ;NO - BRANCH
 3836  023634  104510                          TSTREAD                 ;CLEAR CSR
 3837                                                                  ;READ
 3838                                                                  ;SINGLE ERROR INDICATE SET ?
 3839  023636  103011                          BCC     5$              ;NO - SKIP
 3840  023640                                  SET     HEADER          ;ENABLE PRINTING OF ERROR HEADER INFO
  (1)  023640  012737  177777  002146          MOV     #-1,HEADER
 3841  023646  010137  001422                  MOV     R1,ADDRESS
 3842  023652  104017                          ERROR   +17
 3843  023654                                  SET     HEADER          ;ENABLE PRINTING OF ERROR HEADER INFO
  (1)  023654  012737  177777  002146          MOV     #-1,HEADER
 3844
 3845  023662  111100                  5$:     MOVB    (R1),R0
 3846  023664  122700  177777                  CMPB    #-1,R0          ;CHECK DATA
 3847  023670  001414                          BEQ     7$              ;BR IF OK
 3848  023672  104457                          PERR33
 3849
 3850  023674  104510                  6$:     TSTREAD                 ;CLEAR CSR TO ENABLE ECC
 3851                                                                  ;READ THE BYTE
 3852                                                                  ;SBE ERROR BIT SET ?
 3853  023676  103771                          BCS     5$              ;SHOULD BE SET, BR IF OK
 3854  023700                                  SET     HEADER          ;ENABLE PRINTING OF ERROR HEADER INFO
  (1)  023700  012737  177777  002146          MOV     #-1,HEADER
 3855  023706  010137  001422                  MOV     R1,ADDRESS
 3856  023712  104016                          ERROR   +16
 3857  023714                                  SET     HEADER          ;ENABLE PRINTING OF ERROR HEADER INFO
```

F02


```
  (1)  023714  012737  177777  002146          MOV     #-1,HEADER
3858
3859   023722  132712  177777          7$:     BITB    #-1,(R2)        ;CHECK FOR LAST BYTE
3860   023726  001012                          BNE     8$              ;
3861   023730  005202                          INC     R2
3862   023732  005201                          INC     R1              ;MOVE   TO      NEXT BYTE
3863   023734  013704  002070                  MOV     TESTADD,R4      ;FIRST TEST ADDRESS
3864   023740  032701  000002                  BIT     #2,R1           ;TEST FOR LOWER WORD
3865   023744  001726                          BEQ     4$              ;BR IF IT'S LOW 16 BITS
3866   023746  062704  000002                  ADD     #2,R4           ;ADJUST POINTER FOR ERROR REPT.
3867   023752  000723                          BR      4$
3868   023754  005737  001724          8$:     TST     SBEMSK+2        ;LAST ERROR BIT ?
3869   023760  100405                          BMI     9$              ;MINUS MEANS BIT 31
3870   023762                          DLEFT   SBEMSK
  (1)  023762  006137  001722                  ROL     SBEMSK
  (1)  023766  006137  001724                  ROL     SBEMSK+2
3871   023772  000652                          BR      2$
3872   023774                          9$:     IF #SW11 SET.IN @SWR THEN GOTO 10$
  (5)  023774  032777  004000  156176                                                  BIT #SW11,@SWR
  (5)  024002  001013                                                                   BNE 10$
3873   024004                                  IF QVFLAG NE #0 THEN GOTO 10$
  (5)  024004  005737  002016                                                           TST QVFLAG
  (5)  024010  001010                                                                   BNE 10$
3874   024012  005737  001714                  TST     DATBUF+2        ;LAST DATA BIT?
3875   024016  100405                          BMI     10$             ;MINUS = BIT 31
3876   024020                          DLEFT   DATBUF
  (1)  024020  006137  001712                  ROL     DATBUF
  (1)  024024  006137  001714                  ROL     DATBUF+2
3877   024030  000626                          BR      1$
3878   024032  104471          10$:    ECC1DIS                         ;CLEAR OUT ANY DBE'S OR SBE'S
3879   024034  013701  002070                  MOV     TESTADD,R1
3880   024040  005021                          CLR     (R1)+
3881   024042  005011                          CLR     (R1)
3882   024044  104503                          CLR1CSR
3883   024046  000207                          RETURN
```

```
                                MTP013: SUBTST  <<MTP013         CREATE DOUBLE BIT ERROR TEST>>
                                ;*************************************************************************
                                ;*SUBTEST        MTP013  CREATE DOUBLE BIT ERROR TEST
                                ;*************************************************************************
                                        ;DOUBLE BIT ERROR FORCE TO CHECK DOUBLE ERROR LOGIC
                                CLR1CSR                         ;CLEAR CSR
                                MOV     TESTADD,R1
                          1$:   CLR     DATBUF                  ;MAKE INITIAL DATA
                                CLR     DATBUF+2                ;ALL ZEROS
001722                    2$:   MOV     #1,SBEMSK               ;INITIAL SINGLE ERROR MASK
                                CLR     SBEMSK+2                ;SECOND WORD
001726                    3$:   MOV     #1,DBEMSK               ;INITIAL DOUBLE ERROR MASK
                                CLR     DBEMSK+2                ;32 BITS HERE ALSO
001716  001712            4$:   MOV     DATBUF,TSTDAT           ;
001720  001714                  MOV     DATBUF+2,TSTDAT+2
001734                          TSTB    PASFLG  ;NO COMPLEMENTING FIRST PASS
                                BEQ     5$
                                COM     TSTDAT                  ;COMP FIRST WORD
                                COM     TSTDAT+2                ;SECOND WORD
                          5$:   CLR1CSR
001726  001722                 CMP     SBEMSK,DBEMSK           ;CAN'T HAVE THE SAME ERROR BIT SET
                                BNE     6$                      ;IN BOTH MASKS
001730  001724                 CMP     SBEMSK+2,DBEMSK+2       ;COULD BE EQUAL IN SECOND WORD
                                BEQ     9$                      ;GO MAKE THEM NOT EQUAL
001760  001716            6$:   MOV     #TSTDAT,SOURCE          ;SOURCE ADDRESS FOR CHKGEN
                                CALL    CHKGEN                  ;GO GENERATE CHECK BITS
                                MOV     SBEMSK,R2
                                XOR     R2,TSTDAT
                                MOV     SBEMSK+2,R2
                                XOR     R2,TSTDAT+2
                                MOV     DBEMSK,R2
                                XOR     R2,TSTDAT
                                MOV     DBEMSK+2,R2
                                XOR     R2,TSTDAT+2
                                ECC1DIS
                                MOV     TSTDAT,(R1)+            ;WRITE 16 BITS
                                CB1CSR
                                MOV     TSTDAT+2,(R1)           ;WRITE HIGH WORD
                                CLR1CSR
                                SUB     #2,R1                   ;ADJUST TEST ADDRESS
                                TST     (R1)                    ;READ THE LOCATION
                                WAS1DBE                         ;LOOK FOR DBE STATUS BIT
                                BCS     9$                      ;IT SHOULD BE SET
                                SET     HEADER
002146                          MOV     #-1,HEADER
                                MOV     R1,ADDRESS
                                ERROR   +30
                                SET     HEADER
002146                          MOV     #-1,HEADER

                          9$:   TST     DBEMSK+2                ;CHECK MASK FOR LAST BIT
                                BMI     10$                     ;MINUS = BIT31
                                DLEFT   DBEMSK
                                ROL     DBEMSK
                                ROL     DBEMSK+2
                                BR      4$
```

```
3935  024332                          10$:    IF #SW11 SET.IN @SWR THEN GOTO 11$
 (5)  024332  032777  004000  155640                                                  BIT #SW11,@SWR
 (5)  024340  001013                                                                  BNE 11$
3936  024342                                  IF QVFLAG NE #0 THEN GOTO 11$
 (5)  024342  005737  002016                                                          TST QVFLAG
 (5)  024346  001010                                                                  BNE 11$
3937  024350  005737  001724                  TST     SBEMSK+2          ;CHECK SINGLE ERROR MASK TOO
3938  024354  100405                          BMI     11$               ;BR IF DONE
3939  024356                                  DLEFT   SBEMSK
 (1)  024356  006137  001722                  ROL     SBEMSK
 (1)  024362  006137  001724                  ROL     SBEMSK+2
3940  024366  000644                          BR      3$
3941  024370  105737  001734          11$:    TSTB    PASFLG  ;FIRST PASS
3942  024374  001003                          BNE     12$               ;NON ZERO MEANS WE'RE DONE
3943  024376  105237  001734                  INCB    PASFLG  ;FIRST PASS, NOT DONE
3944  024402  000625                          BR      1$                ;KEEP GOING
3945  024404  104471                  12$:    ECC1DIS                   ;CLEAR OUT ANY DBE'S OR SBE'S
3946  024406  013701  002070                  MOV     TESTADD,R1
3947  024412  005021                          CLR     (R1)+
3948  024414  005011                          CLR     (R1)
3949  024416  104503                          CLR1CSR
3950  024420  000207                          RETURN
```

```
                                MTP014: SUBTST  <<MTP014        WRITE INHIBIT DURING DATIP WITH DBE TEST>>
                                ;*****************************************************************************
                                ;*SUBTEST       MTP014  WRITE INHIBIT DURING DATIP WITH DBE TEST
                                ;*****************************************************************************
                                ;THIS TEST CHECKS THE WRITE INHIBIT ON DOUBLE
                                ;BIT ERRORS DURING A DATIP OPERATION BY USE
                                ;OF AN 'ASRB' INSTRUCTION.
                                ;
024422 005037 001712        1$:     CLR     DATBUF              ;INITIAL DATA
024426 005037 001714                CLR     DATBUF+2            ;2 WORDS WORTH
024432 012737 000001 001722 2$:     MOV     #1,SBEMSK           ;INITIAL ERROR MASK
024440 005037 001724                CLR     SBEMSK+2            ;
024444 012737 000001 001726 3$:     MOV     #1,DBEMSK           ;DOUBLE ERROR MASK
024452 005037 001730                CLR     DBEMSK+2            ;2 WORDS
024456 013737 001712 001716 4$:     MOV     DATBUF,TSTDAT       ;PRESERVE ORIG DATA
024464 013737 001714 001720         MOV     DATBUF+2,TSTDAT+2
024472 105737 001734                TSTB    PASFLG  ;SECOND PASS YET ?
024476 001404                       BEQ     5$                  ;BR IF NO
024500 005137 001716                COM     TSTDAT              ;COMPL DATA  ON SECOND PASS
024504 005137 001720                COM     TSTDAT+2
024510 104503               5$:     CLR1CSR
024512 023737 001726 001722         CMP     DBEMSK,SBEMSK       ;CHECK FOR SAME MASKS
024520 001004                       BNE     6$                  ;BR IF OK
024522 023737 001730 001724         CMP     DBEMSK+2,SBEMSK+2
024530 001466                       BEQ     11$                 ;BR IF THEY'RE EQUAL
024532 012737 001716 001760 6$:     MOV     #TSTDAT,SOURCE      ;SET UP ADDRESS FOR CHKGEN
024540 004737 034350                CALL    CHKGEN              ;GENERATE CHECK BITS
024544 013701 001722                MOV     SBEMSK,R1
024550 074137 001716                XOR     R1,TSTDAT
024554 013701 001724                MOV     SBEMSK+2,R1
024560 074137 001720                XOR     R1,TSTDAT+2
024564 013701 001726                MOV     DBEMSK,R1
024570 074137 001716                XOR     R1,TSTDAT
024574 013701 001730                MOV     DBEMSK+2,R1
024600 074137 001720                XOR     R1,TSTDAT+2
024604 013701 002070        7$:     MOV     TESTADD,R1          ;TEST ADDRESS
024610 104471                       ECC1DIS
024612 013721 001716                MOV     TSTDAT,(R1)+        ;WRITE FIRST 16 BITS
024616 104475                       CB1CSR
024620 013711 001720                MOV     TSTDAT+2,(R1)       ;SECOND 16 BITS+CHECKBITS
024624 105037 001735                CLRB    UPPFLG              ;INDICATE LOWER WORD
024630 013703 002070                MOV     TESTADD,R3          ;TEST ADDRESS
024634 104503               8$:     CLR1CSR                     ;CLEAR IT
024636 106223                       ASRB    (R3)+               ;SPECIAL DATIP INSTRUCTION
024640 014100                       MOV     -(R1),R0
024642 023700 001716                CMP     TSTDAT,R0           ;CHECK FOR UNCHANGED DATA
024646 001401                       BEQ     9$                  ;SHOULD BE UNCHANGED
024650 104455                       PERR31
                                ;
024652 062701 000002        9$:     ADD     #2,R1               ;POINT TO UPPER WORD
024656 011100                       MOV     (R1),R0
024660 023700 001720                CMP     TSTDAT+2,R0         ;READ IT
024664 001401                       BEQ     10$                 ;BR IF UNCHANGED
024666 104455                       PERR31
                                ;
024670 122737 000003 001735 10$:    CMPB    #3,UPPFLG                   ;LOWER WORD
```

```
 4010  024676  001403                        BEQ     11$                 ;BR IF NO
 4011  024700  105237  001735                INCB    UPPFLG
 4012  024704  000753                        BR      8$
 4013  024706  005737  001730        11$:    TST     DBEMSK+2            ;LAST BIT IN MASK ?
 4014  024712  100405                        BMI     12$                 ;BR IF BIT 31
 4015  024714                                DLEFT   DBEMSK
   (1) 024714  006137  001726                ROL     DBEMSK
   (1) 024720  006137  001730                ROL     DBEMSK+2
 4016  024724  000654                        BR      4$
 4017  024726                        12$:    IF #SW11 SET.IN @SWR THEN GOTO 13$                      BIT #SW11,@SWR
   (5) 024726  032777  004000  155244                                                                BNE 13$
   (5) 024734  001013
 4018  024736                                IF QVFLAG NE #0 THEN GOTO 13$                           TST QVFLAG
   (5) 024736  005737  002016                                                                        BNE 13$
   (5) 024742  001010
 4019  024744  005737  001724                TST     SBEMSK+2            ;LAST BIT IN SINGLE ERROR MASK ?
 4020  024750  100405                        BMI     13$                 ;BR IF YES
 4021  024752                                DLEFT   SBEMSK
   (1) 024752  006137  001722                ROL     SBEMSK
   (1) 024756  006137  001724                ROL     SBEMSK+2
 4022  024762  000630                        BR      3$
 4023  024764  105737  001734        13$:    TSTB    PASFLG  ;WHICH PASS
 4024  024770  001003                        BNE     14$                 ;BR IF WE'RE DONE
 4025  024772  105237  001734                INCB    PASFLG  ;INDICATE SECOND PASS COMING
 4026  024776  000611                        BR      1$                  ;GO DO IT!
 4027  025000  104471        14$:    ECC1DIS                             ;CLEAR OUT ANY DBE'S OR SBE'S
 4028  025002  013701  002070                MOV     TESTADD,R1
 4029  025006  005021                        CLR     (R1)+
 4030  025010  005011                        CLR     (R1)
 4031  025012  104503                        CLR1CSR
 4032  025014  000207                        RETURN
```

```
 4035  025016                          MTP015: SUBTST  <<MTP015        WRITE INHIBIT OF BYTE WITH DBE>>
  (3)                                  ;*****************************************************************************
  (3)                                  ;*SUBTEST       MTP015  WRITE INHIBIT OF BYTE WITH DBE
  (2)                                  ;*****************************************************************************
 4036                                                ;CHECK FOR WRITE INHIBIT DURING A WRITE BYTE.
 4037                                                ;CHECKS FOR UNCORRECTED DATA.
 4038  025016  005037  001712          1$:     CLR     DATBUF                  ;INITIAL DATA
 4039  025022  005037  001714                  CLR     DATBUF+2                ;32 BITS WORTH
 4040  025026  012737  000001  001722  2$:     MOV     #1,SBEMSK               ;SINGLE ERROR MASK
 4041  025034  005037  001724                  CLR     SBEMSK+2                ;
 4042  025040  012737  000001  001726  3$:     MOV     #1,DBEMSK               ;DOUBLE ERROR MASK
 4043  025046  005037  001730                  CLR     DBEMSK+2                ;
 4044  025052  013737  001712  001716  4$:     MOV     DATBUF,TSTDAT           ;PRESERVE ORIG DATA
 4045  025060  013737  001714  001720          MOV     DATBUF+2,TSTDAT+2
 4046  025066  105737  001734                  TSTB    PASFLG  ;WHICH PASS ?
 4047  025072  001404                          BEQ     5$                      ;FIRST PASS, NO COMPLEMENTING
 4048  025074  005137  001716                  COM     TSTDAT
 4049  025100  005137  001720                  COM     TSTDAT+2                ;SECOND PASS, COMPLEMENT TSTDAT
 4050  025104  104503                  5$:     CLR1CSR
 4051  025106  023737  001722  001726          CMP     SBEMSK,DBEMSK           ;CHECK FOR SAME MASKS
 4052  025114  001004                          BNE     6$                      ;BR IF NOT EQUAL
 4053  025116  023737  001724  001730          CMP     SBEMSK+2,DBEMSK+2               ;SECOND WORD ALSO
 4054  025124  001463                          BEQ     11$                     ;BR TO MAKE THEM NOT EQUAL
 4055  025126  012737  001716  001760  6$:     MOV     #TSTDAT,SOURCE          ;ADDRESS FOR CHKGEN
 4056  025134  004737  034350                  CALL    CHKGEN                  ;GO GENERATE CHECK BITS
 4057  025140  013701  001722                  MOV     SBEMSK,R1
 4058  025144  074137  001716                  XOR     R1,TSTDAT
 4059  025150  013701  001724                  MOV     SBEMSK+2,R1
 4060  025154  074137  001720                  XOR     R1,TSTDAT+2
 4061  025160  013701  001726                  MOV     DBEMSK,R1
 4062  025164  074137  001716                  XOR     R1,TSTDAT
 4063  025170  013701  001730                  MOV     DBEMSK+2,R1
 4064  025174  074137  001720                  XOR     R1,TSTDAT+2
 4065  025200  013701  002070          7$:     MOV     TESTADD,R1              ;TEST LOCATION
 4066  025204  104471                          ECC1DIS                         ;DISABLE ECC
 4067  025206  013721  001716                  MOV     TSTDAT,(R1)+            ;WRITE FIRST 16 BITS
 4068  025212  104475                          CB1CSR                          ;LOAD CSR WITH IMAGE FROM R2
 4069  025214  013711  001720                  MOV     TSTDAT+2,(R1)           ;WRITE SECOND 16 BITS + CHECKBITS
 4070  025220  104503                          CLR1CSR
 4071  025222  013702  002070                  MOV     TESTADD,R2              ;GET ADDRESS OF TEST LOC
 4072  025226  010203                          MOV     R2,R3                   ;R2 DESIGNATES FIRST BYTE
 4073  025230  062703  000003                  ADD     #3,R3                   ;R3 DESIGNATES LAST BYTE
 4074  025234  112722  000360          8$:     MOVB    #360,(R2)+              ;TRY WRITING A BYTE
 4075  025240  013701  002070                  MOV     TESTADD,R1
 4076  025244  012100                          MOV     (R1)+,R0
 4077  025246  023700  001716                  CMP     TSTDAT,R0               ;CHECK FOR UNCHANGED DATA
 4078  025252  001401                          BEQ     9$                      ;BR IF OK
 4079  025254  104455                          PERR31
 4080
 4081  025256  011100                  9$:     MOV     (R1),R0
 4082  025260  023700  001720                  CMP     TSTDAT+2,R0             ;READ SECOND WORD
 4083  025264  001401                          BEQ     10$                     ;BR IF UNCHANGED
 4084  025266  104455                          PERR31
 4085
 4086  025270  020203                  10$:    CMP     R2,R3                   ;TESTED LAST BYTE ?
 4087  025272  001360                          BNE     8$                      ;BR IF NO
```

```
 4088  025274  005737  001730          11$:    TST     DBEMSK+2        ;CHECKING FOR LAST ERROR BIT
 4089  025300  100405                          BMI     12$             ;BR IF DONE HERE
 4090  025302                                  DLEFT   DBEMSK
   (1) 025302  006137  001726                  ROL     DBEMSK
   (1) 025306  006137  001730                  ROL     DBEMSK+2
 4091  025312  000657                          BR      4$
 4092  025314                          12$:    IF #SW11 SET.IN @SWR THEN GOTO 13$                      BIT #SW11,@SWR
   (5) 025314  032777  004000  154656                                                                  BNE 13$
   (5) 025322  001013
 4093  025324                                  IF QVFLAG NE #0 THEN GOTO 13$                           TST QVFLAG
   (5) 025324  005737  002016                                                                          BNE 13$
   (5) 025330  001010
 4094  025332  005737  001724                  TST     SBEMSK+2        ;LAST SBE MASK
 4095  025336  100405                          BMI     13$             ;BR IF DONE WITH THIS PASS
 4096  025340                                  DLEFT   SBEMSK
   (1) 025340  006137  001722                  ROL     SBEMSK
   (1) 025344  006137  001724                  ROL     SBEMSK+2
 4097  025350  000633                          BR      3$
 4098  025352  105737  001734          13$:    TSTB    PASFLG  ;TEST PASS FLAG
 4099  025356  001003                          BNE     14$             ;NON ZERO MEANS WE'RE DONE
 4100  025360  105237  001734                  INCB    PASFLG  ;NOT DONE
 4101  025364  000614                          BR      1$
 4102  025366  104471                  14$:    ECC1DIS                 ;DISABLE ECC
 4103  025370  013701  002070                  MOV     TESTADD,R1      ;TEST LOCATION
 4104  025374  005021                          CLR     (R1)+
 4105  025376  005011                          CLR     (R1)            ;TO ERASE ANY DBE'S FROM TESTING
 4106  025400  104503                          CLR1CSR                 ;RESTORE CSR
 4107  025402  000207                          RETURN
```

```
                        MTP016: SUBTST  <<MTP016         WRITE INHIBIT OF WORD WITH DBE>>
                        ;*****************************************************************************
                        ;*SUBTEST        MTP016  WRITE INHIBIT OF WORD WITH DBE
                        ;*****************************************************************************
                                ;DOUBLE BIT ERROR WRITE CANCEL WITH
                                ;WORD WRITE.
                                ;CHECKS WRITE INHIBIT WITH WORD WRITES TO
                                ;WORD WITH DOUBLE ERROR.
025404  005037  001712          T12A:   CLR     DATBUF                  ;BACKGROUND FOR DOUBLE ERRORS
025410  005037  001714                  CLR     DATBUF+2                ;2 WORDS WORTH
025414  012737  000001  001722          MOV     #1,SBEMSK               ;SINGLE ERROR MASK
025422  005037  001724                  CLR     SBEMSK+2                ;
025426  012737  000001  001726  T12B:   MOV     #1,DBEMSK               ;DOUBLE ERROR MASK
025434  005037  001730                  CLR     DBEMSK+2
025440  013737  001712  001716  1$:     MOV     DATBUF,TSTDAT           ;DATA FOR TEST
025446  013737  001714  001720          MOV     DATBUF+2,TSTDAT+2       ;BOTH WORDS
025454  105737  001734                  TSTB    PASFLG  ;COMP DATA ON SECOND PASS ONLY
025460  001404                          BEQ     2$                      ;BR IF FIRST PASS
025462  005137  001716                  COM     TSTDAT                  ;COMP FIRST WORD
025466  005137  001720                  COM     TSTDAT+2                ;NOW SECOND WORD
025472  023737  001722  001726  2$:     CMP     SBEMSK,DBEMSK           ;CHECK FOR IDENTICAL MASKS
025500  001004                          BNE     3$                      ;BR IF DIFFERENT
025502  023737  001724  001730          CMP     SBEMSK+2,DBEMSK+2       ;UPPER WORD TOO
025510  001470                          BEQ     8$                      ;BR TO MAKE THEM NOT EQUAL
025512  012737  001716  001760  3$:     MOV     #TSTDAT,SOURCE          ;NEED ADDR OF DATA FOR CHKGEN
025520  004737  034350                  CALL    CHKGEN                  ;GO GENERATE CHECK BITS
025524  013701  001722                  MOV     SBEMSK,R1
025530  074137  001716                  XOR     R1,TSTDAT
025534  013701  001724                  MOV     SBEMSK+2,R1
025540  074137  001720                  XOR     R1,TSTDAT+2
025544  013701  001726                  MOV     DBEMSK,R1
025550  074137  001716                  XOR     R1,TSTDAT
025554  013701  001730                  MOV     DBEMSK+2,R1
025560  074137  001720                  XOR     R1,TSTDAT+2
025564  013701  002070          4$:     MOV     TESTADD,R1              ;FIRST TEST ADRRESS
025570  104471                          ECC1DIS                         ;INHIBIT ECC
025572  013721  001716                  MOV     TSTDAT,(R1)+            ;WRITE FIRST 16 BITS
025576  104475                          CB1CSR
025600  013711  001720                  MOV     TSTDAT+2,(R1)           ;WRITE SECOND 16 BITS + CHECKBITS
025604  105037  001735                  CLRB    UPPFLG                  ;SET FOR 2 LOOPS
025610  162701  000002                  SUB     #2,R1                   ;POINT TO LOW WORD
025614  104503          5$:             CLR1CSR                         ;CLEAR CSR
025616  012711  177400                  MOV     #177400,(R1)            ;TRY WRITING LOCATION
025622  013701  002070                  MOV     TESTADD,R1
025626  011100                          MOV     (R1),R0
025630  023700  001716                  CMP     TSTDAT,R0               ;CHECK FOR ORIGINAL DATA
025634  001401                          BEQ     6$                      ;SHOULD BE UNCHANGED
025636  104455                          PERR31

025640  062701  000002          6$:     ADD     #2,R1
025644  011100                          MOV     (R1),R0
025646  023700  001720                  CMP     TSTDAT+2,R0             ;THIS SHOULD BE UNCHANGED ALSO
025652  001401                          BEQ     7$
025654  104455                          PERR31

025656  105737  001735          7$:     TSTB    UPPFLG                  ;WHICH LOOP ?
```

```
4163  025662  001003                    BNE     8$                ;SECOND, BR OUT
4164  025664  105237  001735            INCB    UPPFLG            ;FIRST, KEEP GOING
4165  025670  000751                    BR      5$
4166  025672  005737  001730    8$:     TST     DBEMSK+2          ;LAST BIT ?
4167  025676  100405                    BMI     9$                ;MINUS = BIT 31
4168  025700                            DLEFT   DBEMSK
  (1) 025700  006137  001726            ROL     DBEMSK
  (1) 025704  006137  001730            ROL     DBEMSK+2
4169  025710  000653                    BR      1$
4170  025712                    9$:     IF #SW11 SET.IN @SWR THEN GOTO 10$
  (5) 025712  032777  004000  154260                                                BIT #SW11,@SWR
  (5) 025720  001014                                                                BNE 10$
4171  025722                            IF QVFLAG NE #0 THEN GOTO 10$
  (5) 025722  005737  002016                                                        TST QVFLAG
  (5) 025726  001011                                                                BNE 10$
4172  025730  005737  001724            TST     SBEMSK+2          ;LAST BIT IN THIS MASK ?
4173  025734  100406                    BMI     10$               ;BR IF LAST BIT
4174  025736                            DLEFT   SBEMSK
  (1) 025736  006137  001722            ROL     SBEMSK
  (1) 025742  006137  001724            ROL     SBEMSK+2
4175  025746  000137  025426            JMP     T12B
4176  025752  105737  001734    10$:    TSTB    PASFLG  ;FIRST PASS ?
4177  025756  001004                    BNE     11$               ;BR IF SECOND
4178  025760  105237  001734            INCB    PASFLG  ;INDICATE SECOND PASS COMING
4179  025764  000137  025404            JMP     T12A
4180  025770  104471            11$:    ECC1DIS
4181  025772  013701  002070            MOV     TESTADD,R1        ;RESTORE TEST ADDRESS
4182  025776  005021                    CLR     (R1)+             ;CLEAR ANY DBE'S FROM TEST
4183  026000  005011                    CLR     (R1)
4184  026002  104503                    CLR1CSR                   ;CLEAR CSR
4185  026004  000207                    RETURN
```

```
4188  026006                        MTP017: SUBTST  <<MTP017        HOLDING 1'S & 0'S TEST>>
 (3)                                ;*********************************************************************
 (3)                                ;*SUBTEST        MTP017  HOLDING 1'S & 0'S TEST
 (2)                                ;*********************************************************************
4189                                        ;*(1)    THIS TEST CHECKS THE MEMORY FOR THE CAPABILITY
4190                                        ;*       OF HOLDING 1'S AND 0'S BY WRITING A BACKGROUND
4191                                        ;*       OF 000377 AND READING IT
4192                                        ;*(2)    MEMORY IS WRITTEN USING A BYTE AT A TIME
4193                                        ;*(3)    STEPS 1 & 2 ARE REPEATED WITH A SWAPPED BACKGROUND PATTERN
4194                                        ;NOTE:   THIS TEST WRITES BYTES & READS WORDS
4195  026006  012701  060000               MOV     #FIRST,R1
4196  026012  010104                       MOV     R1,R4
4197  026014  012705  160000               MOV     #LAST+2,R5
4198  026020  012700  000377               MOV     #377,R0         ;GET THE PATTERN INTO R0
4199  026024  010003                       MOV     R0,R3
4200  026026  000303                       SWAB    R3
4201  026030  110021                 1$:   MOVB    R0,(R1)+        ;WRITE A BYTE
4202  026032  110321                       MOVB    R3,(R1)+        ;WRITE THE MEMORY WITH THE BYTE STORED IN BAKPAT+1
4203  026034  020105                       CMP     R1,R5           ;COMPARE TEST LOC TO TOP + 2
4204  026036  103774                       BLO     1$              ;BRANCH IF LOWER
4205
4206  026040  014102                 2$:   MOV     -(R1),R2
4207  026042  020002                       CMP     R0,R2           ;TEST THE MEMORY TO SEE IF IT CONTAINS
4208                                                                ;THE WORD STORED IN BAKPAT
4209  026044  001401                       BEQ     3$
4210  026046  104446                       PERR22
4211
4212  026050  020104                 3$:   CMP     R1,R4           ;KEEP ON TESTING THE MEMORY UNTIL
4213  026052  101372                       BHI     2$              ;R1 EQUALS THE LOWEST ADDRESS
4214  026054  000303                       SWAB    R3              ;CHANGE THE DATA PATTERN
4215  026056  000300                       SWAB    R0
4216  026060  001763                       BEQ     1$              ;IF THE DATA PATTERN DOES NOT HAVE LOW
4217                                                                ; BYTE =0 THEN FALL THRU
4218  026062  000207                       RETURN
```

```
                        MTP020: SUBTST  <<MTP020        MARCHING 1'S & 0'S IN CHECK BITS TEST>>
                        ;*******************************************************************************
                        ;*SUBTEST       MTP020  MARCHING 1'S & 0'S IN CHECK BITS TEST
                        ;*******************************************************************************
                                ;*THIS TEST IS CONCERNED ONLY WITH THE INTEGRITY
                                ;*OF THE MOS RAMS THAT STORE THE CHECKBITS.

                                ;WRITE UP       ;100 --> CHECKBITS
                        MTP20A: MOV     R3,(R1)         ;V177640
                                MOV     R4,2(R1)        ;V177642
                                ADD     R2,R1           ;V177646
                                CMP     R1,R5           ;V177650        ;TOP + 2 ?
                                BLO     MTP20A          ;V177652        ;BR IF NOT
                                RETURN                  ;V177654

                                ;100 --> 077 DOWN
                        MTP20B: SUB     R2,R1           ;V177640
                                CMP     R4,2(R1)        ;V177642        ;2ND WORD OK?
                                BNE     21$             ;V177646        ;NO - SKIP
                                TST     (R1)            ;V177650        ;1ST WORD OK?
                                BEQ     3$              ;V177652        ;YES - SKIP
                        21$:    PERR26                  ;V177654        ;GOOD=000000,,100000,,100
                        3$:     MOV     R3,2(R1)        ;V177656
                                MOV     R3,(R1)         ;V177662
                                CMP     R1,R0           ;V177664
                                BNE     MTP20B          ;V177666
                                RETURN                  ;V177670
```

4221	026064			
(3)				
(3)				
(2)				
4222				
4223				
4224				
4225				
4226	026064	010311		
4227	026066	010461	000002	
4228	026072	060201		
4229	026074	020105		
4230	026076	103772		
4231	026100	000207		
4232				
4233				
4234	026102	160201		
4235	026104	020461	000002	
4236	026110	001002		
4237	026112	005711		
4238	026114	001401		
4239	026116	104452		
4240	026120	010361	000002	
4241	026124	010311		
4242	026126	020100		
4243	026130	001364		
4244	026132	000207		

```
                                        ;077 --> 100 UP
026134  005711                  MTP20C: TST     (R1)            ;V177640        ;1ST WORD OK?
026136  001003                          BNE     41$             ;V177642        ;NO - SKIP
026140  005761  000002                  TST     2(R1)           ;V177644        ;2ND WORD OK?
026144  001401                          BEQ     5$              ;V177650        ;YES - SKIP
026146  104453                  41$:    PERR27                  ;V177652        ;GOOD=000000,,000000,,077
026150  010311                  5$:     MOV     R3,(R1)         ;V177654        ;WRITE 1ST WORD
026152  010461  000002                  MOV     R4,2(R1)        ;V177656        ;WRITE 2ND WORD
026156  060201                          ADD     R2,R1           ;V177662
026160  020105                          CMP     R1,R5           ;V177664        ;TOP + 2 YET?
026162  103764                          BLO     MTP20C          ;V177666        ;NO - LOOP
026164  000207                          RETURN                  ;V177670

                                        ;100 --> 077 UP
026166  005711                  MTP20D: TST     (R1)            ;V177640        ;1ST WORD OK?
026170  001003                          BNE     61$             ;V177642        ;NO - SKIP
026172  020461  000002                  CMP     R4,2(R1)        ;V177644        ;2ND WORD OK?
026176  001401                          BEQ     7$              ;V177650        ;YES - SKIP
026200  104452                  61$:    PERR26                  ;V177652
026202  010311                  7$:     MOV     R3,(R1)         ;V177654
026204  010361  000002                  MOV     R3,2(R1)        ;V177656
026210  060201                          ADD     R2,R1           ;V177662
026212  020105                          CMP     R1,R5           ;V177664        ;TOP + 2 YET?
026214  103764                          BLO     MTP20D          ;V177666        ;NO - LOOP
026216  000207                          RETURN                  ;V177670
```

E03


```
4272                                            ;077 --> 100 DOWN
4273  026220  160201            MTP20E: SUB     R2,R1           ;V177640
4274  026222  005761  000002            TST     2(R1)           ;V177642        ;2ND WORD OK?
4275  026226  001002                     BNE     81$             ;V177646        ;NO - SKIP
4276  026230  005711                     TST     (R1)            ;V177650        ;1ST WORD OK?
4277  026232  001401                     BEQ     9$              ;V177652        ;YES - SKIP
4278  026234  104453            81$:    PERR27                  ;V177654
4279  026236  010461  000002    9$:     MOV     R4,2(R1)        ;V177656        ;WRITE 2ND WORD
4280  026242  010311                     MOV     R3,(R1)         ;V177662        ;WRITE 1ST WORD
4281  026244  020100                     CMP     R1,R0           ;V177664        ;BOTTOM YET?
4282  026246  001364                     BNE     MTP20E          ;V177666        ;NO - LOOP
4283  026250  000207                     RETURN                  ;V177670
4284
4285                                            ;100 UP
4286  026252  005711            MTP20F: TST     (R1)            ;V177640        ;1ST WORD OK?
4287  026254  001003                     BNE     101$            ;V177642        ;NO - SK..
4288  026256  020461  000002            CMP     R4,2(R1)        ;V177642        ;2ND WORD OK?
4289  026262  001401                     BEQ     11$             ;V177646        ;YES - SKIP
4290  026264  104452            101$:   PERR26                  ;V177650
4291  026266  060201            11$:    ADD     R2,R1           ;V177652
4292  026270  020105                     CMP     R1,R5           ;V177654        ;TOP + 2 YET?
4293  026272  103767                     BLO     MTP20F          ;V177656        ;NO - LOOP
4294  026274  000207                     RETURN                  ;V177660
```

F03


```
                                MTPA20: SUBTST  <<MTPA20         SLOW MARCHING 1'S & 0'S IN CHECK BITS TEST>>
                                ;*****************************************************************************
                                ;*SUBTEST        MTPA20  SLOW MARCHING 1'S & 0'S IN CHECK BITS TEST
                                ;*****************************************************************************
                                        ;*THIS TEST IS CONCERNED ONLY WITH THE INTEGRITY
                                        ;*OF THE MOS RAMS THAT STORE THE CHECKBITS.
026276                                  SET     SBEFLAG
(1) 026276  012737  177777  001412      MOV     #-1,SBEFLAG
026304  013701  001710                  MOV     CSRFIRST,R1     ;FIRST TEST LOC
026310  013702  001772                  MOV     CSRINC,R2
026314  010103                          MOV     R1,R3
026316  005004                          CLR     R4
026320  012705  160000                  MOV     #LAST+2,R5
026324  005037  001526                  CLR     CSR
026330  104505                          CHK1DIS                 ;WRITE CHECKBITS IN CSR
                                        ;*************************************************************
026332  010411                  2$:     MOV     R4,(R1)         ;WRITE CHECKBITS BY WRITING LOC     * WRITE UP
026334  060201                          ADD     R2,R1           ;POINT TO NEXT WORD                 * 0'S --> CHK BI
026336  020105                          CMP     R1,R5           ;TOP + 2                            *
026340  103774                          BLO     2$              ;BR IF NOT                          *
                                        ;*************************************************************
                                        ;*************************************************************
026342  160201                  3$:     SUB     R2,R1           ;ADJUST R1                          *
026344  005711                          TST     (R1)            ;READ                               *
026346  104426                          READCSR                 ;                                   *
026350  032737  077400  001526          BIT     #77400,@#CSR    ;SHOULD BE ALL ZEROS                *
026356  001406                          BEQ     4$              ;BR IF OK                           *
026360  013700  001526                  MOV     CSR,R0          ;                                   *
026364  000300                          SWAB    R0              ;                                   *
026366  042700  000200                  BIC     #200,R0         ;                                   *
026372  104435                          PERR11                  ;                                   *
                                        ;                                                           *
026374  012737  077400  001526  4$:     MOV     #77400,@#CSR    ;MAKE CHECKBITS ALL 1'S             *0'S --> 1'S
026402  104505                          CHK1DIS                 ;                                   * DOWN
026404  005104                          COM     R4              ;COMP THE DATA                      *
026406  010411                          MOV     R4,(R1)         ;WRITE THE CHECKBITS                *
026410  020103                          CMP     R1,R3           ;BOTTOM YET?                        *
026412  001353                          BNE     3$              ;BR IF NO                           *
                                        ;*************************************************************

026414  005037  001526                  CLR     CSR
026420  104505                          CHK1DIS
```

```
                                                ;*****************************************************************
026422  005711                          5$:     TST     (R1)            ;READ THE LOC AGAIN                     *
026424  104426                                  READCSR                 ;                                       *
026426  013700  001526                          MOV     @#CSR,R0        ;GET CSR CONTENTS INTO R0               *
026432  042700  100377                          BIC     #100377,R0      ;ONLY WANT BITS 11-5                    *
026436  022700  077400                          CMP     #77400,R0       ;READ FOR ALL 1'S                       *
026442  001404                                  BEQ     6$              ;BR IF OK                               *
026444  000300                                  SWAB    R0              ;                                       *
026446  042700  000200                          BIC     #200,R0         ;                                       *
026452  104445                                  PERR21                  ;                                       *1'S --> 0'S
                                                                        ;                                       * UP
026454  005037  001526                  6$:     CLR     CSR             ;                                       *
026460  104505                                  CHK1DIS                 ;                                       *
026462  005104                                  COM     R4              ;                                       *
026464  010411                                  MOV     R4,(R1)         ;ACCESS THE LOC                         *
026466  060201                                  ADD     R2,R1           ;POINT TO NEXT ADDRESS                  *
026470  020105                                  CMP     R1,R5           ;TOP+2 YET?                             *
026472  103753                                  BLO     5$              ;BR IF NOT                              *
                                                ;*****************************************************************

026474  010301                                  MOV     R3,R1

                                                ;*****************************************************************
026476  005711                          7$:     TST     (R1)            ;READ                                   *
026500  104426                                  READCSR                 ;                                       *
026502  032737  077400  001526                  BIT     #77400,@#CSR    ;CHECKBITS SHOULD BE ZERO               *
026510  001406                                  BEQ     8$              ;BR IF OK                               *
026512  013700  001526                          MOV     CSR,R0          ;                                       *
026516  000300                                  SWAB    R0              ;                                       *
026520  042700  000200                          BIC     #200,R0         ;                                       *
026524  104435                                  PERR11                  ;                                       *0'S --> 1'S
                                                                        ;                                       * UP
026526  012737  077400  001526          8$:     MOV     #77400,@#CSR    ;COMP CHECKBITS                         *
026534  104505                                  CHK1DIS                 ;                                       *
026536  005104                                  COM     R4              ;                                       *
026540  010411                                  MOV     R4,(R1)         ;WRITE THE LOC                          *
026542  060201                                  ADD     R2,R1           ;POINT TO NEXT ADDRESS                  *
026544  020105                                  CMP     R1,R5           ;TOP+2 YET?                             *
026546  103753                                  BLO     7$              ;BR IF NOT                              *
                                                ;*****************************************************************

026550  005037  001526                          CLR     CSR
026554  104505                                  CHK1DIS
```

```
                                                ;*****************************************************************
026556  160201                          9$:     SUB     R2,R1           ;ADJUST R1                          *
026560  005711                                  TST     (R1)            ;READ                               *
026562  104426                                  READCSR                 ;                                   *
026564  013700  001526                          MOV     @#CSR,R0        ;GET CSR CONTENTS                   *
026570  042700  100377                          BIC     #100377,R0      ;ONLY WANT CHECKBITS                *
026574  022700  077400                          CMP     #77400,R0       ;ALL 1'S?                           *
026600  001404                                  BEQ     10$             ;BR IF OK                           *
026602  000300                                  SWAB    R0              ;                                   *
026604  042700  000200                          BIC     #200,R0         ;                                   *
026610  104445                                  PERR21                  ;                                   *1'S --> 0'S
                                                                        ;                                   * DOWN
026612  005037  001526                  10$:    CLR     CSR             ;                                   *
026616  104505                                  CHK1DIS                 ;                                   *
026620  005104                                  COM     R4              ;COMP THE DATA                      *
026622  010411                                  MOV     R4,(R1)         ;WRITE THE CHECKBITS                *
026624  020103                                  CMP     R1,R3           ;BOTTOM YET?                        *
026626  001353                                  BNE     9$              ;BR IF NO                           *
                                                ;*****************************************************************

                                                ;*****************************************************************
026630  005711                          11$:    TST     (R1)            ;READ                               *
026632  104426                                  READCSR                 ;                                   *
026634  032737  077400  001526                  BIT     #77400,@#CSR    ;SHOULD BE ALL 0'S                  *
026642  001406                                  BEQ     12$             ;BR IF OK                           *
026644  013700  001526                          MOV     CSR,R0          ;                                   *
026650  000300                                  SWAB    R0              ;                                   *
026652  042700  000200                          BIC     #200,R0         ;                                   *
026656  104435                                  PERR11                  ;                                   *0'S
                                                                        ;                                   * UP
026660  060201                          12$:    ADD     R2,R1           ;POINT TO NEXT WORD                 *
026662  005037  001526                          CLR     CSR             ;                                   *
026666  104505                                  CHK1DIS                 ;                                   *
026670  020105                                  CMP     R1,R5           ;TOP + 2 YET?                       *
026672  103756                                  BLO     11$             ;BR IF NOT                          *
                                                ;*****************************************************************

                                                ;FLUSH OUT ALL THESE DBE'S
026674                                          BMOV    MTP030
(1)  026674  004537  037306                          JSR R5,BLOCK1
(1)  026700  030676                                  MTP030
026702  012700  060000                          MOV     #FIRST,R0
026706  012701  040000                          MOV     #SIZE,R1
026712  104471                                  ECC1DIS
026714  004737  177640                          CALL    FASTCITY
026720  104503                                  CLR1CSR
026722  000207                                  RETURN
```

```
 4429  026724                 MTPA21: SUBTST  <<MTPA21        MARCHING 1'S & 0'S PATTERN TEST>>
  (3)                         ;*****************************************************************************
  (3)                         ;*SUBTEST       MTPA21  MARCHING 1'S & 0'S PATTERN TEST
  (2)                         ;*****************************************************************************
 4430                                 ;READ,BYTESWAP-MODIFY,READ,DOWN
 4431  026724  014100         1$:     MOV     -(R1),R0;V177640
 4432  026726  020200                 CMP     R2,R0   ;V177642
 4433  026730  001401                 BEQ     2$      ;V177644
 4434  026732  104443                 PERR17          ;V177646
 4435
 4436  026734  000311         2$:     SWAB    (R1)    ;V177650
 4437  026736  011100                 MOV     (R1),R0 ;V177652
 4438  026740  020300                 CMP     R3,R0   ;V177654
 4439  026742  001401                 BEQ     3$      ;V177656
 4440  026744  104444                 PERR20          ;V177660
 4441
 4442  026746  020401         3$:     CMP     R4,R1   ;V177662        ;DONE?
 4443  026750  001365                 BNE     1$      ;V177664        ;NO - LOOP
 4444  026752  000207                 RETURN          ;V177666        ;YES - RETURN
 4445
 4446  026754                 MTPB21: ;READ,BYTESWAP-MODIFY,READ,UP
 4447  026754  011100         1$:     MOV     (R1),R0 ;V177640
 4448  026756  020300                 CMP     R3,R0   ;V177642
 4449  026760  001401                 BEQ     2$      ;V177644
 4450  026762  104444                 PERR20          ;V177646
 4451
 4452  026764  000311         2$:     SWAB    (R1)    ;V177650
 4453  026766  011100                 MOV     (R1),R0 ;V177652
 4454  026770  020200                 CMP     R2,R0   ;V177654
 4455  026772  001401                 BEQ     3$      ;V177656
 4456  026774  104443                 PERR17          ;V177660
 4457
 4458  026776  062701  000002 3$:     ADD     #2,R1   ;V177662
 4459  027002  020501                 CMP     R5,R1   ;V177666        ;DONE?
 4460  027004  001363                 BNE     1$      ;V177670        ;NO - LOOP
 4461  027006  000207                 RETURN          ;V177672        ;YES - RETURN
 4462
 4463  027010                 MTPC21: ;READ,BYTESWAP-MODIFY,READ,UP
 4464  027010  011100         1$:     MOV     (R1),R0 ;V177640
 4465  027012  020200                 CMP     R2,R0   ;V177642
 4466  027014  001401                 BEQ     2$      ;V177644
 4467  027016  104443                 PERR17          ;V177646
 4468
 4469  027020  000311         2$:     SWAB    (R1)    ;V177650
 4470  027022  011100                 MOV     (R1),R0 ;V177652
 4471  027024  020300                 CMP     R3,R0   ;V177654
 4472  027026  001401                 BEQ     3$      ;V177656
 4473  027030  104444                 PERR20          ;V177660
 4474
 4475  027032  062701  000002 3$:     ADD     #2,R1   ;V177662
 4476  027036  020501                 CMP     R5,R1   ;V177666        ;DONE?
 4477  027040  001363                 BNE     1$      ;V177670        ;NO - LOOP
 4478  027042  000207                 RETURN          ;V177672        ;YES - RETURN
 4479
 4480  027044                 MTPD21: ;READ,BYTESWAP-MODIFY,READ,DOWN
 4481  027044  014100         1$:     MOV     -(R1),R0;V177640
```

```
4482  027046  020300                  CMP     R3,R0   ;V177642
4483  027050  001401                  BEQ     2$      ;V177644
4484  027052  104444                  PERR20          ;V177646
4485
4486  027054  000311          2$:     SWAB    (R1)    ;V177650
4487  027056  011100                  MOV     (R1),R0 ;V177652
4488  027060  020200                  CMP     R2,R0   ;V177654
4489  027062  001401                  BEQ     3$      ;V177656
4490  027064  104443                  PERR17          ;V177660
4491
4492  027066  020401          3$:     CMP     R4,R1   ;V177662        ;DONE?
4493  027070  001365                  BNE     1$      ;V177664        ;NO - LOOP
4494  027072  000207                  RETURN          ;V177666        ;YES - RETURN
4495
```

```
 4498  027074                          MTP022: SUBTST  <<MTP022        REFRESH TEST>>
  (3)                                  ;*****************************************************************************
  (3)                                  ;*SUBTEST        MTP022  REFRESH TEST
  (2)                                  ;*****************************************************************************
 4499          000400                  REFDEL=256.                ;TIME FOR > 2 MS LOOP
 4500          000010                  KDIAG=8.                   ;HOW OFTEN A DIAGNAL STRIPE OCCURS (MUST BE A POWER OF 2)
 4501  027074                                  FOR EVEN := #1 TO #2
  (4)  027074  012737  000001  002034                                                                  MOV #1,EVEN
  (4)  027102                                                                                    B34:;;;;;;
 4502  027102                                    IF EVEN EQ #1
  (4)  027102  023727  002034  000001                                                                  CMP EVEN,#1
  (6)  027110  001005                                                                                  BNE L250
 4503  027112                                      LET R2 := ZEROS
  (3)  027112  013702  002006                                                                          MOV ZEROS,R2
 4504  027116                                      LET R3 := ONES
  (3)  027116  013703  002154                                                                          MOV ONES,R3
 4505  027122                                    ELSE
  (3)  027122  000404                                                                                  BR L251
  (3)  027124                                                                                    L250:;;;;;;
 4506  027124                                      LET R2 := ONES
  (3)  027124  013702  002154                                                                          MOV ONES,R2
 4507  027130                                      LET R3 := ZEROS
  (3)  027130  013703  002006                                                                          MOV ZEROS,R3
 4508  027134                                    END ;OF IF EVEN
  (3)  027134                                                                                    L251:;;;;;;
 4509  027134                                    FOR STRIPES := #0 TO #KDIAG-1
  (4)  027134  005037  002036                                                                          CLR STRIPES
  (4)  027140                                                                                    B35:;;;;;;
 4510
 4511                                            ;WRITE LOOP
 4512  027140  104423                            CACHON
 4513  027142                                    LET COUNT := STRIPES
  (3)  027142  013737  002036  002040                                                                  MOV STRIPES,COUNT
 4514  027150                                    LET R1 := #FIRST
  (3)  027150  012701  060000                                                                          MOV #FIRST,R1
 4515  027154                                    WHILE R1 LOS #LAST
  (3)  027154                                                                                    B36:;;;;;;
  (5)  027154  020127  157776                                                                          CMP R1,#LAST
  (7)  027160  101032                                                                                  BHI L252
 4516  027162                                      IF COUNT LT #0 THEN LET COUNT := #KDIAG-1
  (4)  027162  005737  002040                                                                          TST COUNT
  (6)  027166  002003                                                                                  BGE L253
  (5)  027170  012737  000007  002040                                                                  MOV #KDIAG-1,COUNT
  (3)  027176                                                                                    L253:;;;;;;
 4517  027176                                      IF #374 OFF.IN R1 THEN LET COUNT := COUNT - #1
  (4)  027176  032701  000374                                                                          BIT #374,R1
  (6)  027202  001002                                                                                  BNE L254
  (7)  027204  005337  002040                                                                          DEC COUNT
  (3)  027210                                                                                    L254:;;;;;;
 4518  027210                                      IF COUNT NE #0
  (4)  027210  005737  002040                                                                          TST COUNT
  (6)  027214  001404                                                                                  BEQ L255
 4519  027216                                        LET (R1) := R2
  (3)  027216  010211                                                                                  MOV R2,(R1)
 4520  027220                                        LET 2(R1) := R2
  (3)  027220  010261  000002                                                                          MOV R2,2(R1)
```

```
4521  027224
 (3)  027224  000403
 (3)  027226
4522  027226
 (3)  027226  010311
4523  027230
 (3)  027230  010361  000002
4524  027234
 (3)  027234
4525  027234
 (5)  027234  005337  002040
4526  027240
 (5)  027240  062701  000004
4527  027244
 (3)  027244  000743
 (3)  027246
 (3)  027246
4528
4529
4530  027246  004737  027442
4531
4532
4533  027252
 (3)  027252  013737  002036  002040
4534  027260
 (3)  027260  012701  060000
4535  027264  104424
4536  027266
 (3)  027266
 (5)  027266  020127  157776
 (7)  027272  101046
4537  027274
 (4)  027274  005737  002040
 (6)  027300  002003
 (5)  027302  012737  000007  002040
 (3)  027310
4538  027310
 (4)  027310  032701  000374
 (6)  027314  001002
 (7)  027316  005337  002040
 (3)  027322
4539  027322
 (4)  027322  005737  002040
 (6)  027326  001412
4540  027330
 (3)  027330  011100
4541  027332
 (4)  027332  020200
 (6)  027334  001401
4542  027336  104443
4543  027340
 (3)  027340
4544  027340
 (3)  027340  016100  000002
4545  027344
 (4)  027344  020200
```

```
        ELSE

          LET (R1) := R3
          LET 2(R1) := R3
        END ;OF IF COUNT
        LET COUNT := COUNT - #1
        LET R1 := R1 + #4
      END ;OF WHILE


      ;END OF WRITE LOOP
      CALL        REFRESH
      ;READ LOOP
      LET COUNT := STRIPES
      LET R1 := #FIRST
      CACHOFF
      WHILE R1 LOS #LAST


        IF COUNT LT #0 THEN LET COUNT := #KDIAG-1

        IF #374 OFF.IN R1 THEN LET COUNT := COUNT - #1

        IF COUNT NE #0

          LET R0 := (R1)
          IF R2 NE R0

            PERR17
          END ;OF IF R2
          LET R0 := 2(R1)
          IF R2 NE R0
```

```
        BR L256
L255:::::::
        MOV R3,(R1)
        MOV R3,2(R1)
L256:::::::
        DEC COUNT
        ADD #4,R1
        BR B36
L_52:::::::
E36:::::::



        MOV STRIPES,COUNT
        MOV #FIRST,R1

B37:::::::
        CMP R1,#LAST
        BHI L257
        TST COUNT
        BGE L260
        MOV #KDIAG-1,COUNT
L260:::::::
        BIT #374,R1
        BNE L261
        DEC COUNT
L261:::::::
        TST COUNT
        BEQ L262
        MOV (R1),R0
        CMP R2,R0
        BEQ L263


L263:::::::
        MOV 2(R1),R0
        CMP R2,R0
```

```
  (6)  027346  001401                                                                  BEQ L264
 4546  027350  104443                                  PERR17
 4547  027352                                          END ;OF IF R2
  (3)  027352                                                                      L264:;;;;;;
 4548  027352                                        ELSE
  (3)  027352  000411                                                                  BR L265
  (3)  027354                                                                      L262:;;;;;;
 4549  027354                                          LET R0 := (R1)
  (3)  027354  011100                                                                  MOV (R1),R0
 4550  027356                                          IF R3 NE R0
  (4)  027356  020300                                                                  CMP R3,R0
  (6)  027360  001401                                                                  BEQ L266
 4551  027362  104444                                  PERR20
 4552  027364                                          END ;OF IF R3
  (3)  027364                                                                      L266:;;;;;;
 4553  027364                                          LET R0 := 2(R1)
  (3)  027364  016100  000002                                                          MOV 2(R1),R0
 4554  027370                                          IF R3 NE R0
  (4)  027370  020300                                                                  CMP R3,R0
  (6)  027372  001401                                                                  BEQ L267
 4555  027374  104444                                  PERR20
 4556  027376                                          END ;OF IF R3
  (3)  027376                                                                      L267:;;;;;;
 4557  027376                                        END ;OF IF COUNT
  (3)  027376                                                                      L265:;;;;;;
 4558  027376                                        LET COUNT := COUNT - #1
  (5)  027376  005337  002040                                                          DEC COUNT
 4559  027402                                        LET R1 := R1 + #4
  (5)  027402  062701  000004                                                          ADD #4,R1
 4560  027406                                      END ;OF WHILE
  (3)  027406  000727                                                                  BR B37
  (3)  027410                                                                      L257:;;;;;;
  (3)  027410                                                                      E37:;;;;;;
 4561                                              ;END OF READ LOOP
 4562
 4563  027410                                    END ;OF FOR STRIPES
  (6)  027410  005237  002036                                                          INC STRIPES
  (5)  027414  023727  002036  000007                                                  CMP STRIPES,#KDIAG-1
  (5)  027422  003646                                                                  BLE B35
  (5)  027424                                                                      E35:;;;;;;;
 4564  027424                                  END ;OF FOR EVEN
  (6)  027424  005237  002034                                                          INC EVEN
  (5)  027430  023727  002034  000002                                                  CMP EVEN,#2
  (5)  027436  003621                                                                  BLE B34
  (5)  027440                                                                      E34:;;;;;;;
 4565  027440  000207                          RETURN
 4566  027442                              REFRESH:SUBTST  <<SUBR  REFRESH DELAY>>
  (3)                                      ;*****************************************************************************
  (3)                                      ;*SUBTEST        SUBR    REFRESH DELAY
  (2)                                      ;*****************************************************************************
 4567                                              ;DISTURB EACH ROW FOR > 2 MS
 4568  027442                                      FOR R0 := #FIRST TO #FIRST+374 BY #4
  (3)  027442  012700  060000                                                          MOV #FIRST,R0
  (3)  027446                                                                      B40:;;;;;;
 4569  027446  004737  027512                        CALL REFSUB
 4570  027452                                      END ;OF FOR R0
```

```
   (6)  027452  062700  000004
   (5)  027456  020027  060374
   (5)  027462  003771
   (5)  027464
 4571   027464                          LET R0 := #FIRST+BIT14
   (3)  027464  012700  120000          WHILE R0 LOS #LAST+BIT14+374
 4572   027470
   (3)  027470
   (5)  027470  020027  020372
   (7)  027474  101005
 4573   027476  004737  027512            CALL REFSUB
 4574   027502                            LET R0 := R0 + #4
   (5)  027502  062700  000004
 4575   027506                          END ;OF WHILE
   (3)  027506  000770
   (3)  027510
   (3)  027510
 4576   027510  000207                  RETURN
 4577
 4578   027512  012704  000400  REFSUB: MOV     #REFDEL,R4
 4579   027516  062700  000002          ADD     #2,R0
 4580   027522  005140          1$:     COM     -(R0)
 4581   027524  005120                  COM     (R0)+
 4582   027526  005110                  COM     (R0)
 4583   027530  005110                  COM     (R0)
 4584   027532  077405                  SOB     R4,1$
 4585   027534  162700  000002          SUB     #2,R0
 4586   027540  000207                  RETURN
```

```
        ADD #4,R0
        CMP R0,#FIRST+374
        BLE B40
E40:;;;;;;;
        MOV #FIRST+BIT14,R0
B41:;;;;;;;
        CMP R0,#LAST+BIT14+374
        BHI L270

        ADD #4,R0

        BR B41
L_70:;;;;;;
E41:;;;;;;;
```

```
4589  027542                          MTP023: SUBTST  <<MTP023        SHIFTING DIAGONAL PATTERN TEST>>
 (3)                                  ;*********************************************************************************
 (3)                                  ;*SUBTEST        MTP023  SHIFTING DIAGONAL PATTERN TEST
 (2)                                  ;*********************************************************************************
4590                                          ;*(1)    THIS TEST WRITES THE MEMORY WITH A BACKGROUND OF BAKPAT
4591                                          ;*(2)    IT WRITES A DIAGONAL OF SWAPPED BAKPAT THROUGH EACH MEMORY SEMIBANK (4K)
4592                                          ;*(3)    READS THE MEMORY FOR CORRECT DATA
4593                                          ;*(4)    SHIFTS THE DIAGONAL AND REPEATS STEP 3 UNTIL THE
4594                                          ;*       DIAGONAL HAS BEEN SHIFTED 64 TIMES
4595                                          ;*(5)    WRITES A BACKGROUND OF SWAPPED BAKPAT, A DIAGONAL OF
4596                                          ;*       BAKPAT AND REPEATS FROM STEP 3
4597  027542  105037  001734          1$:     CLRB    PASFLG
4598  027546  010437  001736                  MOV     R4,LOWBNK       ;LOWBNK CONTAINS ADDRESS OF THE LOWEST LOCATION
4599                                                                  ;IN THE SEMIBANK (4K) THAT CAN BE TESTED
4600  027552  010402                          MOV     R4,R2
4601  027554  052702  017776                  BIS     #17776,R2       ;R2 CONTAINS THE ADDRESS OF THE TOP OF THE SEMIBANK (4K)
4602  027560  062702  000002                  ADD     #2,R2           ;ADD 2 TO R2
4603  027564  022702  160000                  CMP     #LAST+2,R2
4604  027570  103002                          BHIS    2$              ;IF R2 IS GREATER THAN LAST+2 THEN GO TO 2$
4605  027572  012702  160000                  MOV     #LAST+2,R2      ;NOW R2 CONTAINS THE ADDRESS OF THE HIGHEST LOCATION
4606                                                                  ;THAT CAN BE TESTED
4607  027576  010437  001740          2$:     MOV     R4,STRTDI       ;LOAD STRTDI WITH THE STARTING ADDRESS OF THE
4608                                                                  ;DIAGONAL
4609  027602  013701  001736                  MOV     LOWBNK,R1       ;R1 IS NOW POINTING TO THE LOWEST LOCATION IN THE 4K
4610                                                                  ;BANK
4611  027606  013703  002174          3$:     MOV     BAKPAT,R3       ;STORE THE CONTENTS OF BAKPAT IN R3
4612  027612  020104                          CMP     R1,R4           ;IS R1 POINTING TO A LOCATION IN THE DIAGONAL ?
4613  027614  001010                          BNE     5$              ;IF NOT THEN GO TO 5$
4614  027616  062704  000002                  ADD     #2,R4           ;THE FOLLOWING CODE IS USED TO PLACE THE
4615  027622  032704  000176                  BIT     #176,R4         ;ADDRESS OF THE NEXT LOCATION IN THE DIAGONAL
4616  027626  001402                          BEQ     4$              ;IN R4
4617  027630  062704  000200                  ADD     #200,R4
4618  027634  000303                  4$:     SWAB    R3              ;DIAGONAL WILL CONTAIN SWAPPED BACKGROUND PATTERN
4619  027636  132737  000001  001734  5$:     BITB    #1,PASFLG       ;CONTENTS OF LOCATION PASFLG WILL BE EVEN IF THE
4620                                                                  ;MEMORY IS BEING WRITTEN AND IT WILL BE ODD
4621                                                                  ;IF IT IS ONLY BEING READ
4622  027644  001001                          BNE     6$              ;IF IT IS BEING READ ONLY THEN GO TO 6$
4623  027646  010311                          MOV     R3,(R1)         ;OTHERWISE WRITE THE MEMORY WITH THE CONTENTS
4624                                                                  ;OF R3
4625  027650  011100                  6$:     MOV     (R1),R0
4626  027652  020003                          CMP     R0,R3           ;CHECK THE LOCATION POINTED BY R1 TO CONTAIN
4627                                                                  ;PROPER DATA
4628  027654  001401                          BEQ     7$              ;IF IT IS OK THEN GO TO 7$
4629  027656  104444                          PERR20
4630
4631  027660  062701  000002          7$:     ADD     #2,R1           ;CAUSE R1 TO POINT TO THE NEXT MEMORY LOCATION
4632  027664  020102                          CMP     R1,R2           ;IS IT THE END OF THE SEMIBANK (4K) ?
4633  027666  103747                          BLO     3$              ;IF NOT THEN GO TO 3$
4634  027670  105237  001734          8$:     INCB    PASFLG
4635  027674  013704  001740                  MOV     STRTDI,R4       ;LOAD R4 WITH THE STARTING ADDRESS OF THE DIAGONAL
4636  027700  132737  000001  001734          BITB    #1,PASFLG       ;HAS THE READ OF THE MEMORY BEEN DONE ?
4637  027706  001333                          BNE     2$              ;IF NOT THEN GO TO 2$
4638  027710  062704  000002                  ADD     #2,R4           ;ADD 2 TO THE STARTING ADDRESS OF THE DIAGONAL
4639  027714  020402                          CMP     R4,R2           ;AND UNLESS THE END OF THE SEMIBANK (4K) IS REACHED
4640  027716  103003                          BHIS    9$              ;
4641  027720  105737  001734                  TSTB    PASFLG  ;OR THE DIAGONAL HAS BEEN ROTATED 64 TIMES
```

```
 4642  027724  100324                    BPL     2$              ;REPEAT FROM 2$
 4643  027726  013704  001736    9$:     MOV     LOWBNK,R4       ;MAKE R4 POINT TO THE LOWEST LOCATION IN THE
 4644                                                            ;IN THE SEMIBANK (4K) UNDER TEST
 4645  027732  000337  002174            SWAB    BAKPAT
 4646  027736  001717                    BEQ     2$              ;AND IF THE TEST HAS NOT BEEN PERFORMED WITH THE
 4647                                                            ;SWAPPED BACK GROUND PATTERN THEN GO TO 2$
 4648  027740  010204                    MOV     R2,R4           ;MAKE THE PRESENT HIGH BOUNDRY AS THE NEXT
 4649                                                            ;LOW BOUNDRY
 4650  027742  020205                    CMP     R2,R5           ;UNLESS THE PRESENT HIGH BOUNDRY IS ALSO THE
 4651                                                            ;HIGH BOUNDRY FOR THE MEMORY UNDER TEST
 4652  027744  103676                    BLO     1$
 4653  027746  000207                    RETURN
```

```
                                MTP024: SUBTST  <<MTP024          FAST GALLOPING PATTERN TEST>>
                                ;*****************************************************************************
                                ;*SUBTEST        MTP024  FAST GALLOPING PATTERN TEST
                                ;*****************************************************************************
                                ;*(1)    THIS TEST WRITES THE MEMORY WITH A BACK GROUND PATTERN
                                ;*      STORED AT LOCATION BAKPAT
                                ;*(2)    TEST BEGINS AT LOWEST LOCATION BEING TESTED
                                ;*      (LETS NAME IT 'A')
                                ;*(3)    LETS NAME THE 1ST LOCATION IN THE ROW/COLUMN UNDER TEST AS 'B'.
                                ;*(4)    SWAPS BYTES FOR LOCATION 'A'.
                                ;*(5)    READS 'A'  READS 'B'
                                ;*(6)    'B' = 'B'+200 (MAKES 'B'=64TH CELL I.E. 200TH OCTAL
                                ;*      LOCATION FROM THE PRESENT LOCATION OF 'B')
                                ;*(7)    REPEATS STEPS 5 AND 6 UNTIL 'B' IS GREATER THAN THE
                                ;*      END OF THE SEMIBANK (4K) OF THE MEMORY IN WHICH 'A' IS RESIDING
                                ;*(8)    A = A+2
                                ;*(9)    REPEATS  STEPS 3-8 UNTILL 'A' REACHES THE END OF THE SEMIBANK (4K)
                                ;*(10)   GOES TO THE NEXT SEMIBANK (4K) OF MEMORY AND REPEATS STEPS
                                ;*      3-9 UNTIL THE END OF THE BANK
                                ;*(11)   AFTER EXECUTING THE TEST BYTES ARE SWAPPED AT
                                ;*      LOCATION BAKPAT AND STEPS 1-10 ARE REPEATED
                                ;*(12)   IN THIS TEST R1 IS POINTING TO LOCATION 'A', R5 TO
                                ;*      LOCATION 'B', R2 TO THE END OF THE SEMIBANK (4K) IN WHICH THE
                                ;*      TEST IS TAKING PLACE AND R4 TO THE LOWEST LOCATION IN THE
                                ;*      COLUMN/ROW CONTAINING 'A' AND 'B'
027750  062702  020000        1$:     ADD     #20000,R2
027754  005037  001742        2$:     CLR     GALWORD
027760  010201                        MOV     R2,R1
027762  010204                        MOV     R2,R4
027764  162704  020000                SUB     #20000,R4       ;R4 = BOTTOM ADDRESS OF 4K SEMIBANK
027770  013741  002174        3$:     MOV     BAKPAT,-(R1)    ;WRITE THE MEMORY UNDER TEST WITH A BACKGROUND OF
                                                                ;BAKPAT
027774  020104                        CMP     R1,R4
027776  101374                        BHI     3$
030000  010405                4$:     MOV     R4,R5           ;R4 AND R5 ARE POINTING TO THE LOWEST LOCATION THAT
                                                                ;CAN BE TESTED IN THIS BLOCK
030002  011100                        MOV     (R1),R0
030004  023700  002174                CMP     BAKPAT,R0       ;BEFORE STARTING THE GALLOPING TEST FOR LOCATION
                                                                ;(R1) CHECK IT
030010  001401                        BEQ     5$              ;CONTINUE IF OK
030012  104447                        PERR23
030014  000311                5$:     SWAB    (R1)
030016  020105                6$:     CMP     R1,R5
030020  001405                        BEQ     7$
030022  011500                        MOV     (R5),R0
030024  020037  002174                CMP     R0,BAKPAT       ;CHECK THAT (R5) HAS BAKPAT IN IT
030030  001401                        BEQ     7$
030032  104450                        PERR24
030034  011100                7$:     MOV     (R1),R0
030036  020037  002176                CMP     R0,SWAPAT       ;CHECK THAT (R1) HAS SWAPPED BAKPAT IN IT
030042  001401                        BEQ     8$
030044  104447                        PERR23
030046  062705  000176        8$:     ADD     #176,R5
```

```
 4709  030052  020502                        CMP     R5,R2           ;AND IF R5 HAS NOT REACHED THE END OF THE BOUNDRY
 4710  030054  103760                        BLO     6$              ;THEN REPEAT FROM 6$
 4711  030056  000321                        SWAB    (R1)+           ;RESTORE THE LOCATION FOR WHICH THE GALLOPING TEST
 4712                                                                ;WAS BEING PERFORMED
 4713  030060  062704  000002                ADD     #2,R4           ;OTHERWISE INCREMENT R4 BY 2
 4714  030064  062737  000002  001742        ADD     #2,GALWORD              ;FOR EVERY ROW/COLUMN TESTED ADD 2
 4715  030072  105737  001742                TSTB    GALWORD
 4716  030076  100004                        BPL     9$              ;UNTIL GALWORD IS 200
 4717  030100  163704  001742                SUB     GALWORD,R4              ;SUBTRACT 200 FROM R4
 4718  030104  005037  001742                CLR     GALWORD
 4719  030110  020102                9$:     CMP     R1,R2           ;IF R1 HAS NOT REACHED THE END OF THE BOUNDRY
 4720  030112  103732                        BLO     4$              ;THEN REPEAT FROM 4$
 4721  030114  163704  001742                SUB     GALWORD,R4      ;RESTORE R4
 4722  030120  000337  002176                SWAB    SWAPAT
 4723  030124  000337  002174                SWAB    BAKPAT
 4724  030130  001711                        BEQ     2$              ;IF THE LOWER BYTE OF BAK..AT IS 0 THEN REPEAT FROM 2$
 4725  030132  010204                        MOV     R2,R4           ;OTHERWISE MAKE THE PRESENT HIGH BOUNDRY AS THE
 4726                                                                ;NEXT LOW BOUNDRY
 4727  030134  020227  160000                CMP     R2,#LAST+2
 4728  030140  001303                        BNE     1$              ;IF PREVIOUS HIGH BOUNDARY WAS THE END OF
 4729                                                                ;THE TEST BOUNDARY THEN EXIT
 4730                                                                ;OTHERWISE REPEAT TEST
 4731  030142  000207                        RETURN
```

```
 4734  030144                          MTP025: SUBTST  <<MTP025         INTERRUPT ENABLE TEST>>
  (3)                                  ;*****************************************************************************
  (3)                                  ;*SUBTEST        MTP025  INTERRUPT ENABLE TEST
  (2)                                  ;*****************************************************************************
 4735  030144  005037  001716                  CLR     TSTDAT          ;GENERATE CHECKBITS ON 0,,0
 4736  030150  005037  001720                  CLR     TSTDAT+2
 4737  030154  012737  001716  001760          MOV     #TSTDAT,SOURCE
 4738  030162  004737  034350                  CALL    CHKGEN
 4739  030166  012737  000003  001464          MOV     #3,NOPAR                ;SETUP PARITY ACTION
 4740  030174  013701  002070                  MOV     TESTADD,R1      ;FIRST TEST ADDRESS
 4741  030200  012737  030234  001750          MOV     #1$,PARTHERE    ;SETUP TRAP DESTINATION
 4742  030206  004737  030456                  CALL    MTPA25          ;WRITE DATA & CHECKBITS
 4743  030212  104473                          ECCIINIT                ;RESTORE CSR TO NORMAL
 4744  030214  005711                          TST     (R1)            ;ACCESS LOCATIONS FOR DBE TRAPS
 4745  030216  005761  000002                  TST     2(R1)
 4746  030222  104507                          ENA1SBE                 ;NONE - GOOD - ACCESS FOR SBE TRAPS
 4747  030224  005711                          TST     (R1)
 4748  030226  005761  000002                  TST     2(R1)
 4749  030232  000404                          BR      2$              ;NONE - GOOD - SKIP
 4750  030234  104426                  1$:     READCSR
 4751  030236                                  FATAL   27
  (1)  030236  005237  001454                  INC     FATAL$                  ;SET FATAL INDICATOR
  (1)  030242  104027                          ERROR   +27
 4752  030244  005237  001716          2$:     INC     TSTDAT          ;CHECK FOR CORRECT ACTION ON SBE'S
 4753  030250  004737  030404                  CALL    MTPD25          ;IN ALL 4 BYTES
 4754  030254  012737  000400  001716          MOV     #400,TSTDAT
 4755  030262  004737  030404                  CALL    MTPD25
 4756  030266  005037  001716                  CLR     TSTDAT
 4757  030272  005237  001720                  INC     TSTDAT+2
 4758  030276  004737  030404                  CALL    MTPD25
 4759  030302  012737  000400  001720          MOV     #400,TSTDAT+2
 4760  030310  004737  030404                  CALL    MTPD25
 4761
 4762  030314  005037  001720                  CLR     TSTDAT+2        ;CHECK FOR CORRECT ACTION ON DBE'S
 4763  030320  012737  000003  001716          MOV     #3,TSTDAT       ;IN ALL 4 BYTES
 4764  030326  004737  030426                  CALL    MTPE25
 4765  030332  012737  001400  001716          MOV     #1400,TSTDAT
 4766  030340  004737  030426                  CALL    MTPE25
 4767  030344  005037  001716                  CLR     TSTDAT
 4768  030350  012737  000003  001720          MOV     #3,TSTDAT+2
 4769  030356  004737  030426                  CALL    MTPE25
 4770  030362  012737  001400  001720          MOV     #1400,TSTDAT+2
 4771  030370  004737  030426                  CALL    MTPE25
 4772  030374  104503                          CLRICSR
 4773  030376  005037  001464                  CLR     NOPAR           ;INDICATE PARITY ACTION
 4774  030402  000207                          RETURN
 4775
 4776  030404  004737  030456          MTPD25: CALL    MTPA25          ;WRITE DATA & CHECKBITS
 4777  030410  104471                          ECCIDIS                 ;DISABLE PARITY TRAPS & CORRECTION
 4778  030412  004737  030476                  CALL    MTPB25          ;CHECK FOR NO TRAPS
 4779  030416  104507                          ENAISBE                 ;ENABLE SBE TRAPS
 4780  030420  004737  030530                  CALL    MTPC25          ;CHECK FOR EXPECTED TRAP
 4781  030424  000207                          RETURN
 4782
 4783  030426  004737  030456          MTPE25: CALL    MTPA25          ;WRITE DATA & CHECKBITS
 4784  030432  104471                          ECCIDIS                 ;DISABLE PARITY TRAPS & CORRECTION
```

```
                                        CALL    MTPB25          ;CHECK FOR NO TRAPS
                                        ECC1INIT                ;ENABLE DBE TRAPS
                                        CALL    MTPC25          ;CHECK FOR EXPECTED TRAP
                                        ENA1SBE                 ;ENABLE SBE TRAPS
                                        CALL    MTPC25          ;CHECK FOR EXPECTED TRAP
                                        RETURN

                                        ;WRITE TSTDAT & TSTDAT+2 & CHECKBITS
                                MTPA25: ECC1DIS                 ;DISABLE ECC
                                        MOV     TSTDAT,(R1)     ;WRITE FIRST 16 BITS
                                        CB1CSR                  ;CHECKBITS --> CSR
                                        MOV     TSTDAT+2,2(R1)  ;WRITE 2ND 16 BITS & CHECKBITS
                                        RETURN

                                        ;CHECK FOR NO TRAP OCCURING CONDITION
                                MTPB25: MOV     #1$,PARTHERE    ;SETUP TRAP DESTINATION
                                        TST     (R1)            ;ACCESS LOCATIONS
                                        TST     2(R1)
                                        RETURN                  ;NO TRAP - GOOD - RETURN

                                1$:     READCSR
                                        ERROR   +24
                                        SET     HEADER
  (1)                                   MOV     #-1,HEADER
                                        RETURN

                                        ;TRAP SHOULD OCCURE TEST
                                MTPC25: MOV     #1$,PARTHERE    ;SETUP TRAP DESTINATION
                                        TST     (R1)            ;ACCESS 1ST LOCATION
                                        BR      2$              ;NO TRAP - BAD NEWS - SKIP
                                1$:     MOV     #3$,PARTHERE    ;SETUP TRAP DESTINATION
                                        TST     2(R1)           ;ACCESS 2ND LOCATION
                                2$:     READCSR                 ;NO TRAP - BAD NEWS
                                        ERROR   +25
                                        SET     HEADER
  (1)                                   MOV     #-1,HEADER
                                3$:     RETURN
```

```
                                MTPA26: SUBTST  <<MTPA26         RANDOM DATA (WRITE)>>
                                ;*****************************************************************************
                                ;*SUBTEST        MTPA26  RANDOM DATA (WRITE)
                                ;*****************************************************************************
4824  030570  000137  172360    1$:     JMP     KDPAR0          ;V177640        GOTO V172360
4825  030574  010221            MOV     R2,(R1)+        ;V177644
4826  030576  010321            MOV     R3,(R1)+        ;V177646
4827  030600  077005            SOB     R0,1$           ;V177650
4828  030602  000207            RETURN                  ;V177652
4829
4830  030604                    MTPB26: SUBTST  <<MTPB26         RANDOM DATA (READ)>>
                                ;*****************************************************************************
                                ;*SUBTEST        MTPB26  RANDOM DATA (READ)
                                ;*****************************************************************************
4831                                    .DSABL  AMA
4832                                    .ENABL  LSB
4833  030604  000137  172360    1$:     JMP     @#KDPAR0        ;V177640        GOTO V172360
4834  030610  020221            CMP     R2,(R1)+        ;V177644
4835  030612  001401            BEQ     2$              ;V177646
4836  030614  104451            PERR25                  ;V177650
4837  030616  005127            2$:     COM     (PC)+           ;V177652
4838  030620  000000            RANODD: 0                       ;V177654        FOR ERROR REPORTING
4839  030622  020321            CMP     R3,(R1)+        ;V177656
4840  030624  001401            BEQ     3$              ;V177660
4841  030626  104451            PERR25                  ;V177662
4842  030630  005167  177764    3$:     COM     RANODD          ;V177664
4843  030634  077015            SOB     R0,1$           ;V177670
4844  030636  000207            RETURN                  ;V177672
4845                                    .DSABL  LSB
4846                                    .ENABL  AMA
4847
4848  030640                    MTPC26: SUBTST  <<RANDOM NUMBER SUBPROGRAM>>
                                ;*****************************************************************************
                                ;*SUBTEST        RANDOM NUMBER SUBPROGRAM
                                ;*****************************************************************************
4849                                    ;CALLER MUST SETUP
4850                                    ;       MOV     SEEDLO,R3
4851                                    ;       MOV     SEEDHI,R2
4852                                    ;       MOV     R3,R5
4853                                    ;       MOV     R2,R4
4854  030640  073427  000007            ASHC    #7,R4           ;V172360
4855  030644  060305            ADD     R3,R5           ;V172364
4856  030646  005504            ADC     R4              ;V172366
4857  030650  060204            ADD     R2,R4           ;V172370
4858  030652  062705  001057    ADD     #1057,R5        ;V172372
4859  030656  000730            BR      .-116           ;V172376        GOTO V172260
4860
4861  030660                    MTPD26: SUBTST  <<RANDOM NUMBER SUBSUBPROGRAM>>
                                ;*****************************************************************************
                                ;*SUBTEST        RANDOM NUMBER SUBSUBPROGRAM
                                ;*****************************************************************************
4862  030660  005504            ADC     R4              ;V172260
4863  030662  062704  047401    ADD     #47401,R4       ;V172262
4864  030666  010503            MOV     R5,R3           ;V172266
4865  030670  010402            MOV     R4,R2           ;V172270
4866  030672  000137  177644    JMP     UIPAR2          ;V172272        GOTO V177644
```

```
030676                          MTP030: SUBTST  <<MT0030        FLUSH OUT DBE'S>>
                                ;*******************************************************************
                                ;*SUBTEST        MT0030  FLUSH OUT DBE'S
                                ;*******************************************************************
030676  011002                  1$:     MOV     (R0),R2         ;V177640
030700  010220                          MOV     R2,(R0)+        ;V177642
030702  077103                          SOB     R1,1$           ;V177644
030704  000207                          RETURN                  ;V177646

030706                          MTP031: SUBTST  <<MTP031        SOB-A-LONG TEST>>
                                ;*******************************************************************
                                ;*SUBTEST        MTP031  SOB-A-LONG TEST
                                ;*******************************************************************
                                        .DSABL  AMA
030706  000000                          0                       ;MOVE TERMINATOR
030710  077001                  1$:     SOB     R0,1$           ;SOB TILL R0 UNDERFLOWS
030712  005167  177772                  COM     1$              ;WRITE COMPLEMENT OF SOB
030716  020167  177766                  CMP     R1,1$           ;READ & CHECK FOR NOT "SOB R0,DOT"
030722  001403                          BEQ     2$              ;OK - SKIP
030724  104454                          PERR30
030726  010167  177756                  MOV     R1,1$
030732  005167  177752          2$:     COM     1$              ;CORRECT SOB INSTRUCTION
030736  010200                          MOV     R2,R0           ;REINITIALIZE SOB CONSTANT
                                        ;UPDATE MOVE REGISTERS
030740  010503                          MOV     R5,R3
030742  005725                          TST     (R5)+           ;BUMP (SAFELY) BY 2
030744  010504                          MOV     R5,R4
030746  020527  160000                  CMP     R5,#LAST+2      ;DONE?
030752  001001                          BNE     3$              ;NO - SKIP
030754  000207                          RETURN                  ;YES

030756  014344                  3$:     MOV     -(R3),-(R4)
030760  001376                          BNE     3$
030762  000752                          BR      1$
        000056                  SOBLENGTH=.-MTP031
                                        .ENABL  AMA
```

```
                                        .TITLE  MISC SUBROUTINES

030764                          REGCOPY:SUBTST  <<SUBR  COPY R0 TO R4,R1 TO R3, & R2 TO R5>>
                                ;*****************************************************************************
                                ;*SUBTEST       SUBR    COPY R0 TO R4,R1 TO R3, & R2 TO R5
                                ;*****************************************************************************
030764  010004                          MOV     R0,R4
030766  010103                          MOV     R1,R3
030770  010205                          MOV     R2,R5
030772  000207                          RETURN

030774                          FLIPWARN:SUBTST <<FLIP WARNING CONSTANTS IN WORST CASE NOISE TESTS>>
                                ;*****************************************************************************
                                ;*SUBTEST       FLIP WARNING CONSTANTS IN WORST CASE NOISE TESTS
                                ;*****************************************************************************
030774                                  PUSH    R0
030774  010046                                                                  MOV R0,-(SP)
030776  005237  002160                  INC     FLIPLOC
031002  042737  177774  002160          BIC     #^C3,FLIPLOC
031010  022737  000001  002160          CMP     #1,FLIPLOC
031016  001414                          BEQ     1$
031020  022737  000002  002160          CMP     #2,FLIPLOC
031026  001413                          BEQ     2$
031030  022737  000003  002160          CMP     #3,FLIPLOC
031036  001414                          BEQ     3$
031040  005000                          CLR     R0
031042  013704  002154                  MOV     ONES,R4
031046  000414                          BR      4$
031050  005000                  1$:     CLR     R0
031052  005004                          CLR     R4
031054  000411                          BR      4$
031056  012700  000401          2$:     MOV     #401,R0
031062  013704  002154                  MOV     ONES,R4
031066  000404                          BR      4$
031070  012700  000401          3$:     MOV     #401,R0
031074  012704  000401                  MOV     #401,R4
031100  010037  022226          4$:     MOV     R0,WARN2
031104  010037  022242                  MOV     R0,WARN3
031110  010037  022266                  MOV     R0,WARN4
031114  010037  022302                  MOV     R0,WARN5
031120                                  POP     R0
031120  012600                                                                  MOV (SP)+,R0
031122  000207                          RETURN

031124                          BACKGND:SUBTST  <<SUBR  WRITE BACKGROUND>>
                                ;*****************************************************************************
                                ;*SUBTEST       SUBR    WRITE BACKGROUND
                                ;*****************************************************************************
                                        ;WRITES DATA FROM R2
031124  104415                          SAVREG
031126  012700  060000                  MOV     #FIRST,R0
031132  012701  040000                  MOV     #SIZE,R1
031136                                  BMOV    MTP000
031136  004537  037306                          JSR R5,BLOCK1
031142  022102                                  MTP000
031144  012737  000207  177644  WARN6:  MOV     #207,UIPAR2             ;WARNING PUTTING "RETURN" INSTRUCTION AFTER WRIT
```

```
  4942  031152  004737  021546            CALL    SUPD01
  4943  031156  104416                    RESREG
  4944  031160  000207                    RETURN
```

```
                                PCONFIG:SUBTST  <<SUBR  PRINT CONFIGURATION MAP>>
                                ;*****************************************************************************
                                ;*SUBTEST        SUBR    PRINT CONFIGURATION MAP
                                ;*****************************************************************************
                                        PUSH    TKVEC,TKVEC+2,R0
                                                                                        MOV TKVEC,-(SP)
                                                                                        MOV TKVEC+2,-(SP)
                                                                                        MOV R0,-(SP)
                                        MOV     SP,PCONFS               ;SAVE LAST GOOD SP
                                        MOV     #PCONF2,TKVEC
                                        MOV     #340,TKVEC+2
                                        MOV     @$TKB,R0                ;KILL ANY OLD INTERRUPT
                                        BIC     #BIT7,PSW               ;LOWER CPU PRIORITY TO 140
                                        BIS     #BIT6,@$TKS             ;ENABLE KEYBOARD INTERRUPTS

                                        TYPE    MSG001
                                        TYPEIT  ,MSG001
                                        TYPE    MSG002
                                        TYPEIT  ,MSG002
                                        TYPE    MSG003
                                        TYPEIT  ,MSG003
                                        ;IF FAT PAPER ON TERMINAL GOTO 1$
                                        IF #SW4 SET.IN @SWR THEN JUMPTO PCONF1
                                                                                        BIT #SW4,@SWR
                                                                                        BEQ L271
                                                                                        JMP PCONF1
                                                                                L271:;;;;;;
                                        MOV     #60.,R0
                                        MOV     R0,R4
                                        CLR     R1
                                        CLR     R3
                                        TYPE    MSG004
                                        TYPEIT  ,MSG004
                                        CALL    TCONFIG                 ;GO TYPE CONFIGURATION (1ST HALF)
                                        TYPE    $CRLF
                                        TYPEIT  ,$CRLF
                                        TYPE    MSG017                  ;PRINT SPACE(S)
                                        TYPEIT  ,MSG017
                                        TYPE    MSG011
                                        TYPEIT  ,MSG011
                                        TYPE    $CRLF
                                        TYPEIT  ,$CRLF
                                        TYPE    MSG017                  ;PRINT SPACE(S)
                                        TYPEIT  ,MSG017
                                        TYPE    MSG012
                                        TYPEIT  ,MSG012
                                        MOV     #60.*2*3,R1
                                        MOV     R1,R3
                                        CALL    TCONFIG
                                        BR      PCONF2

                                PCONF1: MOV     #120.,R0
                                        MOV     R0,R4
                                        CLR     R1
                                        CLR     R3
                                        TYPE    MSG014                  ;SPACE
```

```
  (1)  031364  104401  067047                  TYPEIT  ,MSG014
 4983  031370                                  TYPE    MSG011
  (1)  031370  104401  066662                  TYPEIT  ,MSG011
 4984  031374                                  TYPE    MSG004
  (1)  031374  104401  066436                  TYPEIT  ,MSG004
 4985  031400                                  TYPE    MSG012
  (1)  031400  104401  066750                  TYPEIT  ,MSG012
 4986  031404  004737  031444                  CALL    TCONFIG
 4987
 4988  031410  013706  031442          PCONF2: MOV     PCONFS,SP                ;RESTORE STACK
 4989  031414  042777  000100  150562          BIC     #BIT6,@$TKS
 4990  031422  117700  150560                  MOVB    @$TKB,R0                 ;READ CHAR TO KILL FLAG
 4991  031426                                  POP     R0,TKVEC+2,TKVEC
  (2)  031426  012600                                                                  MOV (SP)+,R0
  (3)  031430  012637  000062                                                          MOV (SP)+,TKVEC+2
  (4)  031434  012637  000060                                                          MOV (SP)+,TKVEC
 4992  031440  000207                          RETURN
 4993
 4994  031442  000000                  PCONFS: 0                                ;STACK SAVED HERE!
```

```
                                        SUBTST  <<SUBR  TYPE CONFIGURATION>>
                                ;*****************************************************************************
                                ;*SUBTEST        SUBR    TYPE CONFIGURATION
                                ;*****************************************************************************
                                ;*****************************************************************************
                                ;CALL:   MOV     #N,R0           ;N=NUMBER OF CHARACTERS
                                ;        MOV     R0,R4           ;BACKUP
                                ;        MOV     #K,R1           ;INDEX CONSTANT
                                ;        MOV     R1,R3           ;BACKUP
                                ;        CALL    TCONFIG         ;ACTUAL CALL
                                ;        RETURN                  ;ONLY RETURN
                                ;*****************************************************************************

                                        ;************
                                        ;** ERROR **
                                        ;************
031444                          TCONFIG:TYPE    MSG005
(1) 031444  104401  066544              TYPEIT  ,MSG005
031450  032761  004000  002230  1$:     BIT     #BIT11,CONFIG(R1)       ;ERROR ON THIS BANK?
031456  001403                          BEQ     2$                      ;NO - SKIP
031460                                  TYPE    MSG013                  ;PRINT "X"
(1) 031460  104401  067045              TYPEIT  ,MSG013
031464  000402                          BR      3$
031466                          2$:     TYPE    MSG014                  ;PRINT SPACE
(1) 031466  104401  067047              TYPEIT  ,MSG014
031472  062701  000006          3$:     ADD     #6,R1                   ;BUMP POINTER
031476  077014                          SOB     R0,1$                   ;LOOP TILL DONE
031500  010400                          MOV     R4,R0
031502  010301                          MOV     R3,R1

                                        ;************
                                        ;**  CPU   *
                                        ;************
031504                                  TYPE    MSG008
(1) 031504  104401  066624              TYPEIT  ,MSG008
031510  016105  002230          4$:     MOV     CONFIG(R1),R5
031514  072527  177774                  ASH     #-4,R5                  ;GET CPU BITS
031520  042705  177760                  BIC     #^C17,R5                ;CLEAR NON INTERESTING BITS
031524  022705  000012                  CMP     #10.,R5
031530  100003                          BPL     8$
031532  062705  000067                  ADD     #67,R5                  ;MAKE NUMBERS OVER 9 HEX
031536  000402                          BR      9$
031540  062705  000060          8$:     ADD     #60,R5                  ;MAKE ASCII
031544  110537  067051          9$:     MOVB    R5,MSG015               ;PLUG INTO MEMORY
031550                                  TYPE    MSG015
(1) 031550  104401  067051              TYPEIT  ,MSG015
031554  062701  000006                  ADD     #6,R1                   ;BUMP POINTER
031560  077025                          SOB     R0,4$                   ;LOOP TILL DONE
031562  010400                          MOV     R4,R0
031564  010301                          MOV     R3,R1
```

```
                                                ;*****************
                                                ;** INTERLEAVE **
                                                ;*****************
                                                TYPE    MSG007
                                                TYPEIT  ,MSG007
                                                ;THIS IS AN ENTRY POINT FROM ERROR REPORTS
031572  016105  002230          TCFIG1: MOV     CONFIG(R1),R5
031576  072527  177763                  ASH     #-13.,R5                ;GET MEMORY TYPE
031602  042705  177774                  BIC     #^C3,R5                 ;CLEAR NON INTERESTING BITS
031606  005705                          TST     R5
031610  001004                          BNE     1$
031612  112737  000077  067051          MOVB    #'?,MSG015
031620  000435                          BR      7$
031622  016105  002230          1$:     MOV     CONFIG(R1),R5
031626  042705  177770                  BIC     #^C7,R5                 ;GET INTERLEAVE BITS
031632  022705  000001                  CMP     #1,R5
031636  001002                          BNE     2$
031640  006305                          ASL     R5                      ;2 TO 1
031642  000420                          BR      5$
031644  022705  000002          2$:     CMP     #2,R5
031650  001002                          BNE     3$
031652  006305                          ASL     R5                      ;4 TO 1
031654  000413                          BR      5$
031656  022705  000004          3$:     CMP     #4,R5
031662  001003                          BNE     4$
031664  012705  000010                  MOV     #8.,R5                  ;8 TO 1
031670  000405                          BR      5$
031672  005705                  4$:     TST     R5
031674  001403                          BEQ     5$
031676  012705  000077                  MOV     #'?,R5
031702  000402                          BR      6$
031704  062705  000060          5$:     ADD     #60,R5                  ;MAKE ASCII
031710  110537  067051          6$:     MOVB    R5,MSG015               ;PLUG INTO MEMORY
031714                          7$:     TYPE    MSG015
031714  104401  067051                  TYPEIT  ,MSG015
031720                                  IF NOTAB NE #0 THEN $RETURN
031720  005737  002044                                                  TST NOTAB
031724  001401                                                          BEQ L272
031726  000207                                                          RTS PC
031730                                                          L272:::::::
031730  062701  000006                  ADD     #6,R1                   ;BUMP POINTER
031734  077062                          SOB     R0,TCFIG1               ;LOOP TILL DONE
031736  010400                          MOV     R4,R0
031740  010301                          MOV     R3,R1

                                                ;******************
                                                ;** MEMORY TYPE **
                                                ;******************
                                                .ENABL  LSB
031742                                          TYPE    MSG009
031742  104401  066636                  TYPEIT  ,MSG009
031746  016105  002230          8$:     MOV     CONFIG(R1),R5
031752  072527  177763                  ASH     #-13.,R5                ;GET MEMORY TYPE
031756  042705  177774                  BIC     #^C3,R5                 ;CLEAR NON INTERESTING BITS
031762                                  IF R5 EQ #1
031762  020527  000001                                                  CMP R5,#1
```

```
  (6)  031766  001004                                                          BNE L273
5092   031770  112737  000113  067051          MOVB  #'K,MSG015
5093   031776                                ELSE
  (3)  031776  000413                                                          BR L274
  (3)  032000                                                          L273:;;;;;;;
5094   032000                                  IF #BIT7!BIT6!BIT5!BIT4 SET.IN CONFIG(R1)
  (4)  032000  032761  000360  002230                                          BIT #BIT7!BIT6!BIT5!BIT4
  (6)  032006  001404                                                          BEQ L275
5095   032010  112737  000112  067051            MOVB        #'J,MSG015
5096   032016                                  ELSE
  (3)  032016  000403                                                          BR L276
  (3)  032020                                                          L275:;;;;;;;
5097   032020  112737  000060  067051            MOVB        #'0,MSG015
5098   032026                                  END ;OF IF #BIT7!BIT6!BIT5!BIT4
  (3)  032026                                                          L276:;;;;;;;
5099   032026                                END ;OF IF R5
  (3)  032026                                                          L274:;;;;;;;
5100   032026                                TYPE    MSG015
  (1)  032026  104401  067051                TYPEIT  ,MSG015
5101   032032  062701  000006                ADD     #6,R1               ;BUMP POINTER
5102   032036  077035                        SOB     R0,8$               ;LOOP TILL DONE
5103   032040  010400                        MOV     R4,R0
5104   032042  010301                        MOV     R3,R1
5105                                         .DSABL  LSB
5106
5107                                         ;*********
5108                                         ;** BOX **
5109                                         ;*********
5110   032044                                TYPE    MSG016
  (1)  032044  104401  067053                TYPEIT  ,MSG016
5111   032050  016105  002230        9$:     MOV     CONFIG(R1),R5
5112   032054  032705  000010                BIT     #BIT3,R5            ;EXTERNAL INTERLEAVE?
5113   032060  001404                        BEQ     10$                 ;NO - SKIP
5114   032062  112737  000115  067051        MOVB    #'M,MSG015          ;INDICATE MULTIPLE BOXES
5115   032070  000417                        BR      12$
5116   032072  032705  020014        10$:    BIT     #BIT13!14,R5        ;MJ11?
5117   032076  001004                        BNE     11$                 ;NO - SKIP
5118   032100  112737  000077  067051        MOVB    #'?,MSG015          ;I DON'T KNOW HOW MJ11'S INTERLEAVE
5119   032106  000410                        BR      12$
5120   032110  072527  177770        11$:    ASH     #-8.,R5             ;GET BOX NUMBER
5121   032114  042705  177774                BIC     #^C3,R5             ;CLEAR NON INTERESTING BITS
5122   032120  062705  000060                ADD     #60,R5              ;MAKE ASCII
5123   032124  110537  067051                MOVB    R5,MSG015           ;PLUG INTO MEMORY
5124   032130                        12$:    TYPE    MSG015
  (1)  032130  104401  067051                TYPEIT  ,MSG015
5125   032134  062701  000006                ADD     #6,R1               ;BUMP POINTER
5126   032140  077035                        SOB     R0,9$               ;LOOP TILL DONE
5127   032142  010400                        MOV     R4,R0
5128   032144  010301                        MOV     R3,R1
5129
5130                                         ;***************
5131                                         ;** PROTECTED **
5132                                         ;***************
5133   032146                                TYPE    MSG010
  (1)  032146  104401  066650                TYPEIT  ,MSG010
5134   032152  005761  002230        13$:    TST     CONFIG(R1)          ;BANK PROTECTED?
```

```
 5135  032156  100004                        BPL     14$                     ;NO - SKIP
 5136  032160  112737  000120  067051        MOVB    #'P,MSG015
 5137  032166  000407                        BR      15$
 5138  032170  032761  010000  002230  14$:  BIT     #BIT12,CONFIG(R1)       ;PROTECTED REGION OF MK11 BOX?
 5139  032176  001406                        BEQ     16$                     ;NO - SKIP
 5140  032200  112737  000113  067051        MOVB    #'K,MSG015
 5141  032206                          15$:  TYPE    MSG015
   (1) 032206  104401  067051                TYPEIT  ,MSG015
 5142  032212  000402                        BR      17$
 5143  032214                          16$:  TYPE    MSG014                  ;PRINT SPACE
   (1) 032214  104401  067047                TYPEIT  ,MSG014
 5144  032220  062701  000006          17$:  ADD     #6,R1                   ;BUMP POINTER
 5145  032224  077026                        SOB     R0,13$                  ;LOOP TILL DONE
 5146  032226  010400                        MOV     R4,R0
 5147  032230  010301                        MOV     R3,R1
 5148  032232  000207                        RETURN
```

```
                        .SBTTL  TRAP    PARITY ERROR HANDLER
                ;*****************************************************************************
                ;VECTOR TO HERE FROM TRAPS TO 114
                ;IGNORE ERRORS BUT COUNT IF NOPAR FLAG = 1.
                ;*****************************************************************************
                ;
                ;       CODE    ACTION
                ;       ++++    ++++++
                ;         0     PRINT UNEXPECTED PARITY TRAP
                ;         1     COUNT ERROR & SAVE 1ST IN "PADDRESS"
                ;         2     SET "ABORT" / SETUP "BADPC" / RETURN VIA PCBUMP
                ;         3     RETURN VIA "PARTHERE"
                ;         4     COUNT ERROR & SAVE 1ST IN "PADDRESS" REGARDLESS OF ERROR REGISTER
                ;
032234  022737  000001  001464  PARITY: CMP     #1,NOPAR                ;COUNTING PARITY ERRORS?
032242  001023                          BNE     3$                      ;NO - SKIP
032244  032737  000001  177744          BIT     #BIT0,MEMERR            ;MEMORY TIMEOUT?
032252  001402                          BEQ     1$                      ;NO - SKIP
032254  000137  032422                  JMP     NONEXIST                ;YES - SIMULATE A TRAP TO 4
032260  005237  001460          1$:     INC     PARCNT                  ;PARITY ERROR COUNTER + 1
032264  022737  000001  001460          CMP     #1,PARCNT               ;FIRST PARITY ERROR?
032272  001003                          BNE     2$                      ;NO - SKIP
032274  013737  177740  001424          MOV     LOADRS,PADDRESS         ;SAVE 1ST BAD PARITY ADDRESS
032302  013737  177744  177744  2$:     MOV     MEMERR,MEMERR           ;CLEAR ERROR BITS
032310  000002                          RTI
032312  022737  000002  001464  3$:     CMP     #2,NOPAR                ;ACTION CODE = 2 ?
032320  001014                          BNE     6$                      ;NO - SKIP
032322                                  SET     ABORTFLAG               ;YES
  (1)   032322  012737  177777  001524          MOV     #-1,ABORTFLAG
032330  011637  001420                  MOV     (SP),BADPC
032334  063716  001764                  ADD     PCBUMP,(SP)             ;UPDATE RETURN PC
032340  013737  177744  177744          MOV     MEMERR,MEMERR           ;CLEAR ERROR BITS
032346  000244                          CLZ                             ;SHOW FAILURE BY .NE.
032350  000002                          RTI
032352  022737  000003  001464  6$:     CMP     #3,NOPAR                ;ACTION CODE = 3 ?
032360  001006                          BNE     7$                      ;NO - SKIP
032362  013716  001750                  MOV     PARTHERE,(SP)
032366  013737  177744  177744          MOV     MEMERR,MEMERR           ;CLEAR ERROR BITS
032374  000002                          RTI
032376  022737  000004  001464  7$:     CMP     #4,NOPAR
032404  001001                          BNE     8$
032406  000724                          BR      1$
                ;
032410  011637  001420          8$:     MOV     (SP),BADPC
032414                                  FATAL   32
  (1)   032414  005237  001454          INC     FATAL$                  ;SET FATAL INDICATOR
  (1)   032420  104032                  ERROR   +32
```

```
                                        .SBTTL  TRAP    NON-EXISTANT MEMORY (HOLES) HANDLER
                                ;*******************************************************************************
                                ;VECTOR TO HERE (SOMETIMES) FROM TRAPS TO 4
                                ;       CODE IN NONEM DETERMINES ACTION AS FOLLOWS:
                                ; 1)    IGNORE ERRORS BUT COUNT IF NONEM (NO NON-EXISTANT MEMORY) FLAG = 1.
                                ; 2)    TO EXIT PATTERN 0 DURING SIZING IF NON-EXIST MEM ERROR
                                ;*******************************************************************************
                                ;
032422  022737  000001  001466  NONEXIST:CMP     #1,NONEM                ;COUNTING NON-EXISTANT MEMORY ERRORS?
032430  001011                          BNE     2$                      ;NO - SKIP
032432  005237  001456                  INC     NEMCNT                  ;BUMP NON-EXISTANT MEMORY COUNTER
032436  022737  000001  001456          CMP     #1,NEMCNT               ;FIRST ERROR?
032444  001002                          BNE     1$                      ;NO - SKIP
032446  010037  001422                  MOV     R0,ADDRESS              ;ASSUME R0 CONTAINS THE ADDRESS ACCESSED
032452  000002                  1$:     RTI
032454  005237  001456          2$:     INC     NEMCNT                  ;BUMP NON-EXISTANT MEMORY COUNTER
032460  012701  000001                  MOV     #1,R1                   ;DUMMY UP R1 FOR A FORCED SOB EXIT
032464  000002                          RTI
                                ;*******************************************************************************
                                        .SBTTL  TRAP    TIMEOUT (TRAP TO 4) HANDLER
032466  011637  001420          TIMEOUT:MOV     (SP),BADPC
032472                                  FATAL   6
032472  005237  001454                  INC     FATAL$                  ;SET FATAL INDICATOR
032476  104006                          ERROR   +6
                                ;*******************************************************************************
                                        .SBTTL  TRAP    MEMORY MANAGEMENT (TRAP TO 250) HANDLER
032500  011637  001420          MMTRAP: MOV     (SP),BADPC
032504                                  FATAL   7
032504  005237  001454                  INC     FATAL$                  ;SET FATAL INDICATOR
032510  104007                          ERROR   +7
                                        .SBTTL  TRAP    RESERVED INSTRUCTION HANDLER
032512  011637  001422          PDP1105:MOV     (SP),ADDRESS            ;FIND OLD PC
032516  162737  000002  001422          SUB     #2,ADDRESS              ;CORRECT FOR BAD INSTRUCTION
032524  013737  001422  001420          MOV     ADDRESS,BADPC           ;NOTE * ALL ILLEGAL INSTRUCTIONS ARE CONSIDERED
                                                                        ;SINGLE WORD INSTRUCTIONS BY PDP 11'S
032532  017737  146664  001440          MOV     @ADDRESS,BAD            ;GET BAD INSTRUCTION
032540                                  FATAL   5
032540  005237  001454                  INC     FATAL$                  ;SET FATAL INDICATOR
032544  104005                          ERROR   +5
```

```
                                        .SBTTL  TRAP    KERNEL TRAP HANDLER
                                        ;***************************************************************************
                                        ;KERNEL IS A TRAP THAT COMES HERE
                                        ;***************************************************************************
032546  042766  140000  000002  $KERNEL:      BIC     #140000,2(SP)
032554  000002                        RTI
                                        ;***************************************************************************
                                        .SBTTL  TRAP    ENERGIZE TRAP HANDLER
032556  052737  001001  177572  $ENERGIZE:BIS   #BIT9!BIT0,MMR0
032564  000002                        RTI
                                        ;***************************************************************************
                                        .SBTTL  TRAP    DEENERGIZE TRAP HANDLER
032566  042737  001001  177572  $DEENERGIZE:BIC #BIT9!BIT0,MMR0
032574  000002                        RTI
                                        ;***************************************************************************
                                        .SBTTL  TRAP    CACHON TRAP HANDLER
032576  013737  002132  177746  $CACHN: MOV     CACHK,CONTRL    ;SETUP CACHE AS PER CONSTANT (USUALLY 1 = FULLY ON)
032604  052737  000001  177746          BIS     #BIT0,CONTRL    ;DISABLE TRAPS (BUT NOT ABORTS)
032612  000002                        RTI
                                        ;***************************************************************************
                                        .SBTTL  TRAP    CACHOFF TRAP HANDLER
                                        ;DISABLE TRAPS (NOT ABORTS), FORCE MISSES, FLUSH, BYPASS
032614  052737  001415  177746  $CACHF: BIS     #BIT0!BIT2!BIT3!BIT8!BIT9,CONTRL
032622  000002                        RTI
```

```
                                                .SBTTL  TRAP    LOAD CSR TRAP HANDLER
                                                ;LOAD CORRECT CSR WITH DATA IN CSR & CSR+2
032624                                  $LOADC: PUSH    R0
032624  010046                                                                                  MOV R0,-(SP)
032626  012700  172100                          MOV     #MK11ADD,R0                 ;CREATE CSR ADDRESS
032632  063700  001634                          ADD     CSRNO,R0
032636  005737  001632                          TST     CSRREL                      ;CSR'S RELOCATED?
032642  001036                                  BNE     $LOAD1                      ;YES - SKIP
032644  013720  001526                          MOV     CSR,(R0)+                   ;LOAD IT
                                                ;THE FOLLOWING LINES INSURE THAT THE "INTERNAL INTERLEAVE" DON'T CHANGE
032650                                          IF INTOVR EQ #0
032650  005737  002042                                                                          TST INTOVR
032654  001025                                                                                  BNE L277
032656                                            PUSH  R1,R2
032656  010146                                                                                  MOV R1,-(SP)
032660  010246                                                                                  MOV R2,-(SP)
032662  013701  001634                            MOV   CSRNO,R1
032666  072127  177776                            ASH   #-2,R1
032672  005201                                    INC   R1                  ;R1 = (1-8)
032674  013702  001706                            MOV   MKCSRS,R2
032700  006102                          1$:       ROL   R2
032702  077102                                    SOB   R1,1$
032704                                            ON.ERROR
032704  103004                                                                                  BCC L300
032706  052737  004000  001530                      BIS #BIT11,CSR+2
032714                                            ELSE
032714  000403                                                                                  BR L301
032716                                                                                  L300:;;;;;;
032716  042737  004000  001530                      BIC #BIT11,CSR+2
032724                                            END ;OF ON.ERROR
032724                                                                                  L301:;;;;;;
032724                                            POP   R2,R1
032724  012602                                                                                  MOV (SP)+,R2
032726  012601                                                                                  MOV (SP)+,R1
032730                                          END ;OF IF INTOVR
032730                                                                                  L277:;;;;;;
032730  013710  001530                          MOV     CSR+2,(R0)
032734                                          POP     R0
032734  012600                                                                                  MOV (SP)+,R0
032736  000002                                  RTI
032740  042700  020000                  $LOAD1: BIC     #20000,R0
032744  013737  177746  001746                  MOV     CONTRL,OLDCACHE             ;SAVE CACHE STATUS
032752  104424                                  CACHOFF
032754  013720  001526                          MOV     CSR,(R0)+                   ;LOAD IT
                                                ;THE FOLLOWING LINES INSURE THAT THE "INTERNAL INTERLEAVE" DON'T CHANGE
032760                                          IF INTOVR EQ #0
032760  005737  002042                                                                          TST INTOVR
032764  001025                                                                                  BNE L302
032766                                            PUSH  R1,R2
032766  010146                                                                                  MOV R1,-(SP)
032770  010246                                                                                  MOV R2,-(SP)
032772  013701  001634                            MOV   CSRNO,R1
032776  072127  177776                            ASH   #-2,R1
033002  005201                                    INC   R1                  ;R1 = (1-8)
033004  013702  001706                            MOV   MKCSRS,R2
033010  006102                          3$:       ROL   R2
```

```
5297  033012  077102                          SOB   R1,3$
5298  033014                                  ON.ERROR
 (8)  033014  103004                                                    BCC L303
5299  033016  052737  004000  001530            BIS #BIT11,CSR+2
5300  033024                                  ELSE
 (3)  033024  000403                                                    BR L304
 (3)  033026                                                          L303:::::::
5301  033026  042737  004000  001530            BIC #BIT11,CSR+2
5302  033034                                  END ;OF ON.ERROR
 (3)  033034                                                          L304:::::::
5303  033034                                  POP   R2,R1
 (2)  033034  012602                                                    MOV (SP)+,R2
 (3)  033036  012601                                                    MOV (SP)+,R1
5304  033040                                  END ;OF IF INTOVR
 (3)  033040                                                          L302:::::::
5305  033040  013710  001530                  MOV     CSR+2,(R0)
5306  033044  013737  001746  177746          MOV     OLDCACHE,CONTRL         ;RESTORE CACHE
5307  033052                                  POP     R0
 (2)  033052  012600                                                    MOV (SP)+,R0
5308  033054  000002                          RTI
5309                                          .SBTTL  TRAP    READ CSR TRAP HANDLER
5310                                          ;READ THE CORRECT CSR INTO LOCATIONS CSR & CSR+2
5311  033056                          $READC: PUSH    R0
 (2)  033056  010046                                                    MOV R0,-(SP)
5312  033060  012700  172100                  MOV     #MK11ADD,R0             ;CREATE CSR ADDRESS
5313  033064  063700  001634                  ADD     CSRNO,R0
5314  033070  005737  001632                  TST     CSRREL                  ;CSR'S RELOCATED?
5315  033074  001005                          BNE     1$                      ;YES - SKIP
5316  033076  012037  001526                  MOV     (R0)+,CSR               ;READ IT
5317  033102  011037  001530                  MOV     (R0),CSR+2              ;READ IT
5318  033106  000415                          BR      2$
5319  033110  042700  020000          1$:     BIC     #20000,R0
5320  033114  013737  177746  001746          MOV     CONTRL,OLDCACHE         ;SAVE CACHE STATUS
5321  033122  104424                          CACHOFF
5322  033124  012037  001526                  MOV     (R0)+,CSR               ;READ IT
5323  033130  011037  001530                  MOV     (R0),CSR+2              ;READ IT
5324  033134  013737  001746  177746          MOV     OLDCACHE,CONTRL         ;RESTORE CACHE
5325  033142                          2$:     POP     R0
 (2)  033142  012600                                                    MOV (SP)+,R0
5326  033144  000002                          RTI
5327
5328                                          .SBTTL  TRAP    TEST (R1) & READ CSR CAREFULLY
5329  033146                          $TSTRD: PUSH    R0
 (2)  033146  010046                                                    MOV R0,-(SP)
5330  033150                                  BMOV    TSTRD1
 (1)  033150  004537  037306                        JSR R5,BLOCK1
 (1)  033154  033262                              TSTRD1
5331  033156  012700  172100                  MOV     #MK11ADD,R0             ;CREATE CSR ADDRESS
5332  033162  063700  001634                  ADD     CSRNO,R0
5333  033166  005737  001632                  TST     CSRREL                  ;CSR'S RELOCATED?
5334  033172  001003                          BNE     1$                      ;YES - SKIP
5335  033174  004737  177640                  CALL    FASTCITY
5336  033200  000413                          BR      2$
5337  033202  042700  020000          1$:     BIC     #20000,R0
5338  033206  013737  177746  001746          MOV     CONTRL,OLDCACHE         ;SAVE CACHE STATUS
5339  033214  104424                          CACHOFF
```

```
5340  033216  004737  177640              CALL    FASTCITY
5341  033222  013737  001746  177746      MOV     OLDCACHE,CONTRL          ;RESTORE CACHE
5342                                      ;IF SINGLE BIT ERROR SET CARRY BIT
5343  033230                      2$:     POP     R0                                      MOV (SP)+,R0
  (2) 033230  012600
5344  033232                              IF #BIT4 SET.IN CSR                             BIT #BIT4,CSR
  (4) 033232  032737  000020  001526                                                      BEQ L305
  (6) 033240  001404
5345  033242  052766  000001  000002        BIS   #BIT0,2(SP)
5346  033250                              ELSE                                            BR L306
  (3) 033250  000403
  (3) 033252                                                                    L305:;;;;;;
5347  033252  042766  000001  000002        BIC   #BIT0,2(SP)
5348  033260                              END ;OF IF #BIT4
  (3) 033260                                                                    L306:;;;;;;
5349  033260  000002                      RTI
5350
5351  033262  005010                      CLR     (R0)                    ;V177640
5352  033264                              SUPERVISOR                      ;V177642
  (1) 033264  052737  040000  177776      BIS     #BIT14,PSW              ;DO IT TO IT!
5353  033272  105711                      TSTB    (R1)                    ;V177646
5354  033274  042737  140000  177776      BIC     #BIT15!BIT14,PSW        ;V177650
5355  033302  012037  001526              MOV     (R0)+,CSR               ;V177656
5356  033306  011037  001530              MOV     (R0),CSR+2              ;V177662
5357  033312  000207                      RETURN                          ;V177666
```

```
 5360                                          .SBTTL  TRAP    ECC DISABLE ALL CSR'S TRAP HANDLER
 5361  033314  012737  000002  001526  $ECCDIS:MOV     #BIT1,CSR
 5362  033322                                IF KPFLAG NE #0 THEN LET CSR := CSR SET.BY #BIT3
  (4)  033322  005737  001504                                                          TST KPFLAG
  (6)  033326  001403                                                                  BEQ L307
  (6)  033330  052737  000010  001526                                                  BIS #BIT3,CSR
  (3)  033336                                                                L307:;;;;;;;
 5363  033336  005037  001530                CLR     CSR+2
 5364  033342  004737  034302                CALL    CSROUT
 5365  033346  000002                        RTI
 5366                                          .SBTTL  TRAP    ECC DISABLE OF 1 SELECTED CSR TRAP HANDLER
 5367  033350  012737  000002  001526  $ECC1DIS:MOV    #BIT1,CSR
 5368  033356                                IF KPFLAG NE #0 THEN LET CSR := CSR SET.BY #BIT3
  (4)  033356  005737  001504                                                          TST KPFLAG
  (6)  033362  001403                                                                  BEQ L310
  (6)  033364  052737  000010  001526                                                  BIS #BIT3,CSR
  (3)  033372                                                                L310:;;;;;;;
 5369  033372  005037  001530                CLR     CSR+2
 5370  033376  104425                        LOADCSR
 5371  033400  000002                        RTI
 5372                                          .SBTTL  TRAP    MK11 INITIALIZE ALL CSR'S TRAP HANDLER
 5373  033402  012737  000001  001526  $ECCINIT:MOV    #BIT0,CSR
 5374  033410  005037  001530                CLR     CSR+2
 5375  033414  004737  034302                CALL    CSROUT
 5376  033420  000002                        RTI
 5377                                          .SBTTL  TRAP    MK11 INITIALIZE 1 SELECTED CSR TRAP HANDLER
 5378  033422  012737  000001  001526  $ECC1INIT:MOV   #BIT0,CSR
 5379  033430  005037  001530                CLR     CSR+2
 5380  033434  104425                        LOADCSR
 5381  033436  000002                        RTI
 5382                                          .SBTTL  TRAP    ENABLE SBE PARITY TRAPS ON ALL CSR'S
 5383  033440  012737  000003  001526  $ENASBE:MOV     #BIT0!BIT1,CSR
 5384  033446                                IF KPFLAG NE #0 THEN LET CSR := CSR SET.BY #BIT3
  (4)  033446  005737  001504                                                          TST KPFLAG
  (6)  033452  001403                                                                  BEQ L311
  (6)  033454  052737  000010  001526                                                  BIS #BIT3,CSR
  (3)  033462                                                                L311:;;;;;;;
 5385  033462  005037  001530                CLR     CSR+2
 5386  033466  004737  034302                CALL    CSROUT
 5387  033472  000002                        RTI
 5388                                          .SBTTL  TRAP    ENABLE SBE PARITY TRAPS ON 1 SELECTED CSR
 5389  033474  012737  000003  001526  $ENA1SBE:MOV    #BIT0!BIT1,CSR
 5390  033502                                IF KPFLAG NE #0 THEN LET CSR := CSR SET.BY #BIT3
  (4)  033502  005737  001504                                                          TST KPFLAG
  (6)  033506  001403                                                                  BEQ L312
  (6)  033510  052737  000010  001526                                                  BIS #BIT3,CSR
  (3)  033516                                                                L312:;;;;;;;
 5391  033516  005037  001530                CLR     CSR+2
 5392  033522  104425                        LOADCSR
 5393  033524  000002                        RTI
 5394                                          .SBTTL  TRAP    WRITE CHECKBITS THRU ALL CSR'S TRAP HANDLER
 5395  033526  013737  001762  001526  $CBCSR: MOV     CHECK,CSR                   ;BITS 14-8
 5396  033534  052737  000006  001526          BIS     #BIT1!BIT2,CSR              ;CHECK MODE
 5397  033542                                IF KPFLAG NE #0 THEN LET CSR := CSR SET.BY #BIT3
  (4)  033542  005737  001504                                                          TST KPFLAG
  (6)  033546  001403                                                                  BEQ L313
```

```
  (6)  033550  052737  000010  001526                                                    BIS #BIT3,CSR
  (3)  033556                                                                      L313:;;;;;;
 5398  033556  005037  001530                  CLR     CSR+2
 5399  033562  004737  034302                  CALL CSROUT
 5400  033566  000002                          RTI
 5401                                          .SBTTL  TRAP    WRITE CHECKBITS THRU 1 SELECTED CSR TRAP HANDLER
 5402  033570  013737  001762  001526  $CB1CSR:MOV     CHECK,CSR                ;BITS 14-8
 5403  033576  052737  000006  001526          BIS     #BIT1!BIT2,CSR           ;CHECK MODE
 5404  033604                                  IF KPFLAG NE #0 THEN LET CSR := CSR SET.BY #BIT3
  (4)  033604  005737  001504                                                                TST KPFLAG
  (6)  033610  001403                                                                        BEQ L314
  (6)  033612  052737  000010  001526                                                        BIS #BIT3,CSR
  (3)  033620                                                                      L314:;;;;;;
 5405  033620  005037  001530                  CLR     CSR+2
 5406  033624  104425                          LOADCSR
 5407  033626  000002                          RTI
```

```
                        .SBTTL  TRAP    WAS THERE A SBE ON ANY CSR TRAP HANDLER
033630          $WASSBE:PUSH    R1,R4                                           MOV R1,-(SP)
033630  010146                                                                  MOV R4,-(SP)
033632  010446
033634  013701  001706          MOV     MKCSRS,R1       ;GET CSR'S BYTE
033640  005004                  CLR     R4
033642                          BEGIN LWSBE
033642                                                                  B42:;;;;;;;
033642                            FOR CSRNO := #0 TO #34 BY #4
033642  005037  001634                                                          CLR CSRNO
033646                                                                  B43:;;;;;;;
033646  106301                    ASLB        R1
033650                            ON.ERROR
033650  103010                                                                  BCC L315
033652  104426                      READCSR
033654                              IF #BIT4 SET.IN CSR
033654  032737  000020  001526                                                  BIT #BIT4,CSR
033662  001403                                                                  BEQ L316
033664                              SET     R4
033664  012704  177777      MOV     #-1,R4
033670                              LEAVE LWSBE
033670  000411                                                                  BR E42
033672                            END ;OF IF #BIT4
033672                                                                  L316:;;;;;;;
033672                          END ;OF ON.ERROR
033672                                                                  L315:;;;;;;;
033672                          IFB R1 EQ #0 THEN LEAVE LWSBE
033672  105701                                                                  TSTB R1
033674  001407                                                                  BEQ E42
033676                          END ;OF FOR CSRNO
033676  062737  000004  001634                                                  ADD #4,CSRNO
033704  023727  001634  000034                                                  CMP CSRNO,#34
033712  003755                                                                  BLE B43
033714                                                                  E43:;;;;;;;;
033714                  END LWSBE
033714                                                                  E42:;;;;;;;
033714  006004          ROR     R4              ;SET C BIT FOR ERROR
033716                  POP     R4,R1
033716  012604                                                                  MOV (SP)+,R4
033720  012601                                                                  MOV (SP)+,R1
033722                  ON.ERROR
033722  103004                                                                  BCC L317
033724  052766  000001  000002      BIS     #BIT0,2(SP)
033732                  ELSE
033732  000403                                                                  BR L320
033734                                                                  L317:;;;;;;;
033734  042766  000001  000002      BIC     #BIT0,2(SP)
033742                  END ;OF ON.ERROR
033742                                                                  L320:;;;;;;;
033742  000002          RTI
                        .SBTTL  TRAP    WAS THERE A SBE IN 1 SELECTED CSR TRAP HANDLER
                        ;ON RETURN IF CARRY IS SET THERE WAS A SBE
033744  104426  $WAS1SBE:READCSR
033746  042766  000001  000002      BIC     #BIT0,2(SP)     ;CLR C BIT ON STACK
033754  032737  000020  001526      BIT     #BIT4,CSR
033762  001403                      BEQ     1$
```

```
5441  033764  052766  000001  000002          BIS     #BIT0,2(SP)     ;SET C BIT ON STACK
5442  033772  000002                  1$:     RTI
5443                                          .SBTTL  TRAP    WAS THERE A DBE ON ANY CSR TRAP HANDLER
5444  033774                        $WASDBE:PUSH    R1,R4
  (2) 033774  010146                                                              MOV R1,-(SP)
  (3) 033776  010446                                                              MOV R4,-(SP)
5445  034000  013701  001706              MOV     MKCSRS,R1       ;GET CSR'S BYTE
5446  034004  005004                      CLR     R4
5447  034006                              BEGIN LWDBE
  (3) 034006                                                            B44:;;;;;;;
5448  034006                                FOR CSRNO := #0 TO #34 BY #4
  (3) 034006  005037  001634                                                      CLR CSRNO
  (3) 034012                                                            B45:;;;;;;;
5449  034012  106301                          ASLB        R1
5450  034014                                  ON.ERROR
  (8) 034014  103010                                                              BCC L321
5451  034016  104426                            READCSR
5452  034020                                   IF #BIT15 SET.IN CSR
  (4) 034020  032737  100000  001526                                             BIT #BIT15,CSR
  (6) 034026  001403                                                              BEQ L322
5453  034030                                     SET     R4
  (1) 034030  012704  177777              MOV     #-1,R4
5454  034034                                     LEAVE LWDBE
  (3) 034034  000411                                                            BR E44
5455  034036                                   END ;OF IF #BIT4
  (3) 034036                                                            L322:;;;;;;
5456  034036                                 END ;OF ON.ERROR
  (3) 034036                                                            L321:;;;;;;
5457  034036                                 IFB R1 EQ #0 THEN LEAVE LWDBE
  (6) 034036  105701                                                              TSTB R1
  (6) 034040  001407                                                              BEQ E44
5458  034042                                FOR CSRNO
  (6) 034042  062737  000004  001634                                             ADD #4,CSRNO
  (5) 034050  023727  001634  000034                                             CMP CSRNO,#34
  (5) 034056  003755                                                              BLE B45
  (5) 034060                                                            E45:;;;;;;;
5459  034060                              END LWDBE
  (3) 034060                                                            E44:;;;;;;
5460  034060  006004                      ROR     R4              ;SET C BIT FOR ERROR
5461  034062                              POP     R4,R1
  (2) 034062  012604                                                              MOV (SP)+,R4
  (3) 034064  012601                                                              MOV (SP)+,R1
5462  034066                              ON.ERROR
  (8) 034066  103004                                                              BCC L323
5463  034070  052766  000001  000002       BIS   #BIT0,2(SP)
5464  034076                              ELSE
  (3) 034076  000403                                                            BR L324
  (3) 034100                                                            L323:;;;;;;
5465  034100  042766  000001  000002       BIC   #BIT0,2(SP)
5466  034106                              END ;OF ON.ERROR
  (3) 034106                                                            L324:;;;;;;
5467  034106  000002                      RTI
5468                                      .SBTTL  TRAP    WAS THERE A DBE ON 1 SELECTED CSR TRAP HANDLER
5469                                      ;ON RETURN IF CARRY IS SET THERE WAS A DBE
5470  034110  104426                $WAS1DBE:       READCSR
5471  034112  005737  001526              TST     CSR             ;DBE?
```

```
 5472  034116  100405                        BMI     1$                      ;YES - SKIP
 5473  034120  005737  001530                TST     CSR+2                   ;DBE IN OTHER CSR WORD?
 5474  034124  100012                        BPL     3$                      ;NO - SKIP
 5475  034126  104021                        ERROR   +21                     ;YES - ONLY ONE FLAG POPPED UP!
 5476  034130  000404                        BR      2$
 5477  034132  005737  001530        1$:     TST     CSR+2                   ;DBE IN OTHER WORD?
 5478  034136  100401                        BMI     2$                      ;YES - SKIP
 5479  034140  104021                        ERROR   +21                     ;NO - ONLY ONE FLAG POPPED UP!
 5480  034142  052766  000001  000002  2$:   BIS     #BIT0,2(SP)             ;SET C BIT ON STACK
 5481  034150  000002                        RTI
 5482  034152  042766  000001  000002  3$:   BIC     #BIT0,2(SP)             ;CLR C BIT ON STACK
 5483  034160  000002                        RTI
 5484                                        .SBTTL  TRAP    CLEAR ALL MK11 CSR'S TRAP HANDLER
 5485  034162  005037  001526        $CLRCSR:CLR     CSR
 5486  034166  005037  001530                CLR     CSR+2
 5487  034172  004737  034302                CALL    CSROUT
 5488  034176  000002                        RTI
 5489                                        .SBTTL  TRAP    CLEAR 1 SELECTED MK11 CSR TRAP HANDLER
 5490  034200  005037  001526        $CLR1CSR:CLR    CSR
 5491  034204  005037  001530                CLR     CSR+2
 5492  034210  104425                        LOADCSR
 5493  034212  000002                        RTI
 5494                                        .SBTTL  TRAP    ECC DISABLE, CHECH MODE, & WRITE CHECKBITS IN ALL CSR'S TRAP HAN
 5495                                        ;CHECKBITS ALREADY IN LOC "CSR"
 5496  034214  052737  000006  001526  $CHKDIS:BIS     #BIT1!BIT2,CSR          ;ECC DISABLE & DIAG CHECK MODE
 5497  034222                                IF KPFLAG NE #0 THEN LET CSR := CSR SET.BY #BIT3
   (4) 034222  005737  001504                                                    TST KPFLAG
   (6) 034226  001403                                                            BEQ L325
   (6) 034230  052737  000010  001526                                            BIS #BIT3,CSR
   (3) 034236                                                                L325:;;;;;;
 5498  034236  005037  001530                CLR     CSR+2
 5499  034242  004737  034302                CALL    CSROUT
 5500  034246  000002                        RTI
 5501                                        .SBTTL  TRAP    ECC DISABLE, CHECH MODE, & WRITE CHECKBITS IN 1 SELECTED CSR TRA
 5502                                        ;CHECKBITS ALREADY IN LOC "CSR"
 5503  034250  052737  000006  001526  $CHK1DIS:BIS    #BIT1!BIT2,CSR          ;ECC DISABLE & DIAG CHECK MODE
 5504  034256                                IF KPFLAG NE #0 THEN LET CSR := CSR SET.BY #BIT3
   (4) 034256  005737  001504                                                    TST KPFLAG
   (6) 034262  001403                                                            BEQ L326
   (6) 034264  052737  000010  001526                                            BIS #BIT3,CSR
   (3) 034272                                                                L326:;;;;;;
 5505  034272  005037  001530                CLR     CSR+2
 5506  034276  104425                        LOADCSR
 5507  034300  000002                        RTI
```

```
 5509  034302                          CSROUT: SUBTST  <<SUBR  WRITE IN ALL CSR'S>>
  (3)                                  ;*****************************************************************************
  (3)                                  ;*SUBTEST       SUBR    WRITE IN ALL CSR'S
  (2)                                  ;*****************************************************************************
 5510  034302                                  PUSH    R1
  (2)  034302  010146                                                                    MOV R1,-(SP)
 5511  034304  013701  001706                  MOV     MKCSRS,R1       ;GET CSR'S BYTE
 5512  034310                                  BEGIN LCSROUT
  (3)  034310                                                                    B46:;;;;;;
 5513  034310                                    FOR CSRNO := #0 TO #34 BY #4
  (3)  034310  005037  001634                                                            CLR CSRNO
  (3)  034314                                                                    B47:;;;;;;
 5514  034314  106301                            ASLB        R1
 5515  034316                                    ON.ERROR
  (8)  034316  103001                                                                    BCC L327
 5516  034320  104425                              LOADCSR
 5517  034322                                    END ;OF ON.ERROR
  (3)  034322                                                                    L327:;;;;;;
 5518  034322                                    IFB R1 EQ #0 THEN LEAVE LCSROUT
  (6)  034322  105701                                                                    TSTB R1
  (6)  034324  001407                                                                    BEQ E46
 5519  034326                                    END ;OF FOR CSRNO
  (6)  034326  062737  000004  001634                                                    ADD #4,CSRNO
  (5)  034334  023727  001634  000034                                                    CMP CSRNO,#34
  (5)  034342  003764                                                                    BLE B47
  (5)  034344                                                                    E47:;;;;;;;
 5520  034344                                  END LCSROUT
  (3)  034344                                                                    E46:;;;;;;
 5521  034344                                  POP     R1
  (2)  034344  012601                                                                    MOV (SP)+,R1
 5522  034346  000207                          RETURN
```

```
                                CHKGEN: SUBTST<<SUBR    GENERATE CHECK BITS>>
                                ;*****************************************************************************
                                ;*SUBTEST        SUBR    GENERATE CHECK BITS
                                ;*****************************************************************************
                                        ;CHECK BIT GENERATOR ROUTINE
                                        ;CALLING SEQUENCE IS:
                                        ;       MOV     #WORD1,SOURCE   ;SOURCE = ADDRESS OF DATA
                                        ;       CALL    CHKGEN
                                        ;
                                        ;CHECK BITS RETURNED IN BITS 14-8 OF LOCATION CHECK
                                        ;
034350                                  PUSH    R0,R1,R2,R3,R4,R5
  (2) 034350  010046                                                            MOV R0,-(SP)
  (3) 034352  010146                                                            MOV R1,-(SP)
  (4) 034354  010246                                                            MOV R2,-(SP)
  (5) 034356  010346                                                            MOV R3,-(SP)
  (6) 034360  010446                                                            MOV R4,-(SP)
  (7) 034362  010546                                                            MOV R5,-(SP)
034364  012702  000077                        MOV     #77,R2          ;DEFAULT CHECKBITS FOR DOUBLE WORD OF ZEROS
034370  012703  034450                        MOV     #CHKTAB,R3      ;ADDRESS OF CHECKBIT TABLE
034374  013705  001760                        MOV     SOURCE,R5       ;GET SOURCE ADDRESS
034400  012501                                MOV     (R5)+,R1        ;GET LSB'S
034402  011500                                MOV     (R5),R0         ;GET MSB'S
                                1$:     SXT     R4              ;EXTEND SIGN OF DOUBLE WORD TO R4
034404  006704                                BICB    (R3)+,R4        ;ELIMINATE BITS THAT DON'T COUNT
034406  142304                                XOR     R4,R2           ;COMPLIMENT MASKED BITS IN CHECKBITS
034410  074402                                ASHC    #1,R0           ;DOUBLE PRECISION LEFT SHIFT R0,,R1
034412  073027  000001                        BNE     1$              ;LOOP TILL ALL BITS ARE CHECKED
034416  001372
                                        BIC     #^C177,R2                       ;KILL ALL JUNK BITS
034420  042702  177600                        SWAB    R2              ;POSITION CHECKBITS IN BITS 14-8
034424  000302
034426  010237  001762                        MOV     R2,CHECK
034432                                  POP     R5,R4,R3,R2,R1,R0
  (2) 034432  012605                                                            MOV (SP)+,R5
  (3) 034434  012604                                                            MOV (SP)+,R4
  (4) 034436  012603                                                            MOV (SP)+,R3
  (5) 034440  012602                                                            MOV (SP)+,R2
  (6) 034442  012601                                                            MOV (SP)+,R1
  (7) 034444  012600                                                            MOV (SP)+,R0
034446  000207                          RETURN

034450                          CHKTAB: .BYTE #3
034450      200                         .BYTE   ^C177           ;BIT 31
034451      301                         .BYTE   ^C076           ;BIT 30
034452      302                         .BYTE   ^C075           ;BIT 29
034453      203                         .BYTE   ^C174           ;BIT 28
034454      304                         .BYTE   ^C073           ;BIT 27
034455      205                         .BYTE   ^C172           ;BIT 26
034456      206                         .BYTE   ^C171           ;BIT 25
034457      307                         .BYTE   ^C070           ;BIT 24
                                        ;BYTE #2
034460      310                         .BYTE   ^C067           ;BIT 23
034461      211                         .BYTE   ^C166           ;BIT 22
034462      212                         .BYTE   ^C165           ;BIT 21
034463      313                         .BYTE   ^C064           ;BIT 20
```

```
5566  034464  214                    .BYTE   ↑C163          ;BIT 19
5567  034465  315                    .BYTE   ↑C062          ;BIT 18
5568  034466  316                    .BYTE   ↑C061          ;BIT 17
5569  034467  217                    .BYTE   ↑C160          ;BIT 16
5570                                 ;BYTE #1
5571  034470  320                    .BYTE   ↑C057          ;BIT 15
5572  034471  221                    .BYTE   ↑C156          ;BIT 14
5573  034472  222                    .BYTE   ↑C155          ;BIT 13
5574  034473  323                    .BYTE   ↑C054          ;BIT 12
5575  034474  224                    .BYTE   ↑C153          ;BIT 11
5576  034475  325                    .BYTE   ↑C052          ;BIT 10
5577  034476  326                    .BYTE   ↑C051          ;BIT  9
5578  034477  227                    .BYTE   ↑C150          ;BIT  8
5579                                 ;BYTE #0
5580  034500  340                    .BYTE   ↑C037          ;BIT  7
5581  034501  241                    .BYTE   ↑C136          ;BIT  6
5582  034502  242                    .BYTE   ↑C135          ;BIT  5
5583  034503  343                    .BYTE   ↑C034          ;BIT  4
5584  034504  244                    .BYTE   ↑C133          ;BIT  3
5585  034505  345                    .BYTE   ↑C032          ;BIT  2
5586  034506  346                    .BYTE   ↑C031          ;BIT  1
5587  034507  247                    .BYTE   ↑C130          ;BIT  0
```

```
 5590  034510                                  SUBTST<<SUBR    MAPPER>>
  (3)                                  ;*****************************************************************************
  (3)                                  ;*SUBTEST        SUBR    MAPPER
  (2)                                  ;*****************************************************************************
 5591                                  ;THIS SUBROUTINE MAPS THE MEMORY BANK (16K WORDS = 1 BANK)
 5592                                  ;IN R3 TO THE TEST PATTERN AREA (SUPERVISOR VIRTUAL (60000 - 157777).
 5593                                  ;
 5594                                  ;CALL   MOV     BANKNO,R3                ;SET UP BANK ARGUEMENT
 5595                                  ;       CALL    MAPPER                   ;ACTUAL CALL
 5596                                  ;       RETURN                           ;ONLY RETURN
 5597
 5598                                          ;SET SUPERVISOR UP FOR 1 TO 1 MAP
 5599  034510                          MAPPER: PUSH    R0,R1,R2,R4,R5
  (2)  034510  010046                                                                                  MOV R0,-(SP)
  (3)  034512  010146                                                                                  MOV R1,-(SP)
  (4)  034514  010246                                                                                  MOV R2,-(SP)
  (5)  034516  010446                                                                                  MOV R4,-(SP)
  (6)  034520  010546                                                                                  MOV R5,-(SP)
 5600  034522  012700  172340                  MOV     #KIPAR0,R0               ;FIRST AREA TO MAP TO
 5601  034526  012701  172240                  MOV     #SIPAR0,R1               ;FIRST ADDRESS REGISTER
 5602  034532  012702  077406                  MOV     #77406,R2                ;CONSTANT FOR 4K PAGE, UP, R/W
 5603  034536  012704  172200                  MOV     #SIPDR0,R4               ;FIRST DISCRIPTOR REGISTER
 5604  034542  012705  000010                  MOV     #8.,R5                   ;COUNTER
 5605  034546  012021                  1$:     MOV     (R0)+,(R1)+              ;PUT IN SUPERVISOR ADDRESS
 5606  034550  010224                          MOV     R2,(R4)+                 ;PUT IN SUPERVISOR DISCRIPTOR
 5607  034552  077503                          SOB     R5,1$                    ;LOOP TILL DONE
 5608  034554  012741  177600                  MOV     #177600,-(R1)            ;CORRECT LAST FIELD FOR PERIPHERALS PAGE
 5609
 5610                                          ;SET UP SUPERVISOR FOR TEST AREA
 5611  034560  022703  000170                  CMP     #120.,R3                 ;MAP NOTHING (1 TO 1)?
 5612  034564  001412                          BEQ     3$                       ;YES - SKIP
 5613  034566  072327  000011                  ASH     #9.,R3                   ;BANK 1 STARTS AT 100,000 LESS 6 LSB'S
 5614                                                                           ;FOR MEMORY MANAGEMENT = 1000
 5615  034572  012701  172246                  MOV     #SIPAR3,R1               ;SETUP FOR AUTO INCREMENTING
 5616  034576  012702  000004                  MOV     #4,R2                    ;COUNTER
 5617  034602  010321                  2$:     MOV     R3,(R1)+                 ;PLUG IN PAR INFO
 5618  034604  062703  000200                  ADD     #200,R3                  ;BUMP ADDRESS 4K
 5619  034610  077204                          SOB     R2,2$                    ;LOOP TILL DONE
 5620  034612                          3$:     POP     R5,R4,R2,R1,R0
  (2)  034612  012605                                                                                  MOV (SP)+,R5
  (3)  034614  012604                                                                                  MOV (SP)+,R4
  (4)  034616  012602                                                                                  MOV (SP)+,R2
  (5)  034620  012601                                                                                  MOV (SP)+,R1
  (6)  034622  012600                                                                                  MOV (SP)+,R0
 5621  034624  000207                          RETURN
 5622                                          .SBTTL  TRAP    MAP KERNEL (ALMOST 1 TO 1) TRAP HANDLER
 5623  034626                          $KMAP:  PUSH    R0,R1,R2,R3,R4
  (2)  034626  010046                                                                                  MOV R0,-(SP)
  (3)  034630  010146                                                                                  MOV R1,-(SP)
  (4)  034632  010246                                                                                  MOV R2,-(SP)
  (5)  034634  010346                                                                                  MOV R3,-(SP)
  (6)  034636  010446                                                                                  MOV R4,-(SP)
 5624  034640  005000                          CLR     R0                       ;1ST AREA TO MAP TO
 5625  034642  012701  172340                  MOV     #KIPAR0,R1               ;FIRST ADDRESS
 5626  034646  012702  077406                  MOV     #77406,R2                ;CONSTANT FOR 4K PAGE,UP,R/W
 5627  034652  012703  172300                  MOV     #KIPDR0,R3               ;1ST PAGE DISCRIPTOR REGISTER
```

```
 5628  034656  012704  000010          MOV     #8.,R4                  ;COUNTER
 5629  034662  010021                  1$:     MOV     R0,(R1)+        ;PUT IN KERNEL ADDRESS
 5630  034664  010223                          MOV     R2,(R3)+        ;PUT IN KERNEL DISCRIPTOR
 5631  034666  062700  000200                  ADD     #200,R0         ;ADD ADDRESS CONSTANT FOR 4K CHANGE
 5632  034672  077405                          SOB     R4,1$           ;LOOP TILL DONE
 5633  034674  012741  177600                  MOV     #177600,-(R1)   ;THE PERIPHERALS PAGE TO KIPAR7
 5634  034700  012741  177400                  MOV     #177400,-(R1)   ;AND NEXT LOWER PAGE TO KIPAR6
 5635  034704                                  POP     R4,R3,R2,R1,R0
                                                                                MOV (SP)+,R4
   (2) 034704  012604                                                           MOV (SP)+,R3
   (3) 034706  012603                                                           MOV (SP)+,R2
   (4) 034710  012602                                                           MOV (SP)+,R1
   (5) 034712  012601                                                           MOV (SP)+,R0
   (6) 034714  012600
 5636  034716  000002                          RTI
```

```
                        RELOCATE:SUBTST <<RELOCATE PROGRAM>>
                        ;*****************************************************************************
                        ;*SUBTEST       RELOCATE PROGRAM
                        ;*****************************************************************************
                              IF #SW12 SET.IN @SWR THEN $RETURN ERROR
035720  032777 010000 145252                                          BIT #SW12,@SWR
035726  001402                                                        BEQ L330
035730  000261                                                        SEC
035732  000207                                                        RTS PC
035734                                                          L330:;;;;;;
                              BEGIN LOADERBANK
035734                                                          B50:;;;;;;
                                FOR BANK := #1 TO #167
035734  012737  000001  001470                                        MOV #1,BANK
035742                                                          B51:;;;;;;
                                  CALL EXBANK
                                  IF ACFLAG NE #0 AND RRFLAG EQ #0 AND BMFLAG EQ #0
035746  005737  001506                                                TST ACFLAG
035752  001424                                                        BEQ L331
035754  005737  001514                                                TST RRFLAG
035760  001021                                                        BNE L331
035762  005737  001520                                                TST BMFLAG
035766  001016                                                        BNE L331
035770  013700  001470                    MOV     BANK,R0
035774  010037  002116                    MOV     R0,LOADBANK
035000  013701  002114                    MOV     LOADHOME,R1
035004  004737  035570                    CALL    BANKMOV
035010  013701  001472                    MOV     BANKINDEX,R1
035014  052761  100000  002232            BIS     #BIT15,CONFIG+2(R1)     ;MARK LOADER
                                          LEAVE LOADERBANK
035022  000410                                                        BR E50
                                  END ;OF IF ACFLAG
                                END ;OF FOR BANK
                                                                L331:;;;;;;
035024  005237  001470                                                INC BANK
035030  023727  001470  000167                                        CMP BANK,#167
035036  003741                                                        BLE B51
                                                                E51:;;;;;;
                                TYPE  MSG096                    ;LOADERS BEING DESTROYED
                              TYPEIT  ,MSG096
                              END LOADERBANK
                                                                E50:;;;;;;
                              BEGIN FINDBANK
                                FOR R1 := #3*2 TO #167*3*2 BY #6
035044  012701  000006                                                MOV #3*2,R1
                                                                B53:;;;;;;
                                  IF #BIT11!BIT15 OFF.IN CONFIG(R1) ;IF NO ERRORS & NOT PROGRAM SPACE
035050  032761  104000  002230                                        BIT #BIT11!BIT15,CONFIG(
035056  001035                                                        BNE L332
                                    IF #BIT15 OFF.IN CONFIG+2(R1) ;IF NOT LOADER BANK
035060  032761  100000  002232                                        BIT #BIT15,CONFIG+2(R1)
035066  001031                                                        BNE L333
                                      IF CPUBIT SET.IN CONFIG(R1) ;IF ACCESSABLE
035070  033761  001476  002230                                        BIT CPUBIT,CONFIG(R1)
035076  001425                                                        BEQ L334
```

```
5661   035100
 (5)   035100  032761  060000  002230          IF #BIT13!BIT14 OFF.IN CONFIG(R1) THEN LEAVE FINDBANK ;IF MJ11
 (5)   035106  001436                                                                  BIT #BIT13!BIT14,CONFIG(
                                                                                       BEQ E52
5662   035110
 (4)   035110  032761  020000  002230          IF #BIT13 SET.IN CONFIG(R1) AND #BIT14 OFF.IN CONFIG(R1) ;IF MK11
 (6)   035116  001415                                                                  BIT #BIT13,CONFIG(R1)
                                                                                       BEQ L335
 (6)   035120  032761  040000  002230                                                 BIT #BIT14,CONFIG(R1)
 (8)   035126  001011                                                                  BNE L335
5663   035130
 (4)   035130  032761  010000  002222            IF #BIT12 SET.IN CONFIG-6(R1) AND #BIT12 OFF.IN CONFIG(R1)
 (6)   035136  001405                                                                  BIT #BIT12,CONFIG-6(R1)
                                                                                       BEQ L336
 (6)   035140  032761  010000  002230                                                 BIT #BIT12,CONFIG(R1)
 (8)   035146  001001                                                                  BNE L336
5664
5665   035150                                      ;IF 1ST UNPROTECTED MK11 BANK
 (3)   035150  000415                              LEAVE FINDBANK
                                                                                       BR E52
5666   035152                                    END ;OF IF #BIT12
 (3)   035152                                                                  L336:;;;;;;
5667   035152                                  END ;OF IF #BIT13
 (3)   035152                                                                  L335:;;;;;;
5668   035152                                END ;OF IF CPUBIT
 (3)   035152                                                                  L334:;;;;;;
5669   035152                              END ;OF IF #BIT15
 (3)   035152                                                                  L333:;;;;;;
5670   035152                            END ;OF IF #BIT11
 (3)   035152                                                                  L332:;;;;;;
5671   035152                          END ;OF FOR
 (6)   035152  062701  000006                                                          ADD #6,R1
 (5)   035156  020127  001312                                                          CMP R1,#167*3*2
 (5)   035162  003732                                                                  BLE B53
 (5)   035164                                                                  E53:;;;;;;;
5672   035164          IF #SW13 OFF.IN @SWR
 (4)   035164  032777  020000  145006                                                  BIT #SW13,@SWR
 (6)   035172  001002                                                                  BNE L337
5673   035174             TYPE        MSG075          ;RELOCATION NOT POSSIBLE
 (1)   035174  104401  071651       TYPEIT  ,MSG075
5674   035200           END ;OF IF #SW13
 (3)   035200                                                                  L337:;;;;;;
5675   035200             $RETURN ERROR
 (2)   035200  000261                                                                  SEC
 (2)   035202  000207                                                                  RTS PC
5676   035204         END FINDBANK
 (3)   035204                                                                  E52:;;;;;;
5677   035204  005000         CLR     R0
5678   035206  071027  000006  DIV     #6,R0
5679   035212       RELOC1: LET NEWBANK := R0
 (3)   035212  010037  001756                                                          MOV R0,NEWBANK
5680   035216  004737  035736         CALL    USERMAP         ;MAP NEWBANK TO USER PAR
5681   035222                        USER                    ;ENTER USER MODE
 (1)   035222  052737  140000  177776 BIS     #BIT15!BIT14,PSW       ;DO IT TO IT!
5682   035230                        BMOV    0,100000,SIZE   ;MOVE PROGRAM
 (1)   035230  004537  037342          JSR R5,BLOCK3
 (1)   035234  040000                  SIZE
 (1)   035236  100000                  100000
 (1)   035240  000000                  0
5683   035242  104417                KERNEL                  ;ENTER KERNEL MODE
```

```
 5684  035244  042737  000040  172516          BIC     #BIT5,MMR3              ;TURN OFF UNIBUS MAP
 5685  035252  013700  001756                  MOV     NEWBANK,R0
 5686  035256  006200                          ASR     R0
 5687  035260                                  ON.ERROR
   (8) 035260  103003                                                          BCC L340
 5688  035262  012737  100000  170200            MOV   #BIT15,MAPLO
 5689  035270                                  END ;OF ON.ERROR
   (3) 035270                                                                L340:;;;;;;
 5690  035270  010037  170202                  MOV     R0,MAPHO
 5691  035274  004737  035524                  CALL    LOWMAP                  ;SETUP LOWER 16K IN UNIBUS MAP
 5692  035300  052737  000040  172516          BIS     #BIT5,MMR3              ;ENERGIZE UNIBUS MAP
 5693  035306  042737  001001  177572          BIC     #BIT9!BIT0,MMR0 ;DEENERGIZE MEMORY MANAGEMENT - BUT DON'T TRAP
 5694  035314  004737  036020                  CALL    NEWKERNEL
 5695  035320  052737  001001  177572          BIS     #BIT9!BIT0,MMR0 ;ENERGIZE MEMORY MANAGEMENT - BUT DON'T TRAP
 5696  035326                                  SET     RLFLAG
   (1) 035326  012737  177777  001516          MOV     #-1,RLFLAG
 5697  035334                                  $RETURN NOERROR
   (2) 035334  000241                                                          CLC
   (2) 035336  000207                                                          RTS PC
 5698
 5699  035340                          UNRELOCATE:SUBTST       <<UNRELOCATE PROGRAM>>
   (3)                               ;*****************************************************************************
   (3)                               ;*SUBTEST       UNRELOCATE PROGRAM
   (2)                               ;*****************************************************************************
 5700                                      ;RESTORE LOADERS
 5701  035340                              PUSH    R0
   (2) 035340  010046                                                          MOV R0,-(SP)
 5702  035342  013701  002116              MOV     LOADBANK,R1
 5703  035346  013700  002114              MOV     LOADHOME,R0
 5704  035352  004737  035570              CALL    BANKMOV
 5705  035356                              PUSH    BANK
   (2) 035356  013746  001470                                                  MOV BANK,-(SP)
 5706  035362  013737  002116  001470      MOV     LOADBANK,BANK
 5707  035370  004737  036052              CALL    EXBANK
 5708  035374  013701  001472              MOV     BANKINDEX,R1
 5709  035400  042761  100000  002232      BIC     #BIT15,CONFIG+2(R1)     ;CLEAR LOADER FLAG
 5710  035406                              POP     BANK
   (2) 035406  012637  001470                                                  MOV (SP)+,BANK
 5711
 5712  035412                          LET NEWBANK := #0
   (3) 035412  005037  001756                                                  CLR NEWBANK
 5713  035416  004737  035736              CALL    USERMAP                 ;MAP NEWBANK TO USER PAR
 5714  035422                              USER                            ;ENTER USER MODE
   (1) 035422  052737  140000  177776      BIS     #BIT15!BIT14,PSW                ;DO IT TO IT!
 5715  035430                              BMOV    0,100000,SIZE           ;MOVE PROGRAM
   (1) 035430  004537  037342                  JSR R5,BLOCK3
   (1) 035434  040000                          SIZE
   (1) 035436  100000                          100000
   (1) 035440  000000                          0
 5716  035442  104417                      KERNEL                          ;ENTER KERNEL MODE
 5717  035444  042737  001001  177572      BIC     #BIT9!BIT0,MMR0 ;DEENERGIZE MEMORY MANAGEMENT - BUT DON'T TRAP
 5718  035452  004737  036020              CALL    NEWKERNEL
 5719  035456  052737  001001  177572      BIS     #BIT9!BIT0,MMR0 ;ENERGIZE MEMORY MANAGEMENT - BUT DON'T TRAP
 5720  035464  005037  001516              CLR     RLFLAG
 5721  035470  042737  000040  172516      BIC     #BIT5,MMR3              ;TURN OFF UNIBUS MAP
 5722  035476  005037  170200              CLR     MAPLO
```

```
 5723  035502  005037  170202                  CLR     MAPHO
 5724  035506  004737  035524                  CALL    LOWMAP                  ;SETUP LOWER 16K OF UNIBUS MAP
 5725  035512  052737  000040  172516          BIS     #BIT5,MMR3              ;ENERGIZE UNIBUS MAP
 5726  035520                                  POP     R0
   (2) 035520  012600                                                          MOV (SP)+,R0
 5727  035522  000207                          RETURN
 5728
 5729  035524                          LOWMAP: SUBTST  <<SETUP LOWER 16K OF UNIBUS MAP>>
   (3)                                   ;*****************************************************************************
   (3)                                   ;*SUBTEST       SETUP LOWER 16K OF UNIBUS MAP
   (2)                                   ;*****************************************************************************
 5730  035524                                  PUSH    R0,R1,R2
   (2) 035524  010046                                                          MOV R0,-(SP)
   (3) 035526  010146                                                          MOV R1,-(SP)
   (4) 035530  010246                                                          MOV R2,-(SP)
 5731  035532  012700  170200                  MOV     #MAPL0,R0
 5732  035536  012701  170204                  MOV     #MAPL1,R1
 5733  035542  012702  000003                  MOV     #3,R2
 5734  035546  012011                  1$:     MOV     (R0)+,(R1)
 5735  035550  062721  020000                  ADD     #BIT13,(R1)+
 5736  035554  012021                          MOV     (R0)+,(R1)+
 5737  035556  077205                          SOB     R2,1$
 5738  035560                                  POP     R2,R1,R0
   (2) 035560  012602                                                          MOV (SP)+,R2
   (3) 035562  012601                                                          MOV (SP)+,R1
   (4) 035564  012600                                                          MOV (SP)+,R0
 5739  035566  000207                          RETURN
```

```
035570                                  BANKMOV:SUBTST  <<MOVE BANKS>>
                                        ;*******************************************************************
                                        ;*SUBTEST        MOVE BANKS
                                        ;*******************************************************************
                                                ;MOVE 3/4 OF A BANK
                                                ;CALLING SEQUENCE
                                                ;R0 = DESTINATION BANK
                                                ;R1 = SOURCE BANK
035570  104415                                  SAVREG
035572  004737  035736                          CALL    USERMAP
035576  104416                                  RESREG
035600  104415                                  SAVREG
035602  072027  000011                          ASH     #9.,R0
035606  072127  000011                          ASH     #9.,R1
035612  012702  177650                          MOV     #UIPAR4,R2
035616  012703  000200                          MOV     #200,R3

035622  010122                                  MOV     R1,(R2)+                ;MAP 1ST HALF BANK
035624  060301                                  ADD     R3,R1                   ;BUMP BY 4K
035626  010122                                  MOV     R1,(R2)+
035630  060301                                  ADD     R3,R1

035632  010022                                  MOV     R0,(R2)+
035634  060300                                  ADD     R3,R0
035636  010022                                  MOV     R0,(R2)+
035640  060300                                  ADD     R3,R0

035642                                          USER
035642  052737  140000  177776                  BIS     #BIT15!BIT14,PSW                ;DO IT TO IT!
035650                                          BMOV    100000,140000,SIZE/2    ;MOV 1ST HALF BANK
035650  004537  037342                              JSR R5,BLOCK3
035654  020000                                      SIZE/2
035656  140000                                      140000
035660  100000                                      100000
035662  104417                                  KERNEL

035664  012702  177650                          MOV     #UIPAR4,R2

035670  010122                                  MOV     R1,(R2)+                ;MAP 2ND HALF BANK
035672  060301                                  ADD     R3,R1                   ;BUMP BY 4K
035674  010122                                  MOV     R1,(R2)+
035676  060301                                  ADD     R3,R1

035700  010022                                  MOV     R0,(R2)+
035702  060300                                  ADD     R3,R0
035704  010022                                  MOV     R0,(R2)+
035706  060300                                  ADD     R3,R0

035710                                          USER
035710  052737  140000  177776                  BIS     #BIT15!BIT14,PSW                ;DO IT TO IT!
035716                                          BMOV    100000,140000,SIZE/4    ;MOV 3ND FOURTH OF BANK
035716  004537  037342                              JSR R5,BLOCK3
035722  010000                                      SIZE/4
035724  140000                                      140000
035726  100000                                      100000
035730  104417                                  KERNEL
```

```
035732  104416                          RESREG
035734  000207                          RETURN
035736                          USERMAP:SUBTST  <<SUBR  MAP USER TO NEW BANK>>
                                ;*******************************************************************
                                ;*SUBTEST        SUBR    MAP USER TO NEW BANK
                                ;*******************************************************************
035736  012701  177640                  MOV     #UIPAR0,R1              ;COPY KERNEL PAR'S & PDR'S (0-3)
035742  012702  172340                  MOV     #KIPAR0,R2
035746  012703  177600                  MOV     #UIPDR0,R3
035752  012704  172300                  MOV     #KIPDR0,R4
035756  012705  000004                  MOV     #4,R5
035762  012221                  1$:     MOV     (R2)+,(R1)+
035764  011423                          MOV     (R4),(R3)+
035766  077503                          SOB     R5,1$

035770  013700  001756                  MOV     NEWBANK,R0
035774  072027  000011                  ASH     #9.,R0                  ;BANK 1 STARTS AT 100,000 LESS 6 LSB'S
                                                                        ;FOR MEMORY MANAGEMENT = 1000
036000  012705  000004                  MOV     #4,R5
036004  010021                  2$:     MOV     R0,(R1)+                ;SETUP UIPAR(4-7)
036006  062700  000200                  ADD     #200,R0                 ;BUMP ADDRESS 4K
036012  011423                          MOV     (R4),(R3)+              ;SETUP UIPDR(4-7)
036014  077505                          SOB     R5,2$
036016  000207                          RETURN

036020                          NEWKERNEL:SUBTST        <<SUBR  SETUP KERNEL PAR'S FOR NEW BANK>>
                                ;*******************************************************************
                                ;*SUBTEST        SUBR    SETUP KERNEL PAR'S FOR NEW BANK
                                ;*******************************************************************
036020  012700  172340                  MOV     #KIPAR0,R0
036024  013701  001756                  MOV     NEWBANK,R1
036030  072127  000011                  ASH     #9.,R1                  ;BANK 1 STARTS AT 100,000 LESS 6 LSB'S
                                                                        ;FOR MEMORY MANAGEMENT = 1000
036034  012705  000006                  MOV     #6,R5
036040  010120                  1$:     MOV     R1,(R0)+                ;SETUP KIPAR(0-5)
036042  062701  000200                  ADD     #200,R1
036046  077504                          SOB     R5,1$
036050  000207                          RETURN
```

```
 5819  036052                          EXBANK: SUBTST  <<SUBR  EXAMINE BANK>>
  (3)                                ;*****************************************************************************
  (3)                                ;*SUBTEST        SUBR    EXAMINE BANK
  (2)                                ;*****************************************************************************
 5820                                ;DOES THE FOLLOWING:
 5821                                ;(1) SETS UP "BANKINDEX" AND R0 BASED ON VALUE OF "BANK"
 5822                                ;(2) SETS THE "MKFLAG" IF THE BANK IS AN MK11
 5823                                ;(3) SETS THE "KPFLAG" IF THE BANK IS THE PROTECTED REGION OF MK11 MEMORY
 5824                                ;(4) SETS THE "ACFLAG" IF THE BANK CAN BE ACCESSED BY THIS CPU
 5825                                ;(5) SETS THE "PFLAG" IF THE BANK IS IN PROGRAM SPACE
 5826                                ;(6) SETS THE "RRFLAG" IF RELOCATION IS REQUIRED TO TEST THIS BANK
 5827                                ;     HOWEVER, IT COMPLIMENTS THIS FLAG IF THE RELOCATION FLAG "RLFLAG" IS SET
 5828                                ;     (THIS IS NECESSARY FOR THE USE OF THE RECURSIVE "MODE" SUBROUTINES)
 5829                                ;(7) SETS THE "BMFLAG" IF THE BANK IS A BAD MEMORY
 5830                                ;     HOWEVER, IT COMPLIMENTS THIS FLAG IF THE "WORST" FLAG IS NOT SET
 5831                                ;     (THIS IS NECESSARY FOR THE USE OF THE RECURSIVE "MODE" SUBROUTINES)
 5832  036052                              PUSH    R0,R1,R2
  (2)  036052  010046                                                                  MOV R0,-(SP)
  (3)  036054  010146                                                                  MOV R1,-(SP)
  (4)  036056  010246                                                                  MOV R2,-(SP)
 5833  036060  005037  001510                    CLR     MKFLAG
 5834  036064  005037  001504                    CLR     KPFLAG
 5835  036070                                    SET     ACFLAG
  (1)  036070  012737  177777  001506           MOV     #-1,ACFLAG
 5836  036076  005037  001512                    CLR     PFLAG
 5837  036102  005037  001514                    CLR     RRFLAG
 5838  036106  005037  001520                    CLR     BMFLAG
 5839  036112  013701  001470                    MOV     BANK,R1
 5840  036116  070127  000006                    MUL     #6,R1                       ;R0 = BANK * 6
 5841  036122  010137  001472                    MOV     R1,BANKINDEX
 5842  036126  032761  010000  002230           BIT     #BIT12,CONFIG(R1)           ;PROTECTED REGION OF MK11 MEMORY?
 5843  036134  001406                          BEQ     1$                          ;NO - SKIP
 5844  036136                                    SET     KPFLAG
  (1)  036136  012737  177777  001504           MOV     #-1,KPFLAG
 5845  036144                                    SET     RRFLAG
  (1)  036144  012737  177777  001514           MOV     #-1,RRFLAG
 5855
 5856  036212  016100  002230          2$:     MOV     CONFIG(R1),R0
 5857  036216  042700  117777                  BIC     #^C60000,R0                 ;GET MEMORY TYPE BITS
 5858  036222  022700  020000                  CMP     #20000,R0                   ;IS IT AN MK11?
 5859  036226  001003                          BNE     3$                          ;NO - SKIP
 5860  036230                                    SET     MKFLAG
  (1)  036230  012737  177777  001510           MOV     #-1,MKFLAG
 5861
 5862  036236  032761  100000  002230  3$:     BIT     #BIT15,CONFIG(R1)           ;BANK = PROGRAM SPACE?
 5863  036244  001406                          BEQ     5$                          ;NO - SKIP
 5864  036246                                    SET     PFLAG
  (1)  036246  012737  177777  001512           MOV     #-1,PFLAG
 5865  036254                                    SET     RRFLAG
  (1)  036254  012737  177777  001514           MOV     #-1,RRFLAG
 5866
 5867  036262  005737  001516          5$:     TST     RLFLAG                      ;IS PROGRAM RELOCATED?
 5868  036266  001402                          BEQ     6$                          ;NO - SKIP
 5869  036270  005137  001514                  COM     RRFLAG                      ;YES - COMPLIMENT RELOCATION REQUIRED FLAG
 5870
 5871  036274  032761  004000  002230  6$:     BIT     #BIT11,CONFIG(R1)           ;ERRORS PRESENT IN THIS BANK?
```

```
 5872  036302  001403                    BEQ     8$                ;NO - SKIP
 5873  036304                            SET     BMFLAG
   (1) 036304  012737  177777  001520    MOV     #-1,BMFLAG
 5874
 5875  036312  005737  002140    8$:     TST     WORST             ;IS THIS A WORST FIRST PASS?
 5876  036316  001002                    BNE     9$                ;YES - SKIP
 5877  036320  005137  001520            COM     BMFLAG            ;NO - COMPLIMENT BAD MEMORY FLAG
 5878  036324                    9$:     POP     R2,R1,R0
   (2) 036324  012602                                                    MOV (SP)+,R2
   (3) 036326  012601                                                    MOV (SP)+,R1
   (4) 036330  012600                                                    MOV (SP)+,R0
 5879  036332  000207                    RETURN
```

```
                                BANKOK: SUBTST  <<SUBR  BANK OK?>>
                                ;*****************************************************************************
                                ;*SUBTEST        SUBR    BANK OK?
                                ;*****************************************************************************
                                        ;TEST TO INSURE THAT THE TYPE OF MEMORY IN THE PRESENT BANK
                                        ;IS OF THE TYPE WE ARE TESTING "TMFLAG".
                                        ;RESULT IS RETURNED IN THE CONDITION CODES (OK = (=0)).
036334  013700  001522                  MOV     TMFLAG,R0
036340  005100                          COM     R0
036342  013701  001510                  MOV     MKFLAG,R1
036346  074001                          XOR     R0,R1
036350  000207                          RETURN                          ;OK = (=OK)

036352                          INCRPT:
036352                          INCPAT: SUBTST  <<SUBR  INCREMENT PATTERN TESTING MK11'S>>
                                ;*****************************************************************************
                                ;*SUBTEST        SUBR    INCREMENT PATTERN TESTING MK11'S
                                ;*****************************************************************************
                                        ;INCREMEMT THE PATTERN & SET UP THE CONDITION CODES
                                        ;RESULT - Z BIT SET INDICATES OVERFLOW
036352  005237  001502                  INC     PATTERN
036356  022737  000040  001502          CMP     #40,PATTERN             ;SET UP CONDITION CODES
036364  000207                          RETURN                          ;NOT EQUAL TO ZERO IS GOOD (NO OVERFLOW)

036366                          SETPAT:
036366                          HIPAT:  SUBTST  <<SUBR  SET HIGHEST PATTERN TESTING TYPE>>
                                ;*****************************************************************************
                                ;*SUBTEST        SUBR    SET HIGHEST PATTERN TESTING TYPE
                                ;*****************************************************************************
036366  012737  000037  001502          MOV     #37,PATTERN             ;SET HIGHEST PATTERN
036374  000207                          RETURN

036376                          INCBNK: SUBTST  <<SUBR  INCREMENT BANK & TEST>>
                                ;*****************************************************************************
                                ;*SUBTEST        SUBR    INCREMENT BANK & TEST
                                ;*****************************************************************************
                                        ;RESULTS RETURNED IN CONDITION CODES
036376  005237  001470                  INC     BANK
036402  022737  000170  001470          CMP     #120.,BANK              ;TOO FAR?
036410  000207                          RETURN

036412                          INCMAR: SUBTST  <<SUBR  INCREMENT MARGINS & TEST>>
                                ;*****************************************************************************
                                ;*SUBTEST        SUBR    INCREMENT MARGINS & TEST
                                ;*****************************************************************************
                                        ;SKIPS THE SPECIAL CASE (MARGIN CANNOT = 1)
                                        ;RETURNS THE RESULT OF MARGIN OVERFLOW IN THE CONDITION CODES
                                        .ENABL  LSB
036412                                  IF #SW11 OFF.IN @SWR AND @VFLAG EQ #0
036412  032777  004000  143560                                          BIT #SW11,@SWR
036420  001004                                                          BNE L342
036422  005737  002016                                                  TST @VFLAG
036426  001001                                                          BNE L342
036430                                    GOTO 1$
036430  000410                                                          BR 1$
036432                                  END :OF IF #SW11
```

```
 (3)  036432                                                                                  L342:;;;;;;
5918  036432  013737  001474  001754          MOV     MARGIN,OLDMARGIN
5919  036440  005037  001474                  CLR     MARGIN                  ;YES - SET MARGIN TO ZERO
5920  036444  004737  036704                  CALL    COREHISTORY
5921  036450  000435                          BR      3$
5922  036452  013737  001474  001754  1$:     MOV     MARGIN,OLDMARGIN
5923  036460  005237  001474                  INC     MARGIN
5924  036464  022737  000001  001474          CMP     #1,MARGIN               ;SPECIAL CASE?
5925  036472  001002                          BNE     2$                      ;NO - SKIP
5926  036474  005237  001474                  INC     MARGIN                  ;YES - SET MARGIN TO "2"
5927  036500                          2$:     IF MKFLAG EQ #0 AND MARGIN NE #6
 (4)  036500  005737  001510                                                                  TST MKFLAG
 (6)  036504  001005                                                                          BNE L343
 (6)  036506  023727  001474  000006                                                          CMP MARGIN,#6
 (8)  036514  001401                                                                          BEQ L343
5928  036516                                    GOTO 3$
 (2)  036516  000412                                                                          BR 3$
5929  036520                                  END ;OF IF MKFLAG
 (3)  036520                                                                                  L343:;;;;;;
5930  036520                                  IF MKFLAG NE #0 AND MARGIN NE #4
 (4)  036520  005737  001510                                                                  TST MKFLAG
 (6)  036524  001405                                                                          BEQ L344
 (6)  036526  023727  001474  000004                                                          CMP MARGIN,#4
 (8)  036534  001401                                                                          BEQ L344
5931  036536                                    GOTO 3$
 (2)  036536  000402                                                                          BR 3$
5932  036540                                  END ;OF IF MKFLAG
 (3)  036540                                                                                  L344:;;;;;;
5933  036540  005037  001474                  CLR     MARGIN                  ;YES - SET MARGIN TO ZERO
5934  036544  004737  036704          3$:     CALL    COREHISTORY
5935  036550  005737  001474                  TST     MARGIN                  ;SET CONDITION CODE - DONE IF Z BIT SET
5936  036554  000207                          RETURN
5937                                          .DSABL  LSB
5938
5939  036556                          DECMAR: SUBTST  <<DECREMENT MARGINS & TEST>>
 (3)                                  ;*******************************************************************************
 (3)                                  ;*SUBTEST        DECREMENT MARGINS & TEST
 (2)                                  ;*******************************************************************************
5940                                          ;SKIPS THE SPECIAL CASE (MARGIN CANNOT = 1)
5941                                          ;RETURNS THE RESULT OF MARGIN UNDERFLOW IN THE CONDITION CODES
5942  036556  013737  001474  001754          MOV     MARGIN,OLDMARGIN
5943  036564  005337  001474                  DEC     MARGIN
5944  036570  022737  000001  001474          CMP     #1,MARGIN               ;SPECIAL CASE?
5945  036576  001002                          BNE     1$                      ;NO - SKIP
5946  036600  005337  001474                  DEC     MARGIN                  ;YES - SET MARGIN TO "0"
5947  036604  004737  036704          1$:     CALL    COREHISTORY
5948  036610  005737  001474                  TST     MARGIN                  ;SETUP CONDITION CODES - NEGATIVE IS UNDERFLOW
5949  036614  000207                          RETURN
5950
5951  036616                          SETMAR: SUBTST  <<SUBR  SET HIGHEST MARGIN>>
 (3)                                  ;*******************************************************************************
 (3)                                  ;*SUBTEST        SUBR    SET HIGHEST MARGIN
 (2)                                  ;*******************************************************************************
5952                                          ;RECOGNIZES THE "INHIBIT MARGINS SWITCH"
5953                                          .ENABL  LSB
5954  036616                                  IF #SW11 SET.IN @SWR OR QVFLAG NE #0
```

```
  (5)  036616  032777  004000  143354                                                  BIT #SW11,@SWR
  (5)  036624  001003                                                                  BNE L345
  (6)  036626  005737  002016                                                          TST QVFLAG
  (8)  036632  001401                                                                  BEQ L346
  (4)  036634                                                                    L345:;;;;;;
 5955  036634                                  GOTO 1$
  (2)  036634  000420                                                                  BR 1$
 5956  036636                                END ;OF IF #SW11
  (3)  036636                                                                    L346:;;;;;;
 5957  036636  013737  001474  001754        MOV     MARGIN,OLDMARGIN
 5958  036644                                IF MKFLAG EQ #0
  (4)  036644  005737  001510                                                          TST MKFLAG
  (6)  036650  001004                                                                  BNE L347
 5959  036652  012737  000005  001474          MOV    #5,MARGIN              ;HIGHEST MJ11 MARGIN
 5960  036660                                ELSE
  (3)  036660  000403                                                                  BR L350
  (3)  036662                                                                    L347:;;;;;;
 5961  036662  012737  000003  001474          MOV    #3,MARGIN              ;HIGHEST MK11 MARGIN
 5962  036670                                END ;OF IF MKFLAG
  (3)  036670                                                                    L350:;;;;;;
 5963  036670  004737  036704                CALL    COREHISTORY
 5964  036674  000402                        BR      2$                     ;SKIP
 5965  036676  005037  001474          1$:   CLR     MARGIN                 ;SET MARGIN TO "0"
 5966  036702  000207                  2$:   RETURN
 5967                                        .DSABL  LSB
```

```
 5969  036704                          COREHISTORY:SUBTST      <<SUBR  REWRITE CORE HISTORY>>
  (3)                                  ;*****************************************************************************
  (3)                                  ;*SUBTEST        SUBR    REWRITE CORE HISTORY
  (2)                                  ;*****************************************************************************
 5970  036704                                  IF MKFLAG EQ #0 AND NULLFLAG EQ #0
  (4)  036704  005737  001510                                                                    TST MKFLAG
  (6)  036710  001040                                                                            BNE L351
  (6)  036712  005737  002014                                                                    TST NULLFLAG
  (8)  036716  001035                                                                            BNE L351
 5971  036720                                    IF OLDMARGIN NE MARGIN
  (4)  036720  023737  001754  001474                                                            CMP OLDMARGIN,MARGIN
  (6)  036726  001431                                                                            BEQ L352
 5972  036730                                      IF OLDMARGIN EQ #4 OR OLDMARGIN EQ #5 OR MARGIN EQ #4 OR MARGIN EQ #5
  (5)  036730  023727  001754  000004                                                            CMP OLDMARGIN,#4
  (5)  036736  001414                                                                            BEQ L353
  (7)  036740  023727  001754  000005                                                            CMP OLDMARGIN,#5
  (7)  036746  001410                                                                            BEQ L353
  (8)  036750  023727  001474  000004                                                            CMP MARGIN,#4
  (8)  036756  001404                                                                            BEQ L353
  (8)  036760  023727  001474  000005                                                            CMP MARGIN,#5
  (10) 036766  001011                                                                            BNE L354
  (6)  036770                                                                          L353:;;;;;;
 5973  036770  013700  001474                        MOV       MARGIN,R0
 5974  036774  006300                                ASL       R0
 5975  036776  010037  177750                        MOV       R0,MAINT                ;SET MARGINS
 5976
 5977  037002                                        LET R2 := ONES
  (3)  037002  013702  002154                                                                    MOV ONES,R2
 5978  037006  004737  031124                        CALL      BACKGND
 5979  037012                                      END ;OF IF
  (3)  037012                                                                          L354:;;;;;;
 5980  037012                                    END ;OF IF
  (3)  037012                                                                          L352:;;;;;;
 5981  037012                                  END ;OF IF
  (3)  037012                                                                          L351:;;;;;;
 5982  037012  005037  002014                  CLR     NULLFLAG
 5983  037016  000207                          RETURN
 5984  037020                          BOOT:   SUBTST  <<BOOTSTRAP ROUTINE>>
  (3)                                  ;*****************************************************************************
  (3)                                  ;*SUBTEST        BOOTSTRAP ROUTINE
  (2)                                  ;*****************************************************************************
 5985                                          ;INITIALIZE ALL CSR'S
 5986                                          ;CLEAR ANY MARGINS
 5987                                          ;UNRELOCATE IF NECESSARY
 5988                                          ;FLUSH OUT ANY DBE'S
 5989                                          ;TURN OFF MEMORY MANAGEMENT
 5990                                          ;TURN OFF THE UNIBUS MAP
 5991                                          ;BOOT RK0 OR RK1
 5992  037020  104472                          ECCINIT
 5993  037022                                  SET4    #BOOT1
  (1)  037022  012737  037062  000004          MOV     #BOOT1,4                        ;SET NEW VECTOR
 5994  037030  005037  177750                  CLR     MAINT
 5995  037034                                  IF RLFLAG NE #0 THEN $CALL UNRELOCATE
  (4)  037034  005737  001516                                                                    TST RLFLAG
  (6)  037040  001402                                                                            BEQ L355
  (4)  037042  004737  035340                                                                    JSR PC,UNRELOCATE
```

```
  (3)  037046                                                                  L355:;:::::
 5996  037046  004737  021206                  CALL    MT0030          ;FLUSH OUT DBE'S
 5997  037052  104421                          DEENERGIZE
 5998  037054  042737  000040  172516          BIC     #BIT5,MMR3                      ;TURN OFF THE UNIBUS MAP
 5999  037062  005001                  BOOT1:  CLR     R1
 6000  037064  000005                  1$:     RESET
 6001  037066  012700  177406                  MOV     #177406,R0
 6002  037072  010160  000004                  MOV     R1,4(R0)
 6003  037076  012710  177400                  MOV     #177400,(R0)
 6004  037102  012740  000005                  MOV     #5,-(R0)
 6005  037106  105710                  2$:     TSTB    (R0)
 6006  037110  100376                          BPL     2$
 6007  037112  062701  020000                  ADD     #BIT13,R1
 6008  037116  005710                          TST     (R0)
 6009  037120  100761                          BMI     1$
 6010  037122  005007                          CLR     PC
```

```
 6012  037124                          EXIT:   SUBTST  <<HALT PROGRAM>>
  (3)                                ;*****************************************************************************
  (3)                                ;*SUBTEST       HALT PROGRAM
  (2)                                ;*****************************************************************************
 6013  037124  004737  037156                CALL    SHUTUP
 6014  037130                          EXIT2:  IF APTFLAG NE #0 OR ACTFLAG NE #0
   (5)  037130  005737  002022                                                    TST APTFLAG
   (5)  037134  001003                                                            BNE L356
   (6)  037136  005737  002020                                                    TST ACTFLAG
   (8)  037142  001402                                                            BEQ L357
   (4)  037144                                                          L356:::::::
 6015  037144  000777                        BR      .
 6016  037146                              ELSE
   (3)  037146  000403                                                            BR L360
   (3)  037150                                                          L357:::::::
 6017  037150  000000                  $EXHALT:  HALT
 6018  037152  000137  003614                  JMP     START
 6019  037156                              END ;OF IF APTFLAG
   (3)  037156                                                          L360:::::::
 6020
 6021  037156                          SHUTUP: SUBTST  <<SHUTDOWN DIAGNOSTIC>>
  (3)                                ;*****************************************************************************
  (3)                                ;*SUBTEST       SHUTDOWN DIAGNOSTIC
  (2)                                ;*****************************************************************************
 6022                                        ;INITIALIZE ALL CSR'S
 6023                                        ;CLEAR MARGINS
 6024                                        ;UNRELOCATE
 6025                                        ;FLUSH OUT DBE'S
 6026                                        ;RESTORE LOADERS
 6027                                        ;TURN OFF MEMORY MANAGEMENT
 6028                                        ;UNMAP THE UNIBUS MAP
 6029  037156  104472                      ECCINIT
 6030  037160  005037  177750              CLR     MAINT
 6031  037164                              IF RLFLAG NE #0 THEN $CALL UNRELOCATE
   (4)  037164  005737  001516                                                    TST RLFLAG
   (6)  037170  001402                                                            BEQ L361
   (4)  037172  004737  035340                                                    JSR PC,UNRELOCATE
   (3)  037176                                                          L361:::::::
 6032  037176                              IF QUICK EQ #0
   (4)  037176  005737  002122                                                    TST QUICK
   (6)  037202  001002                                                            BNE L362
 6033  037204  004737  021206                  CALL  MTO030          ;FLUSH OUT DBE'S
 6034  037210                              END ;OF IF QUICK
   (3)  037210                                                          L362:::::::
 6035  037210  012700  000001              MOV     #1,R0           ;DESTINATION BANK
 6036  037214  013701  002114              MOV     LOADHOME,R1     ;SOURCE BANK
 6037  037220  004737  035570              CALL    BANKMOV
 6038  037224  104421                      DEENERGIZE
 6039  037226  042737  000040  172516      BIC     #BIT5,MMR3              ;TURN OFF UNIBUS MAP
 6040  037234  000207                      RETURN
 6041
 6042  037236                        APTDOWN:SUBTST  <<APT SHUTDOWN SEQUENCE>>
  (3)                                ;*****************************************************************************
  (3)                                ;*SUBTEST       APT SHUTDOWN SEQUENCE
  (2)                                ;*****************************************************************************
 6043  037236                              MAP     #0
```

```
                                                                                    MOV R3,-(SP)
  (3)  037236  010346
  (1)  037240  012703  000000                MOV     #0,R3
  (1)  037244  004737  034510                CALL    MAPPER
                                                                                    MOV (SP)+,R3
  (3)  037250  012603
6044   037252                                SUPERVISOR
  (1)  037252  052737  040000  177776        BIS     #BIT14,PSW          ;DO IT TO IT!
6045   037260  012737  037236  060024        MOV     #APTDOWN,FIRST+24
6046   037266  012737  000340  060026        MOV     #340,FIRST+26
6047   037274  012737  000000  117236        MOV     #0,FIRST+APTDOWN
6048   037302  104417                        KERNEL
6049   037304  000000              APTHLT:   HALT
```

```
                                        SUBTST  <<BLOCK MOVE SUBROUTINE>>
                                ;*******************************************************************************
                                ;*SUBTEST        BLOCK MOVE SUBROUTINE
                                ;*******************************************************************************
                                        ;BLOCK3 HAS 3 ARGUEMENTS
                                        ;BLOCK2 HAS 2 ARGUEMENTS
                                        ;BLOCK1 HAS 1 ARGUEMENTS
                                        ;
                                        ;ALL ARE CALLED BY THE BMOV MACRO
                                        .ENABL  LSB
037306                          BLOCK1: PUSH    R0,R1,R2
(2) 037306  010046                                                      MOV R0,-(SP)
(3) 037310  010146                                                      MOV R1,-(SP)
(4) 037312  010246                                                      MOV R2,-(SP)
037314  012702  177640                  MOV     #FASTCITY,R2
037320  012701  000020                  MOV     #16.,R1
037324  000413                          BR      3$

037326                          BLOCK2: PUSH    R0,R1,R2
(2) 037326  010046                                                      MOV R0,-(SP)
(3) 037330  010146                                                      MOV R1,-(SP)
(4) 037332  010246                                                      MOV R2,-(SP)
037334  012701  000020                  MOV     #16.,R1
037340  000404                          BR      2$

037342                          BLOCK3: PUSH    R0,R1,R2
(2) 037342  010046                                                      MOV R0,-(SP)
(3) 037344  010146                                                      MOV R1,-(SP)
(4) 037346  010246                                                      MOV R2,-(SP)
037350  012501                          MOV     (R5)+,R1
037352  012502                  2$:     MOV     (R5)+,R2
037354  012500                  3$:     MOV     (R5)+,R0

037356  012022                  1$:     MOV     (R0)+,(R2)+
037360  077102                          SOB     R1,1$
037362                                  POP     R2,R1,R0
(2) 037362  012602                                                      MOV (SP)+,R2
(3) 037364  012601                                                      MOV (SP)+,R1
(4) 037366  012600                                                      MOV (SP)+,R0
037370  000205                          RTS     R5
                                        .DSABL  LSB
```

```
                                .TITLE  FIELD SERVICE MODE

                        FIELDSERVICE:SUBTST     ;<SUBR  FIELD SERVICE COMMAND MODE>>
                        ;*******************************************************************************
                        ;*SUBTEST       SUBR    FIELD SERVICE COMMAND MODE
                        ;*******************************************************************************
                                SAVREG
                                TYPE    MSG020          ;FIELD SERVICE COMMAND MODE
                                TYPEIT  ,MSG020

                                IF RLFLAG NE #0                                         TST RLFLAG
                                                                                        BEQ L363
                                  TYPE  MSG048          ;NOT AVAILABLE NOW - TRY LATER!
                                TYPEIT  ,MSG048
                                  RESREG
                                  RETURN
                                END ;OF IF RLFLAG
                                                                        L363:;;;;;;
                                IF NOFSMODE NE #0                                       TST NOFSMODE
                                                                                        BEQ L364
                                  TYPE  MSG048          ;NOT AVAILABLE NOW - TRY LATER!
                                TYPEIT  ,MSG048
                                  RESREG
                                  RETURN
                                END ;OF IF NOFSMODE
                                                                        L364:;;;;;;
                                PUSH    CONTRL,CSRNO    ;SAVE CACHE STATUS              MOV CONTRL,-(SP)
                                                                                        MOV CSRNO,-(SP)
                                CACHOFF
                        FS1:    TYPE    MSG026          ;COMMAND:
                                TYPEIT  ,MSG026
                                RDDEC                   ;READ A DECIMAL NUMBER
                                POP     R0              ;COMMAND --> R0                 MOV (SP)+,R0

                                CMP     R0,#12.
                                BLOS    1$
                                TYPE    MSG021
                                TYPEIT  ,MSG021
                                BR      FS1
                        1$:     CASE R0                                                 MOV R0,-(SP)
                                                                                        ASL @SP
                                                                                        JSR PC,L365
                                  FSCMD0
                                  FSCMD1
                                  FSCMD2
                                  FSCMD3
                                  FSCMD4
                                  FSCMD5
                                  FSCMD6
                                  FSCMD7
                                  FSCMD8
                                  FSCMD9
```

```
 6114  037526  043032                    FCMD10
 6115  037530  043176                    FCMD11
 6116  037532  043570                    FCMD12
 6117  037534                          END ;OF CASE
  (3)  037534                                                    L365:;;;;;;
  (2)  037534  062616                                                  ADD (SP)+,@SP
  (2)  037536  013646                                                  MOV @(SP)+,-(SP)
  (2)  037540  004736                                                  JSR PC,@(SP)+
 6118  037542  000741                    BR      FS1
```

```
                                FSCMD0: SUBTST  <<COMMAND 0     EXIT>>
                                ;*****************************************************************************
                                ;*SUBTEST        COMMAND 0       EXIT
                                ;*****************************************************************************
6121  037544  062706  000002            ADD     #2,SP
6122  037550                            POP     CSRNO,CNTRL
 (2)  037550  012637  001634                                                    MOV (SP)+,CSRNO
 (3)  037554  012637  177746                                                    MOV (SP)+,CNTRL
6123  037560  104416                    RESREG
6124  037562  000207                    RETURN
6125
6126  037564                            FSCMD1: SUBTST  <<FS    COMMAND 1       READ CSR>>
                                ;*****************************************************************************
                                ;*SUBTEST        FS      COMMAND 1       READ CSR
                                ;*****************************************************************************
6127  037564  004737  043616            CALL    WHICHCSR
6128  037570  010637  001752            MOV     SP,FSSTACK
6129  037574                            SET4    #1$
 (1)  037574  012737  037630  000004    MOV     #1$,4                   ;SET NEW VECTOR
6130  037602  104426                    READCSR
6131  037604                            SET     NOERROR
 (1)  037604  012737  177777  002112    MOV     #-1,NOERROR
6132  037612  104026                    ERROR   +26             ;USE ERROR ROUTINE FOR PRINTOUT
6133  037614                            RES4
 (1)  037614  012737  032466  000004    MOV     #TIMEOUT,4              ;RESTORE OLD (DEFAULT) VECTOR
 (1)  037622  005037  177766            CLR     CPUERR
6134  037626  000207                    RETURN
6135  037630                    1$:     TYPE    MSG025          ;THIS CSR DOES NOT EXIST
 (1)  037630  104401  067721            TYPEIT  ,MSG025
6136  037634  013706  001752            MOV     FSSTACK,SP
6137  037640                            RES4
 (1)  037640  012737  032466  000004    MOV     #TIMEOUT,4              ;RESTORE OLD (DEFAULT) VECTOR
 (1)  037646  005037  177766            CLR     CPUERR
6138  037652  000207                    RETURN
```

```
 6140  037654                                FSCMD2: SUBTST  <<FS     COMMAND 2       LOAD CSR>>
   (3)                                       ;*****************************************************************************
   (3)                                       ;*SUBTEST       FS       COMMAND 2       LOAD CSR
   (2)                                       ;*****************************************************************************
 6141  037654  004737  043616                        CALL    WHICHCSR
 6142  037660  010637  001752                        MOV     SP,FSSTACK
 6143  037664                                        SET4    #1$
   (1) 037664  012737  040002  000004                MOV     #1$,4                   ;SET NEW VECTOR
 6144  037672  104426                               READCSR
 6145  037674                                        TYPE    MSG027
   (1) 037674  104401  067764                        TYPEIT  ,MSG027
 6146  037700                                        SET     NOERROR
   (1) 037700  012737  177777  002112                MOV     #-1,NOERROR
 6147  037706  104026                               ERROR   +26             ;USE ERROR ROUTINE FOR PRINTOUT
 6148  037710                                        RES4
   (1) 037710  012737  032466  000004                MOV     #TIMEOUT,4              ;RESTORE OLD (DEFAULT) VECTOR
   (1) 037716  005037  177766                        CLR     CPUERR
 6149  037722                                        TYPE    MSG023          ;FIRST CSR WORD
   (1) 037722  104401  067655                        TYPEIT  ,MSG023
 6150  037726  104413                               RDOCT                   ;READ AN OCTAL NUMBER
 6151  037730                                        POP     CSR             ;PUT IN IN LOC "CSR"
   (2) 037730  012637  001526                                                                MOV (SP)+,CSR
 6152  037734                                        TYPE    MSG024          ;2ND CSR WORD
   (1) 037734  104401  067677                        TYPEIT  ,MSG024
 6153  037740  104413                               RDOCT                   ;READ AN OCTAL NUMBER
 6154  037742                                        POP     CSR+2           ;PUT IN IN LOC "CSR+2"
   (2) 037742  012637  001530                                                                MOV (SP)+,CSR+2
 6155  037746                                        SET     INTOVR          ;INTERNAL INTERLEAVE OVERRIDE
   (1) 037746  012737  177777  002042                MOV     #-1,INTOVR
 6156  037754  104425                               LOADCSR
 6157  037756  005037  002042                        CLR     INTOVR
 6158  037762  104426                               READCSR
 6159  037764                                        TYPE    MSG028
   (1) 037764  104401  070001                        TYPEIT  ,MSG028
 6160  037770                                        SET     NOERROR
   (1) 037770  012737  177777  002112                MOV     #-1,NOERROR
 6161  037776  104026                               ERROR   +26             ;USE FOR PRINTOUT - NOT AN ERROR
 6162  040000  000207                               RETURN
 6163  040002                                1$:     TYPE    MSG025          ;THIS CSR DOES NOT EXIST
   (1) 040002  104401  067721                        TYPEIT  ,MSG025
 6164  040006  013706  001752                        MOV     FSSTACK,SP
 6165  040012                                        RES4
   (1) 040012  012737  032466  000004                MOV     #TIMEOUT,4              ;RESTORE OLD (DEFAULT) VECTOR
   (1) 040020  005037  177766                        CLR     CPUERR
 6166  040024  000207                               RETURN
```

```
 6168  040026                          FSCMD3: SUBTST  <<FS     COMMAND 3       EXAMINE MEMORY>>
  (3)                                  ;*****************************************************************************
  (3)                                  ;*SUBTEST        FS      COMMAND 3       EXAMINE MEMORY
  (2)                                  ;*****************************************************************************
 6169  040026                                  PUSH    BANK,NOPAR,PARTHERE,4
  (2)  040026  013746  001470                                                                    MOV BANK,-(SP)
  (3)  040032  013746  001464                                                                    MOV NOPAR,-(SP)
  (4)  040036  013746  001750                                                                    MOV PARTHERE,-(SP)
  (5)  040042  013746  000004                                                                    MOV 4,-(SP)
 6170  040046  012737  000002  001464                MOV     #2,NOPAR        ;INDICATE PARITY ACTION
 6171  040054                                  TYPE    MSG029          ;EXAMINE MEMORY
  (1)  040054  104401  070015                  TYPEIT  ,MSG029
 6172  040060                          1$:     TYPE    MSG031          ;PHYSICAL ADDRESS (0-16777776)??
  (1)  040060  104401  070054                  TYPEIT  ,MSG031
 6173  040064  104413                          RDOCT                   ;READ OCTAL NUMBER ONTO STACK & $HIOCT
 6174  040066  013737  057464  001470          MOV     $HIOCT,BANK     ;PUT MSB'S IN BANK
 6175  040074                                  POP     R0              ;PUT LSB'S IN R0
  (2)  040074  012600                                                                            MOV (SP)+,R0
 6176  040076  000241                          CLC
 6177  040100  006100                          ROL     R0
 6178  040102  006137  001470                  ROL     BANK
 6179  040106  000241                          CLC
 6180  040110  006000                          ROR     R0
 6181  040112                                  IF BANK GT #167 THEN GOTO 1$    ;CHECK FOR BANK TOO HIGH
  (5)  040112  023727  001470  000167                                                            CMP BANK,#167
  (5)  040120  003357                                                                            BGT 1$
 6182  040122  062700  060000                  ADD     #FIRST,R0
 6183  040126                                  IF #BIT0 SET IN R0 THEN GOTO 1$ ;CHECK FOR ODD ADDRESS
  (5)  040126  032700  000001                                                                    BIT #BIT0,R0
  (5)  040132  001352                                                                            BNE 1$
 6184  040134                                  IF R0 HI #LAST THEN GOTO 1$     ;CHECK FOR ADDRESS OVER 16K
  (5)  040134  020027  157776                                                                    CMP R0,#LAST
  (5)  040140  101347                                                                            BHI 1$
 6185  040142  012737  040214  001750          MOV     #3$,PARTHERE    ;INCASE OF ABORTS
 6186  040150                                  SET4    #4$             ;INCASE OF TIMEOUT TRAPS
  (1)  040150  012737  040222  000004          MOV     #4$,4                   ;SET NEW VECTOR
 6187  040156                                  MAP     BANK
  (3)  040156  010346                                                                            MOV R3,-(SP)
  (1)  040160  013703  001470                  MOV     BANK,R3
  (1)  040164  004737  034510                  CALL    MAPPER
  (3)  040170  012603                                                                            MOV (SP)+,R3
 6188  040172                                  SUPERVISOR
  (1)  040172  052737  040000  177776          BIS     #BIT14,PSW              ;DO IT TO IT!
 6189  040200  011001                          MOV     (R0),R1
 6190  040202  104417                          KERNEL
 6191  040204                                  TYPOCS  R1
  (1)  040204  010146                          MOV     R1,-(SP)        ;;SAVE R1 FOR TYPEOUT
  (1)  040206  104403                          TYPOS                   ;;GO TYPE--OCTAL ASCII
  (1)  040210      006                         .BYTE   6               ;;TYPE 6 DIGITS
  (1)  040211      000                         .BYTE   0               ;;SUPPRESS LEADING ZEROS
 6192  040212  000410                          BR      EXCMD3
 6193
 6194  040214                          3$:     TYPE    MSG032                  ;PARITY ABORT
  (1)  040214  104401  070114                  TYPEIT  ,MSG032
 6195  040220  000405                          BR      EXCMD3
 6196
```

```
 6197  040222  062706  000004          4$:   ADD     #4,SP                   ;FIX STACK
 6198  040226                                TYPE    MSG033                  ;TIMEOUT TRAP
  (1)  040226  104401  070132                TYPEIT  ,MSG033
 6199  040232  000400                        BR      EXCMD3
 6200
 6201  040234  104417                  EXCMD3: KERNEL
 6202  040236                                POP     4,PWRTHERE,NOPAR,BANK
  (2)  040236  012637  000004                                                  MOV (SP)+,4
  (3)  040242  012637  001750                                                  MOV (SP)+,PARTHERE
  (4)  040246  012637  001464                                                  MOV (SP)+,NOPAR
  (5)  040252  012637  001470                                                  MOV (SP)+,BANK
 6203  040256                                RES4
  (1)  040256  012737  032466  000004        MOV     #TIMEOUT,4              ;RESTORE OLD (DEFAULT) VECTOR
  (1)  040264  005037  177766                CLR     CPUERR
 6204  040270  000207                        RETURN
```

```
                                FSCMD4: SUBTST  <<FS     COMMAND 4        MODIFY MEMORY>>
                                ;*****************************************************************************
                                ;*SUBTEST        FS       COMMAND 4        MODIFY MEMORY
                                ;*****************************************************************************
                                        PUSH    BANK,NOPAR,PARTHERE,4
                                                                                        MOV BANK,-(SP)
                                                                                        MOV NOPAR,-(SP)
                                                                                        MOV PARTHERE,-(SP)
                                                                                        MOV 4,-(SP)
                                        MOV     #3,NOPAR                ;INDICATE PARITY ACTION
                                        TYPE    MSG036                  ;MODIFY MEMORY
                                        TYPEIT  ,MSG036
                                1$:     TYPE    MSG031          ;PHYSICAL ADDRESS (0-16777776)??
                                        TYPEIT  ,MSG031
                                        RDOCT                   ;READ OCTAL NUMBER ONTO STACK & $HIOCT
                                        MOV     $HIOCT,BANK     ;PUT MSB'S IN BANK
                                        POP     R0              ;PUT LSB'S IN R0
                                                                                        MOV (SP)+,R0
                                        CLC
                                        ROL     R0
                                        ROL     BANK
                                        CLC
                                        ROR     R0
                                        IF BANK GT #167 THEN GOTO 1$    ;CHECK FOR BANK TOO HIGH
                                                                                        CMP BANK,#167
                                                                                        BGT 1$
                                        ADD     #FIRST,R0
                                        IF #BIT0 SET.IN R0 THEN GOTO 1$ ;CHECK FOR ODD ADDRESS
                                                                                        BIT #BIT0,R0
                                                                                        BNE 1$
                                        IF R0 HI #LAST THEN GOTO 1$     ;CHECK FOR ADDRESS OVER 16K
                                                                                        CMP R0,#LAST
                                                                                        BHI 1$
                                        MOV     #3$,PARTHERE            ;INCASE OF ABORTS
                                        SET4    #4$                     ;INCASE OF TIMEOUT TRAPS
                                        MOV     #4$,4                   ;SET NEW VECTOR
                                        MAP     BANK
                                                                                        MOV R3,-(SP)
                                        MOV     BANK,R3
                                        CALL    MAPPER
                                                                                        MOV (SP)+,R3
                                        SUPERVISOR
                                        BIS     #BIT14,PSW              ;DO IT TO IT!
                                        MOV     (R0),R1
                                        KERNEL                  ;GETTING HERE MEANS WE GOT LUCKY - NO TRAPS
                                        BR      5$

                                3$:     TYPE    MSG032                  ;PARITY ABORT
                                        TYPEIT  ,MSG032
                                        BR      EXCMD4                  ;EXIT

                                4$:     ADD     #4,SP           ;FIX STACK
                                        TYPE    MSG033          ;TIMEOUT TRAP
                                        TYPEIT  ,MSG033
                                        BR      EXCMD4                  ;EXIT
```

```
6238  040472                          5$:     TYPE    MSG037                  ;OLD DATA WAS
 (1)  040472  104401  070307                  TYPEIT  ,MSG037
6239  040476                                  TYPOCS  R1                      ;PRINT IT
 (1)  040476  010146                          MOV     R1,-(SP)        ;;SAVE R1 FOR TYPEOUT
 (1)  040500  104403                          TYPOS                   ;;GO TYPE--OCTAL ASCII
 (1)  040502     006                          .BYTE   6               ;;TYPE 6 DIGITS
 (1)  040503     000                          .BYTE   0               ;;SUPPRESS LEADING ZEROS
6240  040504                                  TYPE    MSG039                  ;INPUT NEW DATA
 (1)  040504  104401  070344                  TYPEIT  ,MSG039
6241  040510  104413                          RDOCT                           ;READ ON OCTAL NUMBER ONTO THE STACK
6242  040512                                  POP     R1                      ;GET NEW NUMBER
 (2)  040512  012601                                                                                  MOV (SP)+,R1
6243  040514                                  SUPERVISOR
 (1)  040514  052737  040000  177776          BIS     #BIT14,PSW              ;DO IT TO IT!
6244  040522  010110                          MOV     R1,(R0)                 ;PUT IT IN MEMORY
6245  040524  011001                          MOV     (R0),R1                 ;READ IT AGAIN
6246  040526  104417                          KERNEL
6247  040530                                  TYPE    MSG038                  ;DATA IS NOW
 (1)  040530  104401  070326                  TYPEIT  ,MSG038
6248  040534                                  TYPOCS  R1                      ;PRINT IT
 (1)  040534  010146                          MOV     R1,-(SP)        ;;SAVE R1 FOR TYPEOUT
 (1)  040536  104403                          TYPOS                   ;;GO TYPE--OCTAL ASCII
 (1)  040540     006                          .BYTE   6               ;;TYPE 6 DIGITS
 (1)  040541     000                          .BYTE   0               ;;SUPPRESS LEADING ZEROS
6249
6250  040542  104417                  EXCMD4: KERNEL
6251  040544                                  POP     4,PARTHERE,NOPAR,BANK
 (2)  040544  012637  000004                                                                          MOV (SP)+,4
 (3)  040550  012637  001750                                                                          MOV (SP)+,PARTHERE
 (4)  040554  012637  001464                                                                          MOV (SP)+,NOPAR
 (5)  040560  012637  001470                                                                          MOV (SP)+,BANK
6252  040564                                  RES4
 (1)  040564  012737  032466  000004          MOV     #TIMEOUT,4              ;RESTORE OLD (DEFAULT) VECTOR
 (1)  040572  005037  177766                  CLR     CPUERR
6253  040576  000207                          RETURN
```

```
                                FSCMD5: SUBTST  <<FS    COMMAND 5       SELECT BANK,MARGIN, & PATTERN>>
                                ;*********************************************************************
                                ;*SUBTEST        FS      COMMAND 5       SELECT BANK,MARGIN, & PATTERN
                                ;*********************************************************************
                                        PUSH    BANK,PATTERN,MARGIN,CSRFIRST,PCBUMP,TKVEC,TKVEC+2
                                                                                        MOV BANK,-(SP)
                                                                                        MOV PATTERN,-(SP)
                                                                                        MOV MARGIN,-(SP)
                                                                                        MOV CSRFIRST,-(SP)
                                                                                        MOV PCBUMP,-(SP)
                                                                                        MOV TKVEC,-(SP)
                                                                                        MOV TKVEC+2,-(SP)
                                        MOV     MARGIN,OLDMARGIN
                                        MOV     SP,FSSTACK              ;SAVE LAST GOOD STACK POINTER
                                        TYPE    MSG040                  ;SELECT BANK, MARGIN, & PATTERN TEST
                                        TYPEIT  ,MSG040
                                1$:     TYPE    MSG030                  ;BANK(0-167)?
                                        TYPEIT  ,MSG030
                                        RDOCT                           ;READ AN OCTAL NUMBER ONTO THE STACK
                                        POP     BANK                    ;PUT IT IN BANK
                                                                                        MOV (SP)+,BANK
                                        IF BANK GT #167 THEN GOTO 1$    ;CHECK FOR BANK TO HIGH
                                                                                        CMP BANK,#167
                                                                                        BGT 1$

                                        MOV     BANK,R1
                                        MUL     #6,R1
                                        IF CPUBIT OFF.IN CONFIG(R1)
                                                                                        BIT CPUBIT,CONFIG(R1)
                                                                                        BNE L366
                                          TYPE  MSG041                  ;BANK NOT ACCESSABLE
                                        TYPEIT  ,MSG041
                                          GOTO 1$
                                                                                        BR 1$
                                        END ;OF IF
                                                                                L366:::::::

                                2$:     TYPE    MSG042                  ;PATTERN(0-37)?
                                        TYPEIT  ,MSG042
                                        RDOCT                           ;READ AN OCTAL NUMBER ONTO THE STACK
                                        POP     PATTERN                 ;PUT IT IN PATTERN
                                                                                        MOV (SP)+,PATTERN
                                        IF PATTERN GT #37 THEN GOTO 2$  ;CHECK FOR PATTERN TO HIGH
                                                                                        CMP PATTERN,#37
                                                                                        BGT 2$

                                        IF PATTERN EQ #0
                                                                                        TST PATTERN
                                                                                        BNE L367
                                          TYPE  MSG043                  ;PATTERN 0 DATA IS?
                                        TYPEIT  ,MSG043
                                          RDOCT                         ;READ AN OCTAL NUMBER ONTO THE STACK
                                          POP   R2                      ;PUT IT IN R2
                                                                                        MOV (SP)+,R2
                                        END ;OF IF
                                                                                L367:::::::
```

```
                                CMD5A:  TYPE    MSG044                  ;MARGIN(0,2,3,4,5)?
  (1) 040762 104401 070526              TYPEIT  ,MSG044
6283  040766 104413                     RDOCT                           ;READ AN OCTAL NUMBER ONTO THE STACK
6284  040770                            POP     MARGIN                  ;PUT IN IN MARGIN
  (2) 040770 012637 001474                                                      MOV (SP)+,MARGIN
6285  040774                            IF MARGIN GT #5 OR MARGIN EQ #1
  (5) 040774 023727 001474 000005                                               CMP MARGIN,#5
  (5) 041002 003004                                                             BGT L370
  (6) 041004 023727 001474 000001                                               CMP MARGIN,#1
  (8) 041012 001001                                                             BNE L371
  (4) 041014                                                            L370:::::::
6286  041014                              GOTO CMD5A
  (2) 041014 000762                                                             BR CMD5A
6287  041016                            END ;OF IF
  (3) 041016                                                            L371:::::::
6288  041016                            IF MARGIN NE #0 AND #SW11 SET.IN @SWR
  (4) 041016 005737 001474                                                      TST MARGIN
  (6) 041022 001407                                                             BEQ L372
  (6) 041024 032777 004000 141146                                               BIT #SW11,@SWR
  (8) 041032 001403                                                             BEQ L372
6289  041034                              TYPE  MSG045                  ;MARGINS INHIBITED BY SW5
  (1) 041034 104401 070553            TYPEIT  ,MSG045
6290  041040                              GOTO CMD5A
  (2) 041040 000750                                                             BR CMD5A
6291  041042                            END ;OF IF MARGIN
  (3) 041042                                                            L372:::::::
6292
6293  041042                            MAP     BANK
  (3) 041042 010346                                                             MOV R3,-(SP)
  (1) 041044 013703 001470            MOV     BANK,R3
  (1) 041050 004737 034510            CALL    MAPPER
  (3) 041054 012603                                                             MOV (SP)+,R3
6294  041056 004737 036052            CALL    EXBANK                  ;SET NEW MARGINS
6295  041062                          IF RRFLAG NE #0
  (4) 041062 005737 001514                                                      TST RRFLAG
  (6) 041066 001403                                                             BEQ L373
6296  041070                              TYPE  MSG049                  ;BANK REQUIRES RELOCATION
  (1) 041070 104401 070720            TYPEIT  ,MSG049
6297  041074                              GOTO CMD5C
  (2) 041074 000527                                                             BR CMD5C
6298  041076                          END ;OF IF RRFLAG
  (3) 041076                                                            L373:::::::
6299  041076 004737 036704            CALL    COREHISTORY
6300  041102                          TYPE    MSG046          ;TO ESCAPE TYPE ANY KEY!
  (1) 041102 104401 070606            TYPEIT  ,MSG046
6301  041106 012737 041354 000060     MOV     #CMD5C,TKVEC
6302  041114 012737 000340 000062     MOV     #340,TKVEC+2
6303  041122 017700 141060            MOV     @$TKB,R0                ;KILL ANY OLD INTERRUPT
6304  041126 042737 000200 177776     BIC     #BIT7,PSW               ;LOWER CPU PRIORITY TO 140
6305  041134 052777 000100 141042     BIS     #BIT6,@$TKS             ;ENABLE KEYBOARD INTERRUPTS
6306
6307  041142 013700 001474            MOV     MARGIN,R0               ;SET MARGINS
6308  041146 006300                   ASL     R0                      ;SET MARGINS
6309
6310  041150 010037 177750            MOV     R0,MAINT
6311
```

```
                                        SET     HEADER
  6312  041154
    (1) 041154  012737  177777  002146      MOV     #-1,HEADER
                                        SET     MUT
  6313  041162
    (1) 041162  012737  177777  001500      MOV     #-1,MUT
  6314  041170  005037  001734    CMDSB:    CLR     PASFLG
  6315  041174  012737  060000  001710      MOV     #FIRST,CSRFIRST
  6316  041202                            IF #SW0 SET.IN @SWR
    (4) 041202  032777  000001  140770                                          BIT #SW0,@SWR
    (6) 041210  001402                                                          BEQ L374
  6317  041212  104470                        ECCDIS
  6318  041214                            ELSE
    (3) 041214  000401                                                          BR L375
    (3) 041216                                                          L374:::::::
  6319  041216  104502                        CLRCSR
  6320  041220                            END ;OF IF
    (3) 041220                                                          L375:::::::
  6321  041220  012737  000002  001464      MOV     #2,NOPAR            ;PARITY ACTION
  6322  041226  012737  000002  001764      MOV     #2,PCBUMP           ;TRAPS ADD 2 TO PC
  6323  041234  013700  001502              MOV     PATTERN,R0
  6324  041240  006300                      ASL     R0
  6325  041242  004770  041254              CALL    @FSPAT(R0)
  6326  041246  005037  001464              CLR     NOPAR
  6327  041252  000746                      BR      CMDSB               ;LOOP TILL KEYBOARD INTERRUPT
  6328
  6329  041254  016312            FSPAT:    MT0000  ;<1 SEC         DATA PATTERN TEST
  6330  041256  016346                      MT0001  ;<1 SEC         ADDRESS TEST
  6331  041260  016406                      MT0002  ;<1 SEC         COMPLEMENT ADDRESS TEST
  6332  041262  016450                      MT0003  ; 1 SEC         3 XOR 9 WORST CASE NOISE TEST
  6333  041264  016616                      MT0004  ; 1 SEC         ROTATING ZEROS TEST
  6334  041266  016700                      MT0005  ; 1 SEC         ROTATING ONES TEST
  6335  041270  016760                      MT0006  ;<1 SEC         INITIAL DATA TEST
  6336  041272  017014                      MT0007  ;<1 SEC         ADDRESS BIT TEST
  6337  041274  017056                      MT0010  ;<1 SEC         BYTE ADDRESSING TEST
  6338  041276  017112                      MT0011  ;<2 SEC         CREATE SINGLE BIT ERROR TEST
  6339  041300  017170                      MT0012  ;<1 SEC         WRITE BYTE CLEARS SBE TEST
  6340  041302  017246                      MT0013  ; 1 SEC         CREATE DOUBLE BIT ERROR TEST
  6341  041304  017322                      MT0014  ; 1 SEC         WRITE INHIBIT DURING DATIP WITH DBE
  6342  041306  017400                      MT0015  ; 1 SEC         WRITE INHIBIT OF BYTE WITH DBE
  6343  041310  017456                      MT0016  ;<1 SEC         WRITE INHIBIT OF WORD WITH DBE
  6344  041312  017534                      MT0017  ;<1 SEC         HOLDING 1'S & 0'S TEST
  6345  041314  017556                      MT0020  ;<1 SEC         MARCHING 1'S & 0'S IN CHECK BITS
  6346  041316  020212                      MT0021  ; 1 SEC         MARCHING 0'S & 1'S TEST
  6347  041320  020334                      MT0022  ; 9 SEC         REFRESH TEST
  6348  041322  020356                      MT0023  ;69 SEC         SHIFTING DIAGONAL TEST
  6349  041324  020404                      MT0024  ;58 SEC         FAST GALLOPING PATTERN TEST
  6350  041326  020442                      MT0025  ;<1 SEC         INTERRUPT ENABLE TEST
  6351  041330  020510                      MT0026  ;<1 SEC         RANDOM DATA TEST
  6352  041332  020640                      MT0027  ; 1 SEC         UNIQUE BANK TEST
  6353  041334  021206                      MT0030  ; 1 SEC         FLUSH OUT DBE'S TEST
  6354  041336  021430                      MT0031  ; 3 SEC         SOB-A-LONG TEST
  6355  041340  021532            MT0999;MT0032
  6356  041342  021532            MT0999;MT0033
  6357  041344  021532            MT0999;MT0034
  6358  041346  021532            MT0999;MT0035
  6359  041350  021532            MT0999;MT0036
  6360  041352  021532            MT0999;MT0037
```

```
041354  013706  001752          CMD5C:  MOV     FSSTACK,SP              ;RECOVER OLD STACK POINTER
041360  042777  000100  140616          BIC     #BIT6,@$TKS
041366                                  POP     TKVEC+2,TKVEC
041366  012637  000062                                                                  MOV (SP)+,TKVEC+2
041372  012637  000060                                                                  MOV (SP)+,TKVEC
041376  117700  140604                  MOVB    @$TKB,R0                ;GET CHARACTER TO GET RID OF FLAG
041402                                  POP     PCBUMP,CSRFIRST,MARGIN
041402  012637  001764                                                                  MOV (SP)+,PCBUMP
041406  012637  001710                                                                  MOV (SP)+,CSRFIRST
041412  012637  001474                                                                  MOV (SP)+,MARGIN
041416  004737  036704                  CALL    COREHISTORY             ;RESTORE OLD MARGINS
041422                                  POP     PATTERN,BANK
041422  012637  001502                                                                  MOV (SP)+,PATTERN
041426  012637  001470                                                                  MOV (SP)+,BANK
041432                                  MAP     BANK                    ;REMAP OLD BANK
041432  010346                                                                          MOV R3,-(SP)
041434  013703  001470                  MOV     BANK,R3
041440  004737  034510                  CALL    MAPPER
041444  012603                                                                          MOV (SP)+,R3
041446  004737  036052                  CALL    EXBANK
041452  000207                          RETURN

041454                          FSCMD6: SUBTST  <<FS    COMMAND 6       TYPE CONFIGURATION MAP>>
                                ;*******************************************************************
                                ;*SUBTEST        FS      COMMAND 6       TYPE CONFIGURATION MAP
                                ;*******************************************************************
041454  004737  031162                  CALL    PCONFIG
041460  000207                          RETURN
```

```
6379  041462                        FSCMD7: SUBTST  <<FS    COMMAND 7       BATTERY BACKUP TEST>>
 (3)                                ;*****************************************************************************
 (3)                                ;*SUBTEST       FS      COMMAND 7       BATTERY BACKUP TEST
 (2)                                ;*****************************************************************************
6380  041462                              PUSH    BANK,PATTERN,MARGIN,CSRNO,TKVEC,TKVEC+2
 (2)  041462  013746  001470                                                                  MOV BANK,-(SP)
 (3)  041466  013746  001502                                                                  MOV PATTERN,-(SP)
 (4)  041472  013746  001474                                                                  MOV MARGIN,-(SP)
 (5)  041476  013746  001634                                                                  MOV CSRNO,-(SP)
 (6)  041502  013746  000060                                                                  MOV TKVEC,-(SP)
 (7)  041506  013746  000062                                                                  MOV TKVEC+2,-(SP)
6381  041512                              IF #SWD SET.IN @SWR                                  BIT #SWD,@SWR
 (4)  041512  032777  000001  140460                                                          BEQ L376
 (6)  041520  001402
6382  041522  104470                        ECCDIS
6383  041524                              ELSE
 (3)  041524  000401                                                                          BR L377
 (3)  041526                                                                        L376:;;;;;;
6384  041526  104502                        CLRCSR
6385  041530                              END ;OF IF
 (3)  041530                                                                        L377:;;;;;;
6386  041530                              TYPE    MSG050          ;BATTERY BACKUP TEST
 (1)  041530  104401  070752               TYPEIT  ,MSG050
6387                                                            ;WRITING & CHECKING ADDRESS PATTERN AS BACKGROUN
6388  041534  004737  020640              CALL    MT0027          ;CREATE & CHECK BACKGROUND
6389  041540  012737  000012  002012      MOV     #10.,TIME
6390  041546                              TYPE    MSG052          ;REMOVE SYSTEM POWER FOR
 (1)  041546  104401  071100               TYPEIT  ,MSG052
6391  041552                              TYPDEC  TIME
 (1)  041552  013746  002012              MOV     TIME,-(SP)      ;;SAVE TIME FOR TYPEOUT
 (1)  041556  104405                      TYPDS                   ;;GO TYPE--DECIMAL ASCII WITH SIGN
6392  041560                              TYPE    MSG053          ;SECONDS MAX!
 (1)  041560  104401  071131               TYPEIT  ,MSG053
6393  041564                              TYPE    MSG046          ;TO ESCAPE TYPE ANY KEY
 (1)  041564  104401  070606               TYPEIT  ,MSG046
6394  041570  012737  041632  000060      MOV     #CMD7A,TKVEC
6395  041576  012737  000340  000062      MOV     #340,TKVEC+2
6396  041604  017700  140376              MOV     @$TKB,R0        ;KILL ANY OLD INTERRUPT
6397  041610  042737  000200  177776      BIC     #BIT7,PSW       ;LOWER CPU PRIORITY TO 140
6398  041616  052777  000100  140360      BIS     #BIT6,@$TKS     ;ENABLE KEYBOARD INTERRUPTS
6399  041624  010637  001752              MOV     SP,FSSTACK
6400
6401  041630  000001                      WAIT
6402
6403  041632  013706  001752      CMD7A:  MOV     FSSTACK,SP      ;RECOVER OLD STACK POINTER
6404  041636  042777  000100  140340      BIC     #BIT6,@$TKS
6405  041644                              POP     TKVEC+2,TKVEC
 (2)  041644  012637  000062                                                                  MOV (SP)+,TKVEC+2
 (3)  041650  012637  000060                                                                  MOV (SP)+,TKVEC
6406  041654  117700  140326              MOVB    @$TKB,R0        ;GET CHARACTER TO GET RID OF FLAG
6407  041660                              TYPE    MSG054          ;NOW STARTING READ TEST OF MEMORY BANKS
 (1)  041660  104401  071147               TYPEIT  ,MSG054
6408  041664                              FOR BANK := #0 TO #167
 (4)  041664  005037  001470                                                                  CLR BANK
 (4)  041670                                                                        B55:;;;;;;;
6409  041670  004737  036052                CALL  EXBANK
```

```
 6410  041674                                  IF ACFLAG NE #0 AND RRFLAG EQ #0
  (4)  041674  005737  001506                                                                  TST ACFLAG
  (6)  041700  001415                                                                          BEQ L400
  (6)  041702  005737  001514                                                                  TST RRFLAG
  (8)  041706  001012                                                                          BNE L400
 6411  041710                                    LET R2 := BANK
  (3)  041710  013702  001470                                                                  MOV BANK,R2
 6412  041714  012700  060000                    MOV #FIRST,R0
 6413  041720  010004                            MOV R0,R4
 6414  041722  012701  040000                    MOV #SIZE,R1
 6415  041726  010103                            MOV R1,R3
 6416  041730  004737  021724                    CALL        SUPDO3
 6417  041734                                  END ;OF IF ACFLAG
  (3)  041734                                                                          L400:;;;;;;
 6418  041734                                END ;OF FOR BANK
  (6)  041734  005237  001470                                                                  INC BANK
  (5)  041740  023727  001470  000167                                                          CMP BANK,#167
  (5)  041746  003750                                                                          BLE B55
  (5)  041750                                                                          E55:;;;;;;;
 6419  041750                                TYPE    MSG047                  ;TEST COMPLETE
  (1)  041750  104401  070641                TYPEIT  ,MSG047
 6420  041754                                POP     CSRNO,MARGIN,PATTERN,BANK
  (2)  041754  012637  001634                                                                  MOV (SP)+,CSRNO
  (3)  041760  012637  001474                                                                  MOV (SP)+,MARGIN
  (4)  041764  012637  001502                                                                  MOV (SP)+,PATTERN
  (5)  041770  012637  001470                                                                  MOV (SP)+,BANK
 6421  041774  000207                        RETURN
 6422
```

```
6424  041776                          FSCMD8: SUBTST  <<FS    COMMAND 8       SOB-A-LONG TEST>>
 (3)                                  ;*****************************************************************************
 (3)                                  ;*SUBTEST       FS      COMMAND 8       SOB-A-LONG TEST
 (2)                                  ;*****************************************************************************
6425  041776                                  PUSH    BANK,PATTERN,MARGIN,TKVEC,TKVEC+2,NOPAR
 (2)  041776  013746  001470                                                            MOV BANK,-(SP)
 (3)  042002  013746  001502                                                            MOV PATTERN,-(SP)
 (4)  042006  013746  001474                                                            MOV MARGIN,-(SP)
 (5)  042012  013746  000060                                                            MOV TKVEC,-(SP)
 (6)  042016  013746  000062                                                            MOV TKVEC+2,-(SP)
 (7)  042022  013746  001464                                                            MOV NOPAR,-(SP)
6426  042026  010637  001752                  MOV     SP,FSSTACK              ;SAVE LAST GOOD STACK POINTER
6427  042032                                  TYPE    MSG055                  ;SOB-A-LONG TEST
 (1)  042032  104401  071217                  TYPEIT  ,MSG055
6428  042036                          CMD8A:  TYPE    MSG044                  ;MARGIN(0,2,3,4,5)?
 (1)  042036  104401  070526                  TYPEIT  ,MSG044
6429  042042  104413                          RDOCT                           ;READ AN OCTAL NUMBER ONTO THE STACK
6430  042044                                  POP     MARGIN                  ;PUT IN IN MARGIN
 (2)  042044  012637  001474                                                            MOV (SP)+,MARGIN
6431  042050                                  IF MARGIN GT #5 OR MARGIN EQ #1
 (5)  042050  023727  001474  000005                                                    CMP MARGIN,#5
 (5)  042056  003004                                                                    BGT L401
 (6)  042060  023727  001474  000001                                                    CMP MARGIN,#1
 (8)  042066  001001                                                                    BNE L402
 (4)  042070                                                                    L401:;;;;;;
6432  042070                                    GOTO CMD8A
 (2)  042070  000762                                                                    BR CMD8A
6433  042072                                  END ;OF IF
 (3)  042072                                                                    L402:;;;;;;
6434  042072                                  IF MARGIN NE #0 AND #SW11 SET.IN @SWR
 (4)  042072  005737  001474                                                            TST MARGIN
 (6)  042076  001407                                                                    BEQ L403
 (6)  042100  032777  004000  140072                                                    BIT #SW11,@SWR
 (8)  042106  001403                                                                    BEQ L403
6435  042110                                    TYPE  MSG045                  ;MARGINS INHIBITED BY SW5
 (1)  042110  104401  070553                  TYPEIT  ,MSG045
6436  042114                                    GOTO CMD8A
 (2)  042114  000750                                                                    BR CMD8A
6437  042116                                  END ;OF IF MARGIN
 (3)  042116                                                                    L403:;;;;;;
6438
6439  042116                                  IF #SW0 SET.IN @SWR
 (4)  042116  032777  000001  140054                                                    BIT #SW0,@SWR
 (6)  042124  001402                                                                    BEQ L404
6440  042126  104470                            ECCDIS
6441  042130                                  ELSE
 (3)  042130  000401                                                                    BR L405
 (3)  042132                                                                    L404:;;;;;;
6442  042132  104502                            CLRCSR
6443  042134                                  END ;OF IF
 (3)  042134                                                                    L405:;;;;;;
6444  042134                                  TYPE    MSG056                  ;BELL = EACH PASS COMPLETE
 (1)  042134  104401  071240                  TYPEIT  ,MSG056
6445
6446  042140                                  TYPE    MSG046                  ;TO ESCAPE TYPE ANY KEY!
 (1)  042140  104401  070606                  TYPEIT  ,MSG046
```

```
 6447  042144  012737  042300  000060          MOV     #CMD8C,TKVEC
 6448  042152  012737  000340  000062          MOV     #340,TKVEC+2
 6449  042160  017700  140022                  MOV     @$TKB,RC                ;KILL ANY OLD INTERRUPT
 6450  042164  042737  000200  177776          BIC     #BIT7,PSW               ;LOWER CPU PRIORITY TO 140
 6451  042172  052777  000100  140004          BIS     #BIT6,@$TKS             ;ENABLE KEYBOARD INTERRUPTS
 6452
 6453  042200  013700  001474                  MOV     MARGIN,RO               ;SET MARGINS
 6454  042204  006300                          ASL     RO                      ;SET MARGINS
 6455  042206  010037  177750                  MOV     RO,MAINT
 6456
 6457  042212                                  SET     HEADER
   (1) 042212  012737  177777  002146          MOV     #-1,HEADER
 6458  042220                                  SET     MUT
   (1) 042220  012737  177777  001500          MOV     #-1,MUT
 6459
 6460  042226                          CMD8B:  FOR BANK := #0 TO #167                          CLR BANK
   (4) 042226  005037  001470                                                            B56:;;;;;;;
   (4) 042232
 6461  042232  004737  036052                    CALL EXBANK
 6462  042236                                    IF ACFLAG NE #0 AND RRFLAG EQ #0              TST ACFLAG
   (4) 042236  005737  001506                                                              BEQ L406
   (6) 042242  001405                                                                      TST RRFLAG
   (6) 042244  005737  001514                                                              BNE L406
   (8) 042250  001002
 6463  042252  004737  021430                      CALL MT0031
 6464  042256                                    END ;OF IF ACFLAG
   (3) 042256                                                                            L406:;;;;;;
 6465  042256                                  END ;OF FOR BANK                            INC BANK
   (6) 042256  005237  001470                                                            CMP BANK,#167
   (5) 042262  023727  001470  000167                                                    BLE B56
   (5) 042270  003760
   (5) 042272                                                                            E56:;;;;;;;
 6466  042272                                  TYPE    $BELL                   ;RING BELL
   (1) 042272  104401  002215                  TYPEIT  $BELL
 6467  042276                                  GOTO    CMD8B
   (2) 042276  000753                                                                      BR CMD8B
 6468
 6469  042300  013706  001752          CMD8C:  MOV     FSSTACK,SP              ;RECOVER OLD STACK POINTER
 6470  042304  042777  000100  137672          BIC     #BIT6,@$TKS
 6471  042312  117700  137670                  MOVB    @$TKB,RO                ;READ CHAR TO KILL FLAG
 6472  042316                                  POP     NOPAR,TKVEC+2,TKVEC,MARGIN,PATTERN,BANK
   (2) 042316  012637  001464                                                            MOV (SP)+,NOPAR
   (3) 042322  012637  000062                                                            MOV (SP)+,TKVEC+2
   (4) 042326  012637  000060                                                            MOV (SP)+,TKVEC
   (5) 042332  012637  001474                                                            MOV (SP)+,MARGIN
   (6) 042336  012637  001502                                                            MOV (SP)+,PATTERN
   (7) 042342  012637  001470                                                            MOV (SP)+,BANK
 6473  042346                                  MAP     BANK
   (3) 042346  010346                                                                    MOV R3,-(SP)
   (1) 042350  013703  001470                  MOV     BANK,R3
   (1) 042354  004737  034510                  CALL    MAPPER
   (3) 042360  012603                                                                    MOV (SP)+,R3
 6474  042362  004737  036052                  CALL    EXBANK
 6475  042366  000207                          RETURN
 6476
 6477  042370                          FSCMD9: SUBTST  <<FS    COMMAND 9       SUPER TIGHT SCOPE LOOP>>
```

```
  (3)                                   ;*******************************************************************************
  (3)                                   ;*SUBTEST        FS        COMMAND 9        SUPER TIGHT SCOPE LOOP
  (2)                                   ;*******************************************************************************
 6478  042370                                   PUSH     BANK,MARGIN,TKVEC,TKVEC+2,NOPAR
  (2)  042370  013746  001470                                                                          MOV BANK,-(SP)
  (3)  042374  013746  001474                                                                          MOV MARGIN,-(SP)
  (4)  042400  013746  000060                                                                          MOV TKVEC,-(SP)
  (5)  042404  013746  000062                                                                          MOV TKVEC+2,-(SP)
  (6)  042410  013746  001464                                                                          MOV NOPAR,-(SP)
 6479  042414  010637  001752                   MOV      SP,FSSTACK                ;SAVE LAST GOOD STACK
 6480  042420                                   TYPE     MSG059                    ;SUPER TIGHT SCOPE LOOP
  (1)  042420  104401  071277                   TYPEIT   ,MSG059
 6481  042424                                   IF #SW0 SET.IN @SWR
  (4)  042424  032777  000001  137546                                                                  BIT #SW0,@SWR
  (6)  042432  001402                                                                                  BEQ L407
 6482  042434  104470                             ECCDIS
 6483  042436                                   ELSE
  (3)  042436  000401                                                                                  BR L410
  (3)  042440                                                                                  L407:;;;;;;
 6484  042440  104502                             CLRCSR
 6485  042442                                   END ;OF IF
  (3)  042442                                                                                  L410:;;;;;;
 6486  042442                           1$:     TYPE     MSG031            ;PHYSICAL ADDRESS (0-16777776)??
  (1)  042442  104401  070054                   TYPEIT   ,MSG031
 6487  042446  104413                           RDOCT                      ;READ OCTAL NUMBER ONTO STACK & $HIOCT
 6488  042450  013737  057464  001470           MOV      $HIOCT,BANK       ;PUT MSB'S IN BANK
 6489  042456                                   POP      R0                ;PUT LSB'S IN R0
  (2)  042456  012600                                                                                  MOV (SP)+,R0
 6490  042460  000241                           CLC
 6491  042462  006100                           ROL      R0
 6492  042464  006137  001470                   ROL      BANK
 6493  042470  000241                           CLC
 6494  042472  006000                           ROR      R0
 6495  042474                                   IF BANK GT #167 THEN GOTO 1$     ;CHECK FOR BANK TOO HIGH
  (5)  042474  023727  001470  000167                                                                  CMP BANK,#167
  (5)  042502  003357                                                                                  BGT 1$
 6496  042504  062700  060000                   ADD      #FIRST,R0
 6497  042510                                   IF #BIT0 SET.IN R0 THEN GOTO 1$ ;CHECK FOR ODD ADDRESS
  (5)  042510  032700  000001                                                                          BIT #BIT0,R0
  (5)  042514  001352                                                                                  BNE 1$
 6498  042516                                   IF R0 HI #LAST THEN GOTO 1$      ;CHECK FOR ADDRESS OVER 16K
  (5)  042516  020027  157776                                                                          CMP R0,#LAST
  (5)  042522  101347                                                                                  BHI 1$
 6499
 6500  042524                           CMD9A:  TYPE     MSG044                    ;MARGIN(0,2,3,4,5)?
  (1)  042524  104401  070526                   TYPEIT   ,MSG044
 6501  042530  104413                           RDOCT                              ;READ AN OCTAL NUMBER ONTO THE STACK
 6502  042532                                   POP      MARGIN                    ;PUT IN IN MARGIN
  (2)  042532  012637  001474                                                                          MOV (SP)+,MARGIN
 6503  042536                                   IF MARGIN GT #5 OR MARGIN EQ #1
  (5)  042536  023727  001474  000005                                                                  CMP MARGIN,#5
  (5)  042544  003004                                                                                  BGT L411
  (6)  042546  023727  001474  000001                                                                  CMP MARGIN,#1
  (8)  042554  001001                                                                                  BNE L412
  (4)  042556                                                                                  L411:;;;;;;
 6504  042556                                     GOTO CMD9A
```

```
 (2)  042556  000762                                                                  BR CMD9A
6505  042560                                  END ;OF IF
 (3)  042560                                                                  L412:;;;;;;
6506  042560                                  IF MARGIN NE #0 AND #SW11 SET.IN @SWR
 (4)  042560  005737  001474                                                          TST MARGIN
 (6)  042564  001407                                                                  BEQ L413
 (6)  042566  032777  004000  137404                                                  BIT #SW11,@SWR
 (8)  042574  001403                                                                  BEQ L413
6507  042576                                    TYPE  MSG045                 ;MARGINS INHIBITED BY SW5
 (1)  042576  104401  070553                  TYPEIT  ,MSG045
6508  042602                                    GOTO CMD9A
 (2)  042602  000750                                                                  BR CMD9A
6509  042604                                  END ;OF IF MARGIN
 (3)  042604                                                                  L413:;;;;;;
6510
6511  042604  012737  000003  001464          MOV     #3,NOPAR                ;PARITY ACTION
6512  042612  012737  042726  001750          MOV     #CMD9LOOP,PARTHERE
6513  042620                                  SET4    #CMD9B
 (1)  042620  012737  042740  000004          MOV     #CMD9B,4                        ;SET NEW VECTOR
6514  042626                                  MAP     BANK
 (3)  042626  010346                                                                  MOV R3,-(SP)
 (1)  042630  013703  001470                  MOV     BANK,R3
 (1)  042634  004737  034510                  CALL    MAPPER
 (3)  042640  012603                                                                  MOV (SP)+,R3
6515  042642                                  TYPE    MSG060                  ;SWR ---> MEM, MEM ---> DISPLAY
 (1)  042642  104401  071327                  TYPEIT  ,MSG060
6516
6517  042646                                  TYPE    MSG046                  ;TO ESCAPE TYPE ANY KEY!
 (1)  042646  104401  070606                  TYPEIT  ,MSG046
6518  042652  012737  042746  000060          MOV     #CMD9C,TKVEC
6519  042660  012737  000340  000062          MOV     #340,TKVEC+2
6520  042666  017701  137314                  MOV     @$TKB,R1                ;KILL ANY OLD INTERRUPT
6521  042672  042737  000200  177776          BIC     #BIT7,PSW               ;LOWER CPU PRIORITY TO 140
6522  042700  052777  000100  137276          BIS     #BIT6,@$TKS             ;ENABLE KEYBOARD INTERRUPTS
6523
6524  042706  013701  001474                  MOV     MARGIN,R1               ;SET MARGINS
6525  042712  006301                          ASL     R1                      ;SET MARGINS
6526  042714  010137  177750                  MOV     R1,MAINT
6527
6528  042720                                  SUPERVISOR
 (1)  042720  052737  040000  177776          BIS     #BIT14,PSW              ;DO IT TO IT!
6529  042726  017710  137246          CMD9LOOP:MOV    @SWR,(R0)               ;WRITE SWITCH REGISTER INTO MEMORY
6530  042732  011077  137244                  MOV     (R0),@DISPLAY           ;READ MEMORY INTO DISPLAY REGISTER
6531  042736  000773                          BR      CMD9LOOP                ;LOOP TILL KEYBOARD INTERRUPT
6532
6533  042740  012716  042726          CMD9B:  MOV     #CMD9LOOP,(SP)
6534  042744  000002                          RTI
6535
6536  042746  013706  001752          CMD9C:  MOV     FSSTACK,SP                      ;RECOVER OLD STACK POINTER
6537  042752  042777  000100  137224          BIC     #BIT6,@$TKS
6538  042760  117700  137222                  MOVB    @$TKB,R0                        ;READ CHAR TO KILL FLAG
6539  042764                                  POP     NOPAR,TKVEC+2,TKVEC,MARGIN,BANK
 (2)  042764  012637  001464                                                          MOV (SP)+,NOPAR
 (3)  042770  012637  000062                                                          MOV (SP)+,TKVEC+2
 (4)  042774  012637  000060                                                          MOV (SP)+,TKVEC
 (5)  043000  012637  001474                                                          MOV (SP)+,MARGIN
```

```
  (6)  043004  012637  001470                                                                  MOV (SP)+,BANK
 6540  043010                                  MAP     BANK
  (3)  043010  010346                                                                          MOV R3,-(SP)
  (1)  043012  013703  001470                  MOV     BANK,R3
  (1)  043016  004737  034510                  CALL    MAPPER
  (3)  043022  012603                                                                          MOV (SP)+,R3
 6541  043024  004737  036052                  CALL    EXBANK
 6542  043030  000207                          RETURN
 6543
 6544  043032                          FCMD10: SUBTST  <<FS    COMMAND 10      ERROR SUMMARY>>
  (3)                                  ;*****************************************************************************
  (3)                                  ;*SUBTEST        FS      COMMAND 10      ERROR SUMMARY
  (2)                                  ;*****************************************************************************
 6545  043032                                  PUSH    R0,R2,R3,BANK
  (2)  043032  010046                                                                          MOV R0,-(SP)
  (3)  043034  010246                                                                          MOV R2,-(SP)
  (4)  043036  010346                                                                          MOV R3,-(SP)
  (5)  043040  013746  001470                                                                  MOV BANK,-(SP)
 6546  043044                                  TYPDEC  $ERTTL
  (1)  043044  013746  001406                  MOV     $ERTTL,-(SP)    ;;SAVE $ERTTL FOR TYPEOUT
  (1)  043050  104405                          TYPDS                   ;;GO TYPE--DECIMAL ASCII WITH SIGN
 6547  043052                                  TYPE    MSG079                  ;ERROR(S) DETECTED
  (1)  043052  104401  072002                  TYPEIT  ,MSG079
 6548  043056                                  IF $ERTTL NE #0
  (4)  043056  005737  001406                                                                  TST $ERTTL
  (6)  043062  001437                                                                          BEQ L414
 6549  043064                                    TYPE  MSG076                  ;BANK  ERRORS
  (1)  043064  104401  071702                  TYPEIT  ,MSG076
 6550  043070                                    FOR BANK := #0 TO #167
  (4)  043070  005037  001470                                                                  CLR BANK
  (4)  043074                                                                          B57:;;;;;;
 6551  043074  013703  001470                      MOV BANK,R3
 6552  043100  070327  000006                      MUL #6,R3
 6553  043104  062703  000005                      ADD #5,R3
 6554  043110                                      IFB CONFIG(R3) NE #0
  (5)  043110  105763  002230                                                                  TSTB CONFIG(R3)
  (7)  043114  001414                                                                          BEQ L415
 6555  043116                                        TYPOCS    BANK,3
  (1)  043116  013746  001470                  MOV     BANK,-(SP)      ;;SAVE BANK FOR TYPEOUT
  (1)                                                                  ;;3
  (1)  043122  104403                          TYPOS                   ;;GO TYPE--OCTAL ASCII
  (1)  043124     006                          .BYTE   6               ;;TYPE 6 DIGITS
  (1)  043125     000                          .BYTE   0               ;;SUPPRESS LEADING ZEROS
 6556  043126  116300  002230                        MOVB      CONFIG(R3),R0
 6557  043132  042700  177400                        BIC       #↑C377,R0
 6558  043136                                        TYPDEC    R0
  (1)  043136  010046                          MOV     R0,-(SP)        ;;SAVE R0 FOR TYPEOUT
  (1)  043140  104405                          TYPDS                   ;;GO TYPE--DECIMAL ASCII WITH SIGN
 6559  043142                                        TYPE      $CRLF
  (1)  043142  104401  002222                  TYPEIT  ,$CRLF
 6560  043146                                      END ;OF IFB CONFIG(R3)
  (3)  043146                                                                          L415:;;;;;;
 6561  043146                                    END ;OF FOR BANK
  (6)  043146  005237  001470                                                                  INC BANK
  (5)  043152  023727  001470  000167                                                          CMP BANK,#167
  (5)  043160  003745                                                                          BLE B57
```

```
  (5)  043162
6562   043162                              END :OF IF $ERTTL
  (3)  043162
6563   043162                              POP     BANK,R3,R2,R0
  (2)  043162  012637  001470
  (3)  043166  012603
  (4)  043170  012602
  (5)  043172  012600
6564   043174  000207                      RETURN
```

```
E57:::::::

L414:::::::
        MOV (SP)+,BANK
        MOV (SP)+,R3
        MOV (SP)+,R2
        MOV (SP)+,R0
```

```
6566  043176                        FCMD11: SUBTST  <<FS    COMMAND 11      REFRESH TEST>>
 (3)                                ;*****************************************************************************
 (3)                                ;*SUBTEST       FS      COMMAND 11      REFRESH TEST
 (2)                                ;*****************************************************************************
6567  043176                                PUSH    BANK,PATTERN,MARGIN,TKVEC,TKVEC+2,NOPAR
 (2)  043176  013746  001470                                                            MOV BANK,-(SP)
 (3)  043202  013746  001502                                                            MOV PATTERN,-(SP)
 (4)  043206  013746  001474                                                            MOV MARGIN,-(SP)
 (5)  043212  013746  000060                                                            MOV TKVEC,-(SP)
 (6)  043216  013746  000062                                                            MOV TKVEC+2,-(SP)
 (7)  043222  013746  001464                                                            MOV NOPAR,-(SP)
6568  043226  010637  001752                MOV     SP,FSSTACK              ;SAVE LAST GOOD STACK POINTER
6569  043232                              TYPE    MSG073                  ;REFRESH TEST
 (1)  043232  104401  071630              TYPEIT  ,MSG073
6570  043236                        CMD11A: TYPE    MSG044                  ;MARGIN(0,2,3,4,5)?
 (1)  043236  104401  070526              TYPEIT  ,MSG044
6571  043242  104413                      RDOCT                           ;READ AN OCTAL NUMBER ONTO THE STACK
6572  043244                              POP     MARGIN                  ;PUT IN IN MARGIN
 (2)  043244  012637  001474                                                            MOV (SP)+,MARGIN
6573  043250                              IF MARGIN GT #5 OR MARGIN EQ #1
 (5)  043250  023727  001474  000005                                                    CMP MARGIN,#5
 (5)  043256  003004                                                                    BGT L416
 (6)  043260  023727  001474  000001                                                    CMP MARGIN,#1
 (8)  043266  001001                                                                    BNE L417
 (4)  043270                                                                    L416:;;;;;;
6574  043270                                GOTO CMD11A
 (2)  043270  000762                                                                    BR CMD11A
6575  043272                              END ;OF IF
 (3)  043272                                                                    L417:;;;;;;
6576  043272                              IF MARGIN NE #0 AND #SW11 SET.IN @SWR
 (4)  043272  005737  001474                                                            TST MARGIN
 (6)  043276  001407                                                                    BEQ L420
 (6)  043300  032777  004000  136672                                                    BIT #SW11,@SWR
 (8)  043306  001403                                                                    BEQ L420
6577  043310                                TYPE  MSG045                    ;MARGINS INHIBITED BY SW5
 (1)  043310  104401  070553              TYPEIT  ,MSG045
6578  043314                                GOTO CMD11A
 (2)  043314  000750                                                                    BR CMD11A
6579  043316                              END ;OF IF MARGIN
 (3)  043316                                                                    L420:;;;;;;
6580
6581  043316                              IF #SWD SET.IN @SWR
 (4)  043316  032777  000001  136654                                                    BIT #SWC,@SWR
 (6)  043324  001402                                                                    BEQ L421
6582  043326  104470                        ECCDIS
6583  043330                              ELSE
 (3)  043330  000401                                                                    BR L422
 (3)  043332                                                                    L421:;;;;;;
6584  043332  104502                        CLRCSR
6585  043334                              END ;OF IF
 (3)  043334                                                                    L422:;;;;;;
6586  043334                              TYPE    MSG056                  ;BELL = EACH PASS COMPLETE
 (1)  043334  104401  071240              TYPEIT  ,MSG056
6587
6588  043340                              TYPE    MSG046                  ;TO ESCAPE TYPE ANY KEY!
 (1)  043340  104401  070606              TYPEIT  ,MSG046
```

```
6589  043344  012737  043500  000060          MOV     #CMD11C,TKVEC
6590  043352  012737  000340  000062          MOV     #340,TKVEC+2
6591  043360  017700  136622                  MOV     @$TKB,RC                ;KILL ANY OLD INTERRUPT
6592  043364  042737  000200  177776          BIC     #BIT7,PSW               ;LOWER CPU PRIORITY TO 140
6593  043372  052777  000100  136604          BIS     #BIT6,@$TKS             ;ENABLE KEYBOARD INTERRUPTS
6594
6595  043400  013700  001474                  MOV     MARGIN,R0               ;SET MARGINS
6596  043404  006300                          ASL     R0                      ;SET MARGINS
6597  043406  010037  177750                  MOV     R0,MAINT
6598
6599  043412                                  SET     HEADER
  (1) 043412  012737  177777  002146          MOV     #-1,HEADER
6600  043420                                  SET     MUT
  (1) 043420  012737  177777  001500          MOV     #-1,MUT
6601
6602  043426                          CMD11B: FOR BANK := #0 TO #167                          CLR BANK
  (4) 043426  005037  001470                                                        B60:::::::
  (4) 043432
6603  043432  004737  036052                    CALL EXBANK
6604  043436                                    IF ACFLAG NE #0 AND RRFLAG EQ #0              TST ACFLAG
  (4) 043436  005737  001506                                                                BEQ L423
  (6) 043442  001405                                                                        TST RRFLAG
  (6) 043444  005737  001514                                                                BNE L423
  (8) 043450  001002
6605  043452  004737  020334                      CALL MT0022
6606  043456                                    END ;OF IF ACFLAG
  (3) 043456                                                                        L423:::::::
6607  043456                                  END ;OF FOR BANK                              INC BANK
  (6) 043456  005237  001470                                                                CMP BANK,#167
  (5) 043462  023727  001470  000167                                                        BLE B60
  (5) 043470  003760                                                                E60:::::::::
  (5) 043472
6608  043472                                  TYPE    $BELL                   ;RING BELL
  (1) 043472  104401  002215                  TYPEIT  ,$BELL
6609  043476                                  GOTO    CMD11B                                BR CMD11B
  (2) 043476  000753
6610
6611  043500  013706  001752          CMD11C: MOV     FSSTACK,SP                      ;RECOVER OLD STACK POINTER
6612  043504  042777  000100  136472          BIC     #BIT6,@$TKS
6613  043512  117700  136470                  MOVB    @$TKB,R0                ;READ CHAR TO KILL FLAG
6614  043516                                  POP     NOPAR,TKVEC+2,TKVEC,MARGIN,PATTERN,BANK
  (2) 043516  012637  001464                                                                MOV (SP)+,NOPAR
  (3) 043522  012637  000062                                                                MOV (SP)+,TKVEC+2
  (4) 043526  012637  000060                                                                MOV (SP)+,TKVEC
  (5) 043532  012637  001474                                                                MOV (SP)+,MARGIN
  (6) 043536  012637  001502                                                                MOV (SP)+,PATTERN
  (7) 043542  012637  001470                                                                MOV (SP)+,BANK
6615  043546                                  MAP     BANK
  (3) 043546  010346                                                                        MOV R3,-(SP)
  (1) 043550  013703  001470                  MOV     BANK,R3
  (1) 043554  004737  034510                  CALL    MAPPER
  (3) 043560  012603                                                                        MOV (SP)+,R3
6616  043562  004737  036052                  CALL    EXBANK
6617  043566  000207                          RETURN
6618
```

```
FIELD SERVICE MODE      MACY11 30A(1052)  28-FEB-78  09:20  PAGE 52-10
CEMKAA.MA1    24-FEB-78 11:01           FS      COMMAND 11      REFRESH TEST                                                    SEQ 0317

  6620  043570                          FCMD12: SUBTST  <<FS     COMMAND 12      SET FILL COUNT>>
    (3)                                 ;*********************************************************************************
    (3)                                 ;*SUBTEST        FS      COMMAND 12      SET FILL COUNT
    (2)                                 ;*********************************************************************************
  6621  043570                                  PUSH    R0
    (2) 043570  010046                                                                          MOV R0,-(SP)
  6622  043572                                  TYPE    MSG085          ;FILL COUNT(OCTAL)?
    (1) 043572  104401  072161                  TYPEIT  .MSG085
  6623  043576  104413                          RDOCT
  6624  043600                                  POP     R0
    (2) 043600  012600                                                                          MOV (SP)+,R0
  6625  043602  042700  177760                  BIC     #^C17,R0
  6626  043606  110037  002027                  MOVB    R0,$FILLS
  6627  043612                                  POP     R0
    (2) 043612  012600                                                                          MOV (SP)+,R0
  6628  043614  000207                          RETURN
```

```
 6630  043616                          WHICHCSR:SUBTST <<SUBR  DETERMINE CORRECT CSR>>
  (3)                                  ;*****************************************************************************
  (3)                                  ;*SUBTEST       SUBR    DETERMINE CORRECT CSR
  (2)                                  ;*****************************************************************************
 6631  043616  013700  001706                  MOV     MKCSRS,R0       ;GET CSR'S FLAG
 6632  043622  122700  000200                  CMPB    #BIT7,R0        ;CSR 0?
 6633  043626  001003                          BNE     1$              ;NO - SKIP
 6634  043630  005037  001634                  CLR     CSRNO           ;YES - SET IT UP
 6635  043634  000207                          RETURN
 6636
 6637  043636                          1$:     TYPE    MSG022          ;WHICH CSR(0-7)
  (1)  043636  104401  067633                  TYPEIT  ,MSG022
 6638  043642  104413                          RDOCT                   ;GET OCTAL NUMBER
 6639  043644                                  POP     R0              ;PUT IN R0
  (2)  043644  012600                                                                  MOV (SP)+,R0
 6640  043646  020027  000007                  CMP     R0,#7           ;CHECK LIMIT
 6641  043652  101371                          BHI     1$              ;IF BAD LOOP TILL HE TYPES IT RIGHT
 6642  043654  012701  177774                  MOV     #-4,R1
 6643  043660  005200                          INC     R0
 6644  043662  062701  000004          2$:     ADD     #4,R1
 6645  043666  077003                          SOB     R0,2$
 6646  043670  010137  001634                  MOV     R1,CSRNO
 6647  043674  000207                          RETURN
 6648
```

```
                        .TITLE  ERROR REPORTING & SCOPE LOOPS

                        SUBTST  <<SUBR  ERROR DATA SETUP>>
;*******************************************************************************
;*SUBTEST        SUBR    ERROR DATA SETUP
;*******************************************************************************

;  USE THIS       IF THIS CONDITION DISCRIBES THE ERROR

;  PERR01         TRAP
;                 BAD DATA IN R0 UNLESS ABORTED
;                         THEN BAD DATA IS POINTED TO BY -(R4)
;                 GOOD DATA IN R5

;  PERR02         TRAP
;                 BAD DATA IN R1 UNLESS ABORTED
;                         THEN BAD DATA IS POINTED TO BY -(R4)
;                 GOOD DATA IN R2

;  PERR03         TRAP
;                 BAD DATA IS POINTED TO BY -(R1)
;                 GOOD DATA IN R4

;  PERR04         TRAP
;                 BAD DATA IN R4 UNLESS ABORTED
;                         THEN BAD DATA IS POINTED TO BY -2(R0)
;                 GOOD DATA IN R2

;  PERR05         JSR     PC
;                 BAD DATA IS POINTED TO BY -(R0)
;                 GOOD DATA IN R2
;                 RETURN AFTER SETTING UP GOOD,BAD,ADDRESS

;  PERR06         JSR     PC
;                 BAD DATA IS POINTED TO BY -(R0)
;                 GOOD DATA IS ZERO
;                 RETURN AFTER SETTING UP GOOD,BAD,ADDRESS

;  PERR07         TRAP
;                 BAD DATA IN R2 UNLESS ABORTED
;                         THEN BAD DATA IS POINTED TO BY (R1)
;                 GOOD DATA IN DATBUF

;  PERR10         TRAP
;                 BAD DATA IN R2 UNLESS ABORTED
;                         THEN BAD DATA IS POINTED TO BY 2(R1)
;                 GOOD DATA IN DATBUF+2

;  PERR11         TRAP
;                 BYTE TEST
;                 BAD DATA IN RIGHT BYTE OF R0 UNLESS ABORTED
;                         THEN BAD DATA IS POINTED TO BY (R1)
;                 GOOD DATA IS A ZERO BYTE

;  PERR12         TRAP
;                 BYTE TEST
```

```
                        ;               BAD DATA IN RIGHT BYTE OF R0 UNLESS ABORTED
                        ;                       THEN BAD DATA IS POINTED TO BY (R1)
                        ;               GOOD DATA IS A BYTE OF ONES
                        ;
                        ; PERR13       TRAP
                        ;               BAD DATA IN R0 UNLESS ABORTED
                        ;                       THEN BAD DATA IS POINTED TO BY (R1)
                        ;               GOOD DATA IS ZERO
                        ;
                        ; PERR14       TRAP
                        ;               BAD DATA IN R0 UNLESS ABORTED
                        ;                       THEN BAD DATA IS POINTED TO BY (R1)
                        ;               GOOD DATA IS ONES
                        ;
                        ; PERR15       TRAP
                        ;               BAD DATA IN R0 UNLESS ABORTED
                        ;                       THEN BAD DATA IS POINTED TO BY (R1)
                        ;               GOOD DATA IN TSTDAT
                        ;
                        ; PERR16       TRAP
                        ;               BAD DATA IN R0 UNLESS ABORTED
                        ;                       THEN BAD DATA IS POINTED TO BY (R1)
                        ;               GOOD DATA IN TSTDAT+2
                        ;
                        ; PERR17       TRAP
                        ;               BAD DATA IN R0 UNLESS ABORTED
                        ;                       THEN BAD DATA IS POINTED TO BY (R1)
                        ;               GOOD DATA IN R2
                        ;
                        ; PERR20       TRAP
                        ;               BAD DATA IN R0 UNLESS ABORTED
                        ;                       THEN BAD DATA IS POINTED TO BY (R1)
                        ;               GOOD DATA IN R3
                        ;
                        ; PERR21       TRAP
                        ;               7 BIT BYTE TEST
                        ;               BAD DATA IN RIGHT BYTE OF R0 UNLESS ABORTED
                        ;                       THEN BAD DATA IS POINTED TO BY (R1)
                        ;               GOOD DATA IS A 7 BIT BYTE ON ONES
                        ;
                        ; PERR22       TRAP
                        ;               BAD DATA IN R2 UNLESS ABORTED
                        ;                       THEN BAD DATA IS POINTED TO BY (R1)
                        ;               GOOD DATA IN R0
                        ;
                        ; PERR23       TRAP
                        ;               BAD DATA IN R0 UNLESS ABORTED
                        ;                       THEN BAD DATA IS POINTED TO BY (R1)
                        ;               GOOD DATA IN BAKPAT
                        ;
                        ; PERR24       TRAP
                        ;               BAD DATA IN R0 UNLESS ABORTED
                        ;                       THEN BAD DATA IS POINTED TO BY (R5)
                        ;               GOOD DATA IN BAKPAT
                        ;
                        ; PERR25       TRAP
```

```
7171                            ;               BAD DATA POINTED TO BY -(R1)
7172                            ;               GOOD DATA IN R2 UNLESS LOC V177654 IS SET
7173                            ;                       THEN GOOD DATA IS IN R3
7174                            ;
7175                            ;  PERR26       TRAP
7176                            ;               BAD DATA IS DOUBLE WORD POINTED TO BY R1 AND IN LOW 7 BITS OF R0
7177                            ;               GOOD DATA IS 000000,,100000,,100
7178                            ;
7179                            ;  PERR27       TRAP
7180                            ;               BAD DATA IS DOUBLE WORD POINTED TO BY R1 AND IN LOW 7 BITS OF R0
7181                            ;               GOOD DATA IS 000000,,000000,,077
7182                            ;
7183                            ;  PERR30       TRAP
7184                            ;               BAD DATA IS POINTED TO BY -16(SP)
7185                            ;               GOOD DATA IS IN R1
7186                            ;
7187                            ;  PERR31       TRAP
7188                            ;               SPECIAL ECC FAILURE HANDLER
7189                            ;
7190                            ;  PERR32       TRAP
7191                            ;               SPECIAL ECC FAILURE HANDLER
7192                            ;
7193                            ;  PERR33       TRAP
7194                            ;               SPECIAL ECC FAILURE HANDLER
7195                            ;
7196                            ;       CALLING SEQUENCE FOR TRAP TYPES
7197                            ;       BEQ     2$              ;NO - ERROR,BRANCH FOR CARD
7198                            ;       PERRXX                  ;TRAP TO ERROR ROUTINE
7199                            ;2$:    NEXT    INSTRUCTION     ;CONTINUE TESTING
```

```
 7202  046646  010437  001422          $PER01: MOV     R4,ADDRESS
 7203  046652  162737  000002  001422          SUB     #2,ADDRESS
 7204  046660  010037  001440                  MOV     R0,BAD
 7205  046664  010537  001432                  MOV     R5,GOOD
 7206  046670                                  JUMPTO PERRAW                   JMP PERRAW
   (2) 046670  000137  047450
 7207
 7208  046674  010437  001422          $PER02: MOV     R4,ADDRESS
 7209  046700  162737  000002  001422          SUB     #2,ADDRESS
 7210  046706  010137  001440                  MOV     R1,BAD
 7211  046712  010237  001432                  MOV     R2,GOOD
 7212  046716                                  JUMPTO PERRAW                   JMP PERRAW
   (2) 046716  000137  047450
 7213
 7214  046722  010137  001422          $PER03: MOV     R1,ADDRESS
 7215  046726  162737  000002  001422          SUB     #2,ADDRESS
 7216  046734  010437  001432                  MOV     R4,GOOD
 7217  046740  016137  177776  001440          MOV     -2(R1),BAD
 7218  046746                                  JUMPTO PERRAW                   JMP PERRAW
   (2) 046746  000137  047450
 7219
 7220  046752  010037  001422          $PER04: MOV     R0,ADDRESS
 7221  046756  162737  000002  001422          SUB     #2,ADDRESS
 7222  046764  010437  001440                  MOV     R4,BAD
 7223  046770  010237  001432                  MOV     R2,GOOD
 7224  046774  000137  047450                  JMP     PERRAW
 7225
 7226  047000  010237  001432          PERR05: MOV     R2,GOOD
 7227  047004  014037  001440          PERA05: MOV     -(R0),BAD
 7228  047010  010037  001422                  MOV     R0,ADDRESS
 7229  047014  062700  000002                  ADD     #2,R0           ;RESTORE R0
 7230  047020  011637  001420                  MOV     (SP),BADPC
 7231  047024  000207                          RETURN
 7232
 7233  047026  005037  001432          PERR06: CLR     GOOD
 7234  047032  000764                          BR      PERA05
 7235
 7236  047034  010137  001422          $PER07: MOV     R1,ADDRESS
 7237  047040  010237  001440                  MOV     R2,BAD
 7238  047044  013737  001712  001432          MOV     DATBUF,GOOD
 7239  047052  000576                          BR      PERRAW
 7240
 7241  047054                          $PER10: LET ADDRESS := R1 + #2
   (3) 047054  010137  001422                                          MOV R1,ADDRESS
   (5) 047060  062737  000002  001422                                  ADD #2,ADDRESS
 7242  047066                                  LET BAD := R2
   (3) 047066  010237  001440                                          MOV R2,BAD
 7243  047072                                  LET GOOD := DATBUF+2
   (3) 047072  013737  001714  001432                                  MOV DATBUF+2,GOOD
 7244  047100  000563                          BR      PERRAW
 7245
 7246  047102                          $PER11: LET ADDRESS := R1
   (3) 047102  010137  001422                                          MOV R1,ADDRESS
 7247  047106                                  LET BAD := R0
   (3) 047106  010037  001440                                          MOV R0,BAD
 7248  047112                                  LET GOOD := #0
```

```
  (3)  047112  005037  001432                                                  CLR GOOD
 7249  047116  000577                          BR      PERRAB
 7250
 7251  047120                          $PER12: LET ADDRESS := R1                 MOV R1,ADDRESS
  (3)  047120  010137  001422                  LET BAD := R0                     MOV R0,BAD
 7252  047124
  (3)  047124  010037  001440                  LET GOOD := #377                  MOV #377,GOOD
 7253  047130
  (3)  047130  012737  000377  001432          BR      PERRAB
 7254  047136  000567
```

```
 7256  047140
  (3)  047140  010137  001422                $PER13: LET ADDRESS := R1          MOV R1,ADDRESS
 7257  047144
  (3)  047144  010037  001440                        LET BAD := R0              MOV R0,BAD
 7258  047150
  (3)  047150  005037  001432                        LET GOOD := #0             CLR GOOD
 7259  047154  000535                              BR      PERRAW
 7260
 7261  047156                               $PER14: LET ADDRESS := R1
  (3)  047156  010137  001422                                                    MOV R1,ADDRESS
 7262  047162
  (3)  047162  010037  001440                        LET BAD := R0              MOV R0,BAD
 7263  047166
  (3)  047166  013737  002154  001432               LET GOOD := ONES           MOV ONES,GOOD
 7264  047174  000525                              BR      PERRAW
 7265
 7266  047176                               $PER15: LET ADDRESS := R1
  (3)  047176  010137  001422                                                    MOV R1,ADDRESS
 7267  047202
  (3)  047202  010037  001440                        LET BAD := R0              MOV R0,BAD
 7268  047206
  (3)  047206  013737  001716  001432               LET GOOD := TSTDAT         MOV TSTDAT,GOOD
 7269  047214  000515                              BR      PERRAW
 7270
 7271  047216                               $PER16: LET ADDRESS := R1
  (3)  047216  010137  001422                                                    MOV R1,ADDRESS
 7272  047222
  (3)  047222  010037  001440                        LET BAD := R0              MOV R0,BAD
 7273  047226
  (3)  047226  013737  001720  001432               LET GOOD := TSTDAT+2       MOV TSTDAT+2,GOOD
 7274  047234  000505                              BR      PERRAW
 7275
 7276  047236                               $PER17: LET ADDRESS := R1
  (3)  047236  010137  001422                                                    MOV R1,ADDRESS
 7277  047242
  (3)  047242  010037  001440                        LET BAD := R0              MOV R0,BAD
 7278  047246
  (3)  047246  010237  001432                        LET GOOD := R2             MOV R2,GOOD
 7279  047252  000476                              BR      PERRAW
 7280
 7281  047254                               $PER20: LET ADDRESS := R1
  (3)  047254  010137  001422                                                    MOV R1,ADDRESS
 7282  047260
  (3)  047260  010037  001440                        LET BAD := R0              MOV R0,BAD
 7283  047264
  (3)  047264  010337  001432                        LET GOOD := R3             MOV R3,GOOD
 7284  047270  000467                              BR      PERRAW
 7285
 7286  047272                               $PER21: LET ADDRESS := R1
  (3)  047272  010137  001422                                                    MOV R1,ADDRESS
 7287  047276
  (3)  047276  010037  001440                        LET BAD := R0              MOV R0,BAD
 7288  047302
  (3)  047302  012737  000177  001432               LET GOOD := #177           MOV #177,GOOD
 7289  047310  000525                              BR      PERRA7
 7290
```

```
 7291  047312
  (3)  047312  010137  001422          $PER22: LET ADDRESS := R1          MOV R1,ADDRESS
 7292  047316                                  LET BAD := R2               MOV R2,BAD
  (3)  047316  010237  001440                  LET GOOD := R0              MOV R0,GOOD
 7293  047322
  (3)  047322  010037  001432                  BR      PERRAW
 7294  047326  000450
 7295
 7296  047330                          $PER23: LET ADDRESS := R1          MOV R1,ADDRESS
  (3)  047330  010137  001422                  LET BAD := R0               MOV R0,BAD
 7297  047334                                  LET GOOD := BAKPAT          MOV BAKPAT,GOOD
  (3)  047334  010037  001440                  BR      PERRAW
 7298  047340
  (3)  047340  013737  002174  001432  $PER24: LET ADDRESS := R5          MOV R5,ADDRESS
 7299  047346  000440                          LET BAD := R0               MOV R0,BAD
 7300                                          LET GOOD := BAKPAT          MOV BAKPAT,GOOD
 7301  047350                                  BR      PERRAW
  (3)  047350  010537  001422
 7302  047354
  (3)  047354  010037  001440
 7303  047360
  (3)  047360  013737  002174  001432
 7304  047366  000430
```

```
 7306  047370                          $PER25: LET ADDRESS := R1 - #2
  (3)  047370  010137  001422                                                                  MOV R1,ADDRESS
  (5)  047374  162737  000002  001422                                                          SUB #2,ADDRESS
 7307  047402                                  IF ABORTFLAG EQ #0
  (4)  047402  005737  001524                                                                  TST ABORTFLAG
  (6)  047406  001007                                                                          BNE L454
 7308  047410                                    SUPERVISOR
  (1)  047410  052737  040000  177776          BIS     #BIT14,PSW              ;DO IT TO IT!
 7309  047416                                    LET BAD := -2(R1)
  (3)  047416  016137  177776  001440                                                          MOV -2(R1),BAD
 7310  047424  104417                            KERNEL
 7311  047426                                  END ;OF IF ABORTFLAG
  (3)  047426                                                                          L454:;;;;;;
 7312  047426                                  IF 177654 EQ #0
  (4)  047426  005737  177654                                                                  TST 177654
  (6)  047432  001003                                                                          BNE L455
 7313  047434                                    LET GOOD := R2
  (3)  047434  010237  001432                                                                  MOV R2,GOOD
 7314  047440                                  ELSE
  (3)  047440  000402                                                                          BR L456
  (3)  047442                                                                          L455:;;;;;;
 7315  047442                                    LET GOOD := R3
  (3)  047442  010337  001432                                                                  MOV R3,GOOD
 7316  047446                                  END ;OF IF
  (3)  047446                                                                          L456:;;;;;;
 7317  047446  000400                          BR      PERRAW
 7318  047450                          PERRAW: SUBTST  <<DATA WAS A WORD>>
  (3)                                  ;*****************************************************************************
  (3)                                  ;*SUBTEST       DATA WAS A WORD
  (2)                                  ;*****************************************************************************
 7319  047450  004737  047730                  CALL    PERXOR
 7320  047454  004737  050020                  CALL    PERBNK
 7321  047460                                  IF ABORTFLAG EQ #0
  (4)  047460  005737  001524                                                                  TST ABORTFLAG
  (6)  047464  001003                                                                          BNE L457
 7322  047466                                    LET BADPC := (SP)
  (3)  047466  011637  001420                                                                  MOV (SP),BADPC
 7323  047472                                  ELSE
  (3)  047472  000402                                                                          BR L460
  (3)  047474                                                                          L457:;;;;;;
 7324  047474  004737  050324                    CALL GETDATA
 7325  047500                                  END ;OF IF ABORTFLAG
  (3)  047500                                                                          L460:;;;;;;
 7326  047500                                  IF ABORTFLAG EQ #0
  (4)  047500  005737  001524                                                                  TST ABORTFLAG
  (6)  047504  001002                                                                          BNE L461
 7327  047506  104011                            ERROR +11
 7328  047510                                  ELSE
  (3)  047510  000401                                                                          BR L462
  (3)  047512                                                                          L461:;;;;;;
 7329  047512  104012                            ERROR +12
 7330  047514                                  END ;OF IF ABORTFLAG
  (3)  047514                                                                          L462:;;;;;;
 7331  047514  000002                          RTI
 7332
 7333  047516                          PERRAB: SUBTST  <<DATA WAS A BYTE>>
```

```
  (3)                                  ;*****************************************************************************
  (3)                                  ;*SUBTEST        DATA WAS A BYTE
  (2)                                  ;*****************************************************************************
7334  047516  004737  047730                  CALL     PERXOR
7335  047522  004737  050020                  CALL     PERBNK
7336  047526                                  IF ABORTFLAG EQ #0
  (4)  047526  005737  001524                                                                  TST ABORTFLAG
  (6)  047532  001003                                                                          BNE L463
7337  047534                                    LET BADPC := (SP)
  (3)  047534  011637  001420                                                                  MOV (SP),BADPC
7338  047540                                  ELSE
  (3)  047540  000402                                                                          BR L464
  (3)  047542                                                                          L463:;;;;;;
7339  047542  004737  050324                    CALL GETDATA
7340  047546                                  END ;OF IF ABORTFLAG
  (3)  047546                                                                          L464:;;;;;;
7341  047546                                  IF ABORTFLAG EQ #0
  (4)  047546  005737  001524                                                                  TST ABORTFLAG
  (6)  047552  001002                                                                          BNE L465
7342  047554  104014                            ERROR +14
7343  047556                                  ELSE
  (3)  047556  000401                                                                          BR L466
  (3)  047560                                                                          L465:;;;;;;
7344  047560  104015                            ERROR +15
7345  047562                                  END ;OF IF ABORTFLAG
  (3)  047562                                                                          L466:;;;;;;
7346  047562  000002                          RTI
```

```
  7348  047564                          PERRA7: SUBTST  <<DATA WAS A 7 BIT BYTE>>
   (3)                                ;*****************************************************************************
   (3)                                ;*SUBTEST       DATA WAS A 7 BIT BYTE
   (2)                                ;*****************************************************************************
  7349  047564  004737  047730                  CALL    PERXOR
  7350  047570  004737  050020                  CALL    PERBNK
  7351  047574  104022                          ERROR   +22
  7352  047576  000002                          RTI
  7353
  7354  047600                          $PER26: LET GOOD2 := #100000
   (3)  047600  012737  100000  001434                                          MOV #100000,GOOD2
  7355  047606                                  LET GOOD3 := #100
   (3)  047606  012737  000100  001436                                          MOV #100,GOOD3
  7356  047614                                  GOTO PERRA3
   (2)  047614  000406                                                          BR PERRA3
  7357
  7358  047616  005037  001434          $PER27: CLR     GOOD2
  7359  047622                                  LET GOOD3 := #077
   (3)  047622  012737  000077  001436                                          MOV #077,GOOD3
  7360  047630                                  GOTO PERRA3
   (2)  047630  000400                                                          BR PERRA3
  7361
  7362  047632                          PERRA3: SUBTST  <<DATA WAS 3 WORDS>>
   (3)                                ;*****************************************************************************
   (3)                                ;*SUBTEST       DATA WAS 3 WORDS
   (2)                                ;*****************************************************************************
  7363  047632                                  PUSH    R0
   (2)  047632  010046                                                          MOV R0,-(SP)
  7364  047634  104505                          CHK1DIS
  7365  047636  104426                          READCSR
  7366  047640  013700  001526                  MOV     CSR,R0
  7367  047644  000300                          SWAB    R0
  7368  047646  042700  177600                  BIC     #^C177,R0
  7369  047652  104503                          CLR1CSR
  7370  047654                                  LET ADDRESS := R1
   (3)  047654  010137  001422                                                  MOV R1,ADDRESS
  7371  047660  005037  001432                  CLR     GOOD
  7372  047664                                  SUPERVISOR
   (1)  047664  052737  040000  177776          BIS     #BIT14,PSW              ;DO IT TO IT!
  7373  047672  011137  001440                  MOV     (R1),BAD
  7374  047676  016137  000002  001442          MOV     2(R1),BAD2
  7375  047704  104417                          KERNEL
  7376  047706  110037  001444                  MOVB    R0,BAD3
  7377  047712  105037  001445                  CLRB    BAD3+1
  7378  047716  004737  050020                  CALL    PERBNK
  7379  047722  104034                          ERROR   +34
  7380  047724                                  POP     R0
   (2)  047724  012600                                                          MOV (SP)+,R0
  7381  047726  000002                          RTI
  7382
  7383  047730                          PERXOR: SUBTST  <<DETERMINE XOR OF GOOD & BAD>>
   (3)                                ;*****************************************************************************
   (3)                                ;*SUBTEST       DETERMINE XOR OF GOOD & BAD
   (2)                                ;*****************************************************************************
  7384  047730                                  PUSH    R0
   (2)  047730  010046                                                          MOV R0,-(SP)
```

```
 7385  047732  013700  001432            MOV     GOOD,R0
 7386  047736  013737  001440  001446    MOV     BAD,BADXOR
 7387  047744  074037  001446            XOR     R0,BADXOR
 7388  047750                            POP     R0
   (2) 047750  012600                                        MOV (SP)+,R0
 7389  047752  000207                    RETURN
```

```
047754                                  $PER30: LET GOOD := R1
047754  010137  001432                                                                  MOV R1,GOOD
047760                                          LET ADDRESS := (SP) - 16
047760  011637  001422                                                                  MOV (SP),ADDRESS
047764  163737  000016  001422                                                          SUB 16,ADDRESS
047772                                          IF ABORTFLAG EQ #0
047772  005737  001524                                                                  TST ABORTFLAG
047776  001007                                                                          BNE L467
050000                                            SUPERVISOR
050000  052737  040000  177776                  BIS     #BIT14,PSW              ;DO IT TO IT!
050006                                            LET BAD := @ADDRESS
050006  017737  131410  001440                                                          MOV @ADDRESS,BAD
050014  104417                                    KERNEL
050016                                          END ;OF IF ABORTFLAG
050016                                                                          L467:;;;;;;
050016                                          GOTO PERRAW
050016  000614                                                                          BR PERRAW

050020                                  PERBNK: SUBTST<<LOG ERROR ON BAD BANK>>
                                        ;*******************************************************************
                                        ;*SUBTEST        LOG ERROR ON BAD BANK
                                        ;*******************************************************************
                                                ;WHILE WE'RE HERE LET'S MARK THE BAD BANK IN THE CONFIGURATION TABLE
050020                                          PUSH    R0,R1
050020  010046                                                                          MOV R0,-(SP)
050022  010146                                                                          MOV R1,-(SP)
050024  013701  001470                          MOV     BANK,R1
050030  070127  000006                          MUL     #6,R1
050034  052761  004000  002230                  BIS     #BIT11,CONFIG(R1)
050042  062701  000005                          ADD     #5,R1                   ;POINT TO MAX ERROR BYTE
050046  105261  002230                          INCB    CONFIG(R1)              ;BUMP BANK COUNTER
050052  001002                                  BNE     12$                     ;NO OVERFLOW - SKIP
050054  105361  002230                          DECB    CONFIG(R1)              ;SET BACK TO 255.
050060  126137  002230  002134  12$:            CMPB    CONFIG(R1),ERRMAX               ;IS IT PAST MAX?
050066  101403                                  BLOS    11$                     ;NO - SKIP
050070                                          SET     TOOMANY                 ;YES
050070  012737  177777  002054                  MOV     #-1,TOOMANY
050076                                  11$:    POP     R1,R0
050076  012601                                                                          MOV (SP)+,R1
050100  012600                                                                          MOV (SP)+,R0
050102  000207                                  RETURN

050104  010137  001422                  PERECC: MOV     R1,ADDRESS
050110  010037  001440                          MOV     R0,BAD
050114                                          IF R1 EQ TESTADD
050114  020137  002070                                                                  CMP R1,TESTADD
050120  001004                                                                          BNE L470
050122  013737  001716  001432                    MOV TSTDAT,GOOD
050130                                          ELSE
050130  000403                                                                          BR L471
050132                                                                          L470:;;;;;;
050132  013737  001720  001432                    MOV   TSTDAT+2,GOOD
050140                                          END ;OF IF R1
050140                                                                          L471:;;;;;;
050140  004737  047730                          CALL    PERXOR
050144                                          SET     HEADER
```

```
  (1)  050144  012737  177777  002146          MOV     #-1,HEADER
7425   050152  000207                          RETURN
7426
7427   050154  004737  050104          $PER31: CALL    PERECC
7428   050160                                  IF REALPAT EQ #11
  (4)  050160  023727  001744  000011                          CMP REALPAT,#11
  (6)  050166  001001                                        BNE L472
7429   050170  104042                            ERROR +42
7430   050172                                  END ;OF IF REALPAT
  (3)  050172                                                L472:::::::
7431   050172                                  IF REALPAT EQ #14
  (4)  050172  023727  001744  000014                          CMP REALPAT,#14
  (6)  050200  001001                                        BNE L473
7432   050202  104045                            ERROR +45
7433   050204                                  END ;OF IF REALPAT
  (3)  050204                                                L473:::::::
7434   050204                                  IF REALPAT EQ #15
  (4)  050204  023727  001744  000015                          CMP REALPAT,#15
  (6)  050212  001001                                        BNE L474
7435   050214  104046                            ERROR +46
7436   050216                                  END ;OF IF REALPAT
  (3)  050216                                                L474:::::::
7437   050216                                  IF REALPAT EQ #16
  (4)  050216  023727  001744  000016                          CMP REALPAT,#16
  (6)  050224  001001                                        BNE L475
7438   050226  104047                            ERROR +47
7439   050230                                  END ;OF IF REALPAT
  (3)  050230                                                L475:::::::
7440   050230                                  SET HEADER
  (1)  050230  012737  177777  002146          MOV     #-1,HEADER
7441   050236  000002                          RTI
```

```
 7443  050240  004737  050104          $PER32: CALL    PERECC
 7444  050244  104043                          ERROR   +43
 7445  050246                                  SET     HEADER
  (1)  050246  012737  177777  002146          MOV     #-1,HEADER
 7446  050254  000002                          RTI
 7447
 7448  050256  010137  001422          $PER33: MOV     R1,ADDRESS
 7449  050262  010037  001440                  MOV     R0,BAD
 7450  050266  105037  001441                  CLRB    BAD+1
 7451  050272  012737  000377  001432          MOV     #377,GOOD
 7452  050300  004737  047730                  CALL    PERXOR
 7453  050304                                  SET     HEADER
  (1)  050304  012737  177777  002146          MOV     #-1,HEADER
 7454  050312  104044                          ERROR   +44
 7455  050314                                  SET     HEADER
  (1)  050314  012737  177777  002146          MOV     #-1,HEADER
 7456  050322  000002                          RTI
 7457
 7458  050324                          GETDATA:SUBTST  <<GET DATA FROM ABORTED AREA IF POSSIBLE>>
  (3)                             ;*****************************************************************************
  (3)                             ;*SUBTEST       GET DATA FROM ABORTED AREA IF POSSIBLE
  (2)                             ;*****************************************************************************
 7459  050324                                  PUSH    R0,4,114
  (2)  050324  010046                                                             MOV R0,-(SP)
  (3)  050326  013746  000004                                                     MOV 4,-(SP)
  (4)  050332  013746  000114                                                     MOV 114,-(SP)
 7460  050336  010637  050442                  MOV     SP,GETDA1
 7461  050342  012737  050406  000004          MOV     #1$,4
 7462  050350  012737  050406  000114          MOV     #1$,114
 7463  050356  013700  001422                  MOV     ADDRESS,R0
 7464  050362  052737  000001  177750          BIS     #BIT0,MAINT
 7465  050370  052737  004000  177746          BIS     #BIT11,CONTRL           ;DISABLE ABORTS
 7466  050376  011037  001440                  MOV     (R0),BAD
 7467  050402  005037  001524                  CLR     ABORTFLAG
 7468  050406  013706  050442          1$:     MOV     GETDA1,SP               ;RESTORE KNOWN GOOD STACK POINTER
 7469  050412  042737  004000  177746          BIC     #BIT11,CONTRL           ;ENABLE ABORTS
 7470  050420  042737  000001  177750          BIC     #BIT0,MAINT
 7471  050426                                  POP     114,4,R0
  (2)  050426  012637  000114                                                     MOV (SP)+,114
  (3)  050432  012637  000004                                                     MOV (SP)+,4
  (4)  050436  012600                                                             MOV (SP)+,R0
 7472  050440  000207                          RETURN
 7473  050442  000000                  GETDA1: 0
```

```
                                        .SBTTL  ROUTINE SCOPE HANDLER
                                ;*******************************************************************************
                                ;*THIS ROUTINE CONTROLS THE LOOPING OF SUBTESTS. IT WILL INCREMENT
                                ;*AND LOAD THE DISPLAY DATA INTO THE DISPLAY REGISTER
                                ;*THE SWITCH OPTIONS PROVIDED BY THIS ROUTINE ARE:
                                ;*SW14=1         LOOP ON TEST
                                ;*SW9=1 LOOP ON ERROR
                                ;*CALL
                                ;*
050444  005237  060174          $SCOPE: INC     $DEVCT          ;TELL APT WE ARE ALIVE
050450                                  IF RESULT IS LT
 (7)    050450  002004                                                          BGE L476
050452  005037  060174                    CLR   $DEVCT
050456  105237  060176                    INCB  $UNIT
050462                                  END ;OF IF RESULT
 (3)    050462                                                          L476:::::::
050462  104410                          CKSWR                   ;;TEST  FOR CHANGE IN SOFT-SWR
050464                                  IF STOPOK NE #0 AND #SW8 SET.IN @SWR
 (4)    050464  005737  002102                                                    TST STOPOK
 (6)    050470  001410                                                          BEQ L477
 (6)    050472  032777  000400  131500                                          BIT #SW8,@SWR
 (8)    050500  001404                                                          BEQ L477
050502  005037  002102                    CLR   STOPOK
050506  000137  037124                    JMP   EXIT
050512                                  END ;OF IF STOPOK
 (3)    050512                                                          L477:::::::
050512                                  IF NOSCOPE NE #0
 (4)    050512  005737  002124                                                    TST NOSCOPE
 (6)    050516  001401                                                          BEQ L500
050520  000002                    RTI
050522                                  END ;OF IF NOSCOPE
 (3)    050522                                                          L500:::::::
050522                          1$:     IF #SW14 SET.IN @SWR THEN GOTO $OVER
 (5)    050522  032777  040000  131450                                          BIT #SW14,@SWR
 (5)    050530  001042                                                          BNE $OVER
                                ;*****START OF CODE FOR THE XOR TESTER*****
050532  000416                  $XTSTR: BR      2$              ;;IF RUNNING ON THE "XOR" TESTER CHANGE
                                                                ;;THIS INSTRUCTION TO A "NOP" (NOP=240)
050534  013746  000004                  MOV     ERRVEC,-(SP)    ;;SAVE THE CONTENTS OF THE ERROR VECTOR
050540  012737  050560  000004          MOV     #1$,ERRVEC      ;;SET FOR TIMEOUT
050546  005737  177060                  TST     177060  ;;TIME OUT ON XOR?
050552  012637  000004                  MOV     (SP)+,ERRVEC    ;;RESTORE THE ERROR VECTOR
050556  000421                          BR      $SVLAD          ;;GO TO THE NEXT TEST
050560  022626                  1$:     CMP     (SP)+,(SP)+     ;;CLEAR THE STACK AFTER A TIME OUT
050562  012637  000004                  MOV     (SP)+,ERRVEC    ;;RESTORE THE ERROR VECTOR
050566  000407                          BR      4$              ;;LOOP ON THE PRESENT TEST
050570                          2$:;*****END OF CODE FOR THE XOR TESTER*****
050570  105737  001404          3$:     TSTB    $ERFLG          ;;HAS AN ERROR OCCURRED?
050574  001412                          BEQ     $SVLAD          ;;BR IF NO
050576  032777  001000  131374          BIT     #SW9,@SWR       ;;LOOP ON ERROR?
050604  001404                          BEQ     5$              ;;BR IF NO
050606  013737  002172  002170  4$:     MOV     $LPERR,$LPADR   ;;SET LOOP ADDRESS TO LAST SCOPE
050614  000410                          BR      $OVER
050616  105037  001404          5$:     CLRB    $ERFLG          ;;ZERO THE ERROR FLAG
050622  011637  002170          $SVLAD: MOV     (SP),$LPADR     ;;SAVE SCOPE LOOP ADDRESS
050626  011637  002172                  MOV     (SP),$LPERR     ;;SAVE ERROR LOOP ADDRESS
```

```
 7519  050632  005037  002032                  CLR     $ESCAPE          ;;CLEAR THE ESCAPE FROM ERROR ADDRESS
 7520  050636  004737  050650          $OVER:  CALL    GETDIS
 7521  050642  013716  002170                  MOV     $LPADR,(SP)      ;;FUDGE RETURN ADDRESS
 7522  050646  000002                          RTI                      ;;FIXES PS
```

```
 7524  050650                                GETDIS: SUBTST  <<SUBR  DISPLAY>>
  (3)                                        ;*****************************************************************************
  (3)                                        ;*SUBTEST        SUBR    DISPLAY
  (2)                                        ;*****************************************************************************
 7525  050650  113737  001470  001403               MOVB    BANK,$BANK
 7526  050656  113737  001744  001402               MOVB    REALPAT,$PATMAR
 7527  050664                                       PUSH    R0,4
  (2)  050664  010046                                                          MOV R0,-(SP)
  (3)  050666  013746  000004                                                  MOV 4,-(SP)
 7528  050672                                       SET4    #2$
  (1)  050672  012737  050714  000004               MOV     #2$,4                   ;SET NEW VECTOR
 7529  050700  005000                               CLR     R0
 7530  050702  013700  177750                       MOV     MAINT,R0
 7531  050706  072027  000004                       ASH     #4,R0
 7532  050712  000402                               BR      3$
 7533  050714  062706  000004            2$:        ADD     #4,SP                   ;FIX STACK FROM TRAP
 7534  050720  042700  177437            3$:        BIC     #^C340,R0
 7535  050724  150037  001402                       BISB    R0,$PATMAR
 7536  050730  005737  001516                       TST     RLFLAG                  ;ARE WE RELOCATED?
 7537  050734  001403                               BEQ     1$                      ;NO - SKIP
 7538  050736  052737  100000  001402               BIS     #BIT15,$PATMAR          ;YES - SET MSB
 7539  050744  013777  001402  131230  1$:        MOV     $PATMAR,@DISPLAY
 7540  050752  013737  001402  000174               MOV     $PATMAR,DISPREG         ;SOFTWARE DISPLAY REGISTER
 7541  050760                                       POP     4,R0
  (2)  050760  012637  000004                                                  MOV (SP)+,4
  (3)  050764  012600                                                          MOV (SP)+,R0
 7542  050766  000207                               RETURN
```

```
                                        .SBTTL  ROUTINE ERROR HANDLER
                                ;*******************************************************************************
                                ;*THIS ROUTINE WILL INCREMENT THE ERROR FLAG AND THE ERROR COUNT.
                                ;*SAVE THE ERROR ITEM NUMBER AND THE ADDRESS OF THE ERROR CALL
                                ;*AND GO TO $ERRTYP ON ERROR
                                ;*THE SWITCH OPTIONS PROVIDED BY THIS ROUTINE ARE:
                                ;*SW15=1          HALT ON ERROR
                                ;*SW13=1          INHIBIT ERROR TYPEOUTS
                                ;*SW10=1          BELL ON ERROR
                                ;*SW9=1 LOOP ON ERROR
                                ;*CALL
                                ;*      ERROR   N       ;;ERROR=EMT AND N=ERROR ITEM NUMBER

                                        .ENABL  LSB
050770                          $ERROR: IF NOERROR EQ #0
050770  005737  002112                                                                  TST NOERROR
050774  001030                                                                          BNE L501
050776  104410                          CKSWR                   ;;TEST FOR CHANGE IN SOFT-SWR
051000  105237  001404          1$:     INCB    $ERFLG          ;;SET THE ERROR FLAG
051004  001775                          BEQ     1$              ;;DON'T LET THE FLAG GO TO ZERO
051006  013777  001402  131166          MOV     $PATMAR,@DISPLAY ;DISPLAY RELOCATION FLAG, BANK #, MARGINS, & PATTERN #
051014  013737  001402  060170          MOV     $PATMAR,$TESTN  ;FOR APT
051022  013737  001402  000174          MOV     $PATMAR,DISPREG ;SOFTWARE SWITCH REGISTER
051030  032777  002000  131142          BIT     #SW10,@SWR      ;;BELL ON ERROR?
051036  001402                          BEQ     2$              ;;NO - SKIP
051040                                  TYPE    $BELL           ;;RING BELL
051040  104401  002215                  TYPEIT  ,$BELL
051044  005237  001406          2$:     INC     $ERTTL          ;;COUNT THE NUMBER OF ERRORS
051050  042737  100000  001406          BIC     #BIT15,$ERTTL   ;NO NEGATIVE NUMBERS PLEASE
051056                                  END ;OF IF NOERROR
051056                                                                          L501:;;;;;;
051056  011637  001416                  MOV     (SP),ERRORPC    ;;GET ADDRESS OF ERROR INSTRUCTION
051062  162737  000002  001416          SUB     #2,ERRORPC
051070  117737  130322  001414          MOVB    @ERRORPC,$ITEMB ;;STRIP AND SAVE THE ERROR ITEM CODE
051076                                  IF NOERROR EQ #0
051076  005737  002112                                                                  TST NOERROR
051102  001027                                                                          BNE L502
051104                                  IF BADPC NE #0
051104  005737  001420                                                                  TST BADPC
051110  001405                                                                          BEQ L503
051112                                    LET ERRORPC := BADPC
051112  013737  001420  001416                                                          MOV BADPC,ERRORPC
051120                                    LET BADPC := 0
051120  005037  001420                                                                  CLR BADPC
051124                                  END ;IF
051124                                                                          L503:;;;;;;
051124  013737  001416  060166          MOV     ERRORPC,$FATAL  ;FOR APT
051132  032777  020000  131040          BIT     #SW13,@SWR      ;;SKIP TYPEOUT IF SET
051140  001012                          BNE     3$              ;;SKIP TYPEOUTS
051142                                  IF #SW5 SET.IN @SWR AND TOOMANY NE #0
051142  032777  000040  131030                                                          BIT #SW5,@SWR
051150  001404                                                                          BEQ L504
051152  005737  002054                                                                  TST TOOMANY
051156  001401                                                                          BEQ L504
051160                                    GOTO 3$
051160  000402                                                                          BR 3$
```

```
 7586  051162                                  END ;OF IF #SW5
  (3)  051162                                                                                  L504:;;;;;;
 7587  051162                                END ;OF IF NOERROR
  (3)  051162                                                                                  L502:;;;;;;
 7588  051162  004737  051354                CALL    $ERRTYP          ;;GO TO USER ERROR ROUTINE
 7589  051166                        3$:     IF NOERROR EQ #0
  (4)  051166  005737  002112                                                                        TST NOERROR
  (6)  051172  001063                                                                                BNE L505
 7590  051174  005777  131000                  TST   @SWR             ;;HALT ON ERROR
 7591  051200  100002                          BPL   7$               ;;SKIP IF CONTINUE
 7592  051202  000000                $HALT:    HALT                   ;;HALT ON ERROR!
 7593  051204  104410                          CKSWR                  ;;TEST FOR CHANGE IN SOFT-SWR
 7594  051206                        7$:       IF NOSCOPE EQ #0 AND #SW9 SET.IN @SWR
  (4)  051206  005737  002124                                                                        TST NOSCOPE
  (6)  051212  001006                                                                                BNE L506
  (6)  051214  032777  001000  130756                                                                BIT #SW9,@SWR
  (8)  051222  001402                                                                                BEQ L506
 7595  051224  013716  002172                    MOV $LPERR,(SP)      ;;FUDGE RETURN FOR LOOPING
 7596  051230                                  END ;OF IF NOSCOPE
  (3)  051230                                                                                  L506:;;;;;;
 7597  051230  005737  002032                  TST   $ESCAPE          ;;CHECK FOR AN ESCAPE ADDRESS
 7598  051234  001402                          BEQ   9$               ;;BR IF NONE
 7599  051236  013716  002032                  MOV   $ESCAPE,(SP)     ;;FUDGE RETURN ADDRESS FOR ESCAPE
 7600  051242                        9$:       IF DETFLAG EQ #0
  (4)  051242  005737  001702                                                                        TST DETFLAG
  (6)  051246  001035                                                                                BNE L507
 7601  051250                                    IF ACTFLAG NE #0 OR APTFLAG NE #0 OR FATAL$ NE #0
  (5)  051250  005737  002020                                                                        TST ACTFLAG
  (5)  051254  001006                                                                                BNE L510
  (7)  051256  005737  002022                                                                        TST APTFLAG
  (7)  051262  001003                                                                                BNE L510
  (7)  051264  005737  001454                                                                        TST FATAL$
  (9)  051270  001405                                                                                BEQ L511
  (5)  051272                                                                                  L510:;;;;;;
 7602  051272  012737  000001  060164                MOV       #1,$MSGTY       ;FOR APT
 7603  051300  000137  037124                        JMP       EXIT
 7604  051304                                    END ;OF IF ACTFLAG
  (3)  051304                                                                                  L511:;;;;;;
 7605  051304                                    IF XXDPCHAIN NE #0 AND $ERTTL HI #20
  (4)  051304  005737  002024                                                                        TST XXDPCHAIN
  (6)  051310  001414                                                                                BEQ L512
  (6)  051312  023727  001406  000020                                                                CMP $ERTTL,#20
  (8)  051320  101410                                                                                BLOS L512
 7606  051322                                        TYPE      MSG066          ;ERROR COUNT EXCEEDED 20 - ABORTING FOR XXDP CHA
  (1)  051322  104401  071453                TYPEIT  ,MSG066
 7607  051326  013700  000042                        MOV       42,R0
 7608  051332  005037  000042                        CLR       42
 7609  051336  000137  011630                        JMP       $ZAP42
 7610  051342                                    END ;OF IF XXDPCHAIN
  (3)  051342                                                                                  L512:;;;;;;
 7611  051342                                  END ;OF IF DETFLAG
  (3)  051342                                                                                  L507:;;;;;;
 7612  051342                                END ;OF IF NOERROR
  (3)  051342                                                                                  L505:;;;;;;
 7613  051342  005037  002054        10$:    CLR     TOOMANY
 7614  051346  005037  177766                CLR     CPUERR
```

```
7615  051352  000002                          RTI                     ;;RETURN
7616                                          .DSABL  LSB
```

```
                        .SBTTL  ROUTINE ERROR MESSAGE TYPEOUT

                        ;*******************************************************************************
                        ;*THIS ROUTINE USES THE "ITEM CONTROL BYTE" ($ITEMB) TO DETERMINE WHICH
                        ;*ERROR IS TO BE REPORTED. IT THEN OBTAINS, FROM THE "ERROR TABLE" ($ERRTB),
                        ;*AND REPORTS THE APPROPRIATE INFORMATION CONCERNING THE ERROR.

051354  104415          $ERRTYP:SAVREG
051356                          TYPE    $CRLF                   ;;"CARRIAGE RETURN" & "LINE FEED"
 (1)    051356  104401  002222          TYPEIT  ,$CRLF
051362  005000                  CLR     R0                      ;;PICKUP THE ITEM INDEX
051364  153700  001414          BISB    $ITEMB,R0
051370  001004                  BNE     1$                      ;;IF ITEM NUMBER IS ZERO, JUST
                                                                ;;TYPE THE PC OF THE ERROR
051372                          TYPOCT  ERRORPC,<ERROR ADDRESS>
 (1)    051372  013746  001416          MOV     ERRORPC,-(SP)   ;;SAVE ERRORPC FOR TYPEOL.
 (1)                                                            ;;ERROR ADDRESS
 (1)    051376  104402          TYPOC                           ;;GO TYPE--OCTAL ASCII(ALL DIGITS)
051400  000507                  BR      11$                     ;;GET OUT
051402  005300          1$:     DEC     R0                      ;;ADJUST THE INDEX SO THAT IT WILL
051404  006300                  ASL     R0                      ;;      WORK FOR THE ERROR TABLE
051406  006300                  ASL     R0
051410  006300                  ASL     R0
051412  062700  060606          ADD     #$ERRTB,R0              ;;FORM TABLE POINTER
051416  012037  051454          MOV     (R0)+,3$                ;;PICKUP "ERROR MESSAGE" POINTER
051422  001417                  BEQ     4$                      ;;SKIP TYPEOUT IF NO POINTER
051424  005737  002112          TST     NOERROR                 ;IS THIS REALLY AN ERROR?
051430  001003                  BNE     12$                     ;YES - SKIP
051432  005737  002146          TST     HEADER                  ;TYPE HEADER?
051436  100011                  BPL     4$                      ;NO - SKIP
051440  005737  001454          12$:    TST     FATAL$          ;WAS IT A FATAL ERROR?
051444  001402                  BEQ     2$                      ;NO - SKIP
051446                          TYPE    MSG067                  ;FATAL
 (1)    051446  104401  071536          TYPEIT  ,MSG067
051452                  2$:     TYPE                            ;;TYPE THE "ERROR MESSAGE"
 (1)    051452  104401          TYPEIT
051454  000000          3$:     .WORD   0                       ;;"ERROR MESSAGE" POINTER GOES HERE
051456                          TYPE    $CRLF                   ;;"CARRIAGE RETURN" & "LINE FEED"
 (1)    051456  104401  002222          TYPEIT  ,$CRLF
051462  012037  051506  4$:     MOV     (R0)+,5$                ;;PICKUP "DATA HEADER" POINTER
051466  001412                  BEQ     6$                      ;;SKIP TYPEOUT IF 0
051470  005737  002112          TST     NOERROR                 ;IS THIS REALLY AN ERROR?
051474  001003                  BNE     13$                     ;YES - SKIP
051476  005737  002146          TST     HEADER                  ;TYPE HEADER?
051502  100004                  BPL     6$                      ;NO - SKIP
051504                  13$:    TYPE                            ;;TYPE THE "DATA HEADER"
 (1)    051504  104401          TYPEIT
051506  000000          5$:     .WORD   0                       ;;"DATA HEADER" POINTER GOES HERE
051510                          TYPE    $CRLF                   ;;"CARRIAGE RETURN" & "LINE FEED"
 (1)    051510  104401  002222          TYPEIT  ,$CRLF
051514  012001          6$:     MOV     (R0)+,R1                ;;PICKUP "DATA TABLE" POINTER
051516  001433                  BEQ     10$                     ;;BR IF NO DATA TO BE TYPED
051520  012002                  MOV     (R0)+,R2                ;;PICKUP "DATA FORMAT" POINTER
051522  112203          7$:     MOVB    (R2)+,R3
051524  006303                  ASL     R3                      ;MAKE IT A WORD ADDRESS
051526  004773  051534          CALL    @8$(R3)
```

```
                        BR      9$
                8$:     TAG70$
                        TAG71$
                        TAG72$
                        TAG73$
                        TAG74$
                        TAG75$
                        TAG76$
                        TAG77$
                        TAG78$
                        TAG79$
                        TAG80$
                        TAG81$
                        TAG82$
                        TAG83$
                9$:     ADD     #2,R1           ;UPDATE DATA TABLE POINTE.(
                        TST     (R1)            ;;IS THERE ANOTHER NUMBER?
                        BEQ     10$             ;;BR IF NO
                        TYPE    MSG018          ;TYPE 2 SPACES
  (1)                   TYPEIT  ,MSG018
                        BR      9$              ;;LOOP

               10$:     TST     MUT             ;IS THERE A MEMORY UNDER TEST
                        BEQ     11$             ;NO - SKIP
                        INC     HEADER          ;YES - BUMP HEADER FLAG
               11$:     RESREG
                        IF #SW7 SET.IN @SWR AND DETFLAG EQ #0
                                                                        BIT #SW7,@SWR
                                                                        BEQ L513
                                                                        TST DETFLAG
                                                                        BNE L513
                          CALL  DETAIL
                        END ;OF IF #SW7
                                                                L513:;;;;;;
                        IF NOERROR NE #0
                                                                        TST NOERROR
                                                                        BEQ L514
                          CLR   NOERROR
                          SET   HEADER
                        MOV     #-1,HEADER
                        END ;OF IF NOERROR
                                                                L514:;;;;;;
                        RETURN

                ;*****************************************************************************
                ;*** OCTAL ***
                ;*****************************************************************************
                TAG70$: TYPOCT  @(R1)           ;;TYPE AN OCTAL NUMBER
                        MOV     @(R1),-(SP)     ;;SAVE @(R1) FOR TYPEOUT
                        TYPOC                   ;;GO TYPE--OCTAL ASCII(ALL DIGITS)
                        RETURN

                ;*****************************************************************************
                ;*** DECIMAL ***
                ;*****************************************************************************
                TAG71$: TYPDEC  @(R1)           ;;TYPE A DECIMAL NUMBER
```

```
 (1)  051676  017146  000000                  MOV     @(R1),-(SP)      ;;SAVE @(R1) FOR TYPEOUT
 (1)  051702  104405                          TYPDS                    ;;GO TYPE--DECIMAL ASCII WITH SIGN
      051704  000207                          RETURN

                                        ;*******************************************************************
                                        ;*** INTERLEAVE ***
                                        ;*******************************************************************
      051706                            TAG72$: PUSH    R1,R5
 (2)  051706  010146                                                                          MOV R1,-(SP)
 (3)  051710  010546                                                                          MOV R5,-(SP)
      051712  013701  001470                  MOV     BANK,R1
      051716  070127  000006                  MUL     #6,R1
      051722                                  SET     NOTAB                    ;INDICATE NO TABLE TO BE PRINTED - NOW
 (1)  051722  012737  177777  002044          MOV     #-1,NOTAB
      051730  004737  031572                  CALL    TCFIG1
      051734  005037  002044                  CLR     NOTAB
      051740                                  POP     R5,R1
 (2)  051740  012605                                                                          MOV (SP)+,R5
 (3)  051742  012601                                                                          MOV (SP)+,R1
      051744                                  TYPE    MSG014                   ;1 SPACE
 (1)  051744  104401  067047                  TYPEIT  ,MSG014
      051750  000207                          RETURN

                                        ;*******************************************************************
                                        ;*** BOX ***
                                        ;*******************************************************************
      051752  004737  052472            TAG73$: CALL    WHOBOX
      051756  000207                          RETURN

                                        ;*******************************************************************
                                        ;*** PATTERN ***
                                        ;*******************************************************************
      051760                            TAG74$: TYPOCS  REALPAT,<TYPE (0-77)>,2,Z
 (1)  051760  013746  001744                  MOV     REALPAT,-(SP)    ;;SAVE REALPAT FOR TYPEOUT
 (1)                                                                   ;;TYPE (0-77)
 (1)  051764  104403                          TYPOS                    ;;GO TYPE--OCTAL ASCII
 (1)  051766     002                          .BYTE   2                ;;TYPE 2 DIGIT(S)
 (1)  051767     001                          .BYTE   1                ;;TYPE LEADING ZEROS
      051770  000207                          RETURN

                                        ;*******************************************************************
                                        ;*** BANK ***
                                        ;*******************************************************************
      051772                            TAG75$: TYPOCS  BANK,<TYPE (0-167)>,3
 (1)  051772  013746  001470                  MOV     BANK,-(SP)       ;;SAVE BANK FOR TYPEOUT
 (1)                                                                   ;;TYPE (0-167)
 (1)  051776  104403                          TYPOS                    ;;GO TYPE--OCTAL ASCII
 (1)  052000     003                          .BYTE   3                ;;TYPE 3 DIGIT(S)
 (1)  052001     000                          .BYTE   0                ;;SUPPRESS LEADING ZEROS
      052002  000207                          RETURN

                                        ;*******************************************************************
                                        ;*** MTYPE ***
                                        ;*******************************************************************
      052004  005737  001510            TAG76$: TST     MKFLAG
      052010  001003                          BNE     1$
```

```
7748  052012                          TYPE    MSG068          ;MJ11
 (1)  052012  104401  071545         TYPEIT  ,MSG068
7749  052016  000207                 RETURN
7750  052020                    1$:   TYPE    MSG069          ;MK11
 (1)  052020  104401  071552         TYPEIT  ,MSG069
7751  052024  000207                 RETURN
7752
7753                              ;*************************************************************************
7754                              ;*** SPARE FUNCTION ***
7755                              ;*************************************************************************
7756  052026  000207              TAG77$: RETURN
7757
7758                              ;*************************************************************************
7759                              ;*** UNKNOWN DATA ***
7760                              ;*************************************************************************
7761  052030                      TAG78$: TYPE    MSG061
 (1)  052030  104401  071371         TYPEIT  ,MSG061
7762  052034  000207                 RETURN
7763
7764                              ;*************************************************************************
7765                              ;*** PHYSICAL ADDRESS ***
7766                              ;*************************************************************************
7767  052036  013737  001422  001426  TAG79$: MOV     ADDRESS,PHYADD
7768  052044  162737  060000  001426          SUB     #FIRST,PHYADD
7769  052052  013737  001470  001430          MOV     BANK,PHYADD+2
7770  052060  006237  001430                  ASR     PHYADD+2
7771  052064  103003                          BCC     1$
7772  052066  052737  100000  001426          BIS     #BIT15,PHYADD
7773  052074  012746  001426            1$:   MOV     #PHYADD,-(SP)           ;POINTER TO DOUBLE WORD ON STACK
7774  052100  004737  060044                  CALL    $DB20                   ;CALL DOUBLE PRECISION CONVERSION ROUTINE
7775  052104  062706  000002                  ADD     #2,SP                   ;FIX STACK
7776  052110                                  TYPE    $OCT8
 (1)  052110  104401  060152                 TYPEIT  ,$OCT8
7777  052114  000207                         RETURN
7778                              ;*************************************************************************
7779                              ;*** MARGIN ***
7780                              ;*************************************************************************
7781  052116                      TAG80$: TYPOCS  MARGIN,<TYPE (0,2,3,4, OR 5)>,1
 (1)  052116  013746  001474         MOV     MARGIN,-(SP)        ;;SAVE MARGIN FOR TYPEOUT
 (1)                                                             ;;TYPE (0,2,3,4, OR 5)
 (1)  052122  104403                 TYPOS                       ;;GO TYPE--OCTAL ASCII
 (1)  052124     001                 .BYTE   1                   ;;TYPE 1 DIGIT(S)
 (1)  052125     000                 .BYTE   0                   ;;SUPPRESS LEADING ZEROS
7782  052126  000207                 RETURN
```

```
                        ;*****************************************************************************
                        ;*** OCTAL BYTE ***
                        ;*****************************************************************************
                        TAG81$: TYPE    MSG018          ;2 SPACES
                                TYPEIT  ,MSG018
                                TYPOCS  @(R1),<TYPE BYTE>,3,Z
                                MOV     @(R1),-(SP)     ;;SAVE @(R1) FOR TYPEOUT
                                                        ;;TYPE BYTE
                                TYPOS                   ;;GO TYPE--OCTAL ASCII
                                .BYTE   3               ;;TYPE 3 DIGIT(S)
                                .BYTE   1               ;;TYPE LEADING ZEROS
                                TYPE    MSG014          ;SPACE
                                TYPEIT  ,MSG014
                                RETURN
                        ;*****************************************************************************
                        ;*** ARRAY ***
                        ;*****************************************************************************
                        TAG82$: PUSH    R0,R2                                   MOV R0,-(SP)
                                                                                MOV R2,-(SP)
                                MOV     #"??,MSG074
                                IF MKFLAG NE #0                                 TST MKFLAG
                                                                                BEQ L515
                                  PUSH  CSR,CSR+2,R1,CSRNO                      MOV CSR,-(SP)
                                                                                MOV CSR+2,-(SP)
                                                                                MOV R1,-(SP)
                                                                                MOV CSRNO,-(SP)
                                  MOV   MKCSRS,R1
                                  SET   R2
                                MOV     #-1,R2
                                  BEGIN FINDTURKEY                              B112:::::::
                                    FOR CSRNO := #0 TO #34 BY #4                        CLR CSRNO
                                                                                B113:::::::
                                      ASLB      R1
                                      ON.ERROR                                  BCC L516
                                        READCSR
                                        IF #BIT15!BIT4 SET.IN CSR               BIT #BIT15!BIT4,CSR
                                                                                BEQ L517
                                          MOV   CSR,R2
                                          ASH   #-4,R2
                                          BIC   #^C16,R2
                                          IF #BIT9 SET.IN CSR+2 THEN LET R2 := R2 SET.BY #BIT0
                                                                                BIT #BIT9,CSR+2
                                                                                BEQ L520
                                                                                BIS #BIT0,R2
                                                                                L520:::::::
                                          LEAVE FINDTURKEY                      BR E112
                                        END ;OF IF #BIT15!BIT4
                                                                                L517:::::::
```

```
 7812  052274                                  END ;OF ON.ERROR
  (3)  052274                                                                  L516:;;;;;;
 7813  052274                                  IFB R1 EQ #0 THEN LEAVE FINDTURKEY
  (6)  052274  105701                                                                  TSTB R1
  (6)  052276  001407                                                                  BEQ E112
 7814  052300                                END ;OF FOR CSRNO
  (6)  052300  062737  000004  001634                                                  ADD #4,CSRNO
  (5)  052306  023727  001634  000034                                                  CMP CSRNO,#34
  (5)  052314  003743                                                                  BLE B113
  (5)  052316                                                                  E113:;;;;;;;
 7815  052316                              END FINDTURKEY
  (3)  052316                                                                  E112:;;;;;;
 7816  052316                              POP   CSRNO,R1,CSR+2,CSR
  (2)  052316  012637  001634                                                          MOV (SP)+,CSRNO
  (3)  052322  012601                                                                  MOV (SP)+,R1
  (4)  052324  012637  001530                                                          MOV (SP)+,CSR+2
  (5)  052330  012637  001526                                                          MOV (SP)+,CSR
 7817  052334                              IF R2 GE #0
  (4)  052334  005702                                                                  TST R2
  (6)  052336  002416                                                                  BLT L521
 7818  052340                                IF R2 GT #9.
  (4)  052340  020227  000011                                                          CMP R2,#9.
  (6)  052344  003406                                                                  BLE L522
 7819  052346  162702  000012                  SUB       #10.,R2
 7820  052352  000302                          SWAB      R2
 7821  052354  062702  030061                  ADD       #"10,R2
 7822  052360                                ELSE
  (3)  052360  000403                                                                  BR L523
  (3)  052362                                                                  L522:;;;;;;
 7823  052362  000302                          SWAB      R2
 7824  052364  062702  030040                  ADD       #" 0,R2
 7825  052370                                END ;OF IF #BIT3
  (3)  052370                                                                  L523:;;;;;;
 7826  052370  010237  071646                MOV R2,MSG074
 7827  052374                              END ;OF IF R2
  (3)  052374                                                                  L521:;;;;;;
 7828  052374                            END ;OF IF MKFLAG
  (3)  052374                                                                  L515:;;;;;;
 7829  052374                            TYPE    MSG074
  (1)  052374  104401  071646            TYPEIT  ,MSG074
 7830  052400                            POP     R2,R0
  (2)  052400  012602                                                                  MOV (SP)+,R2
  (3)  052402  012600                                                                  MOV (SP)+,R0
 7831  052404  000207                    RETURN
 7832                                ;*****************************************************************************
 7833                                ;*** BINARY BYTE ***
 7834                                ;*****************************************************************************
 7835  052406                        TAG83$: PUSH    R0,R2
  (2)  052406  010046                                                                  MOV R0,-(SP)
  (3)  052410  010246                                                                  MOV R2,-(SP)
 7836  052412                            TYPE    MSG018          ;2 SPACES
  (1)  052412  104401  067076            TYPEIT  ,MSG018
 7837  052416  017100  000000            MOV     @(R1),R0
 7838  052422                            FOR R2 := #1 TO #8.
  (4)  052422  012702  000001                                                          MOV #1,R2
  (4)  052426                                                                  B114:;;;;;;
```

```
 7839  052426  106100                          ROLB  R0
 7840  052430                                  ON.ERROR
  (8)  052430  103004                                    BCC L524
 7841  052432  112737  000061  067051            MOVB        #'1,MSG015
 7842  052440                                  ELSE
  (3)  052440  000403                                    BR L525
  (3)  052442                                  L524:::::::
 7843  052442  112737  000060  067051            MOVB        #'0,MSG015
 7844  052450                                  END ;OF ON.ERROR
  (3)  052450                                  L525:::::::
 7845  052450                                    TYPE  MSG015
  (1)  052450  104401  067051                TYPEIT  ,MSG015
 7846  052454                                  END ;OF FOR R2
  (6)  052454  005202                                    INC R2
  (5)  052456  020227  000010                            CMP R2,#8.
  (5)  052462  003761                                    BLE B114
  (5)  052464                                  E114:::::::::
 7847  052464                                  POP     R2,R0
  (2)  052464  012602                                    MOV (SP)+,R2
  (3)  052466  012600                                    MOV (SP)+,R0
 7848  052470  000207                          RETURN
```

```
                        WHOBOX: SUBTST  <<SUBR  WHO'S BOX IS THIS?>>
                        ;*****************************************************************************
                        ;*SUBTEST        SUBR    WHO'S BOX IS THIS?
                        ;*****************************************************************************
7851 052472 005737 001510       TST     MKFLAG          ;MK11?
7852 052476 001003              BNE     1$                      ;YES - SKIP
7853                            ;NO - MJ11 - DON'T KNOW ABOUT BOX NUMBERS
7854 052500                     TYPE    MSG057                  ; ?
 (1) 052500 104401 071273       TYPEIT  ,MSG057
7855 052504 000207              RETURN
7856
7857 052506                     1$:     PUSH    CSRNO,R1,R2,R3,R4
 (2) 052506 013746 001634                                               MOV CSRNO,-(SP)
 (3) 052512 010146                                                      MOV R1,-(SP)
 (4) 052514 010246                                                      MOV R2,-(SP)
 (5) 052516 010346                                                      MOV R3,-(SP)
 (6) 052520 010446                                                      MOV R4,-(SP)
7858 052522 005037 002002              CLR     SUCCESS
7859 052526 012703 000200              MOV     #200,R3
7860 052532 012701 000060              MOV     #'0,R1
7861 052536                            BEGIN WHODUNIT
 (3) 052536                                                     B115:;;;;;;
7862 052536                            FOR R4 := #0 TO #16 BY #2
 (3) 052536 005004                                                      CLR R4
 (3) 052540                                                     B116:;;;;;;
7863 052540                              IFB R3 SET.IN MKCSRS
 (5) 052540 130337 001706                                               BITB R3,MKCSRS
 (7) 052544 001447                                                      BEQ L526
7864 052546 016437 001572 001636           MOV CSRFB(R4),FIRSTB
7865 052554 016437 001612 001640           MOV CSRLB(R4),LASTB
7866 052562                                IF FIRSTB LE BANK AND LASTB GE BANK
 (4) 052562 023737 001636 001470                                        CMP FIRSTB,BANK
 (6) 052570 003035                                                      BGT L527
 (6) 052572 023737 001640 001470                                        CMP LASTB,BANK
 (8) 052600 002431                                                      BLT L527
7867 052602 104426                           READCSR
7868 052604 016402 001552              MOV         CSR2(R4),R2
7869 052610 072227 177764              ASH         #-12.,R2
7870 052614 042702 177770              BIC         #^C7,R2
7871 052620                            CASE R2
 (2) 052620 010246                                                      MOV R2,-(SP)
 (2) 052622 006316                                                      ASL @SP
 (3) 052624 004737 052650                                               JSR PC,L530
7872 052630 052744                           ADD..0
7873 052632 052754                           ADD..1
7874 052634 052774                           ADD..2
7875 052636 053024                           ADD..3
7876 052640 053054                           ADD..4
7877 052642 053056                           ADD..5
7878 052644 053076                           ADD..6
7879 052646 053126                           ADD..7
7880 052650                            END ;OF CASE R2
 (3) 052650                                                     L530:;;;;;;
 (2) 052650 062616                                                      ADD (SP)+,@SP
 (2) 052652 013646                                                      MOV @(SP)+,-(SP)
 (2) 052654 004736                                                      JSR PC,@(SP)+
```

```
                                        IF SUCCESS NE #0 THEN LEAVE WHODUNIT
                                                                                        TST SUCCESS
                                                                                        BNE E115
                                          END ;OF IF FIRSTB
                                                                                L527:;;;;;;;
                                        END ;OF IFB R3
                                                                                L526:;;;;;;;
                                        ROR   R3
                                        INC   R1
                                      END ;OF FOR CSRNO
                                                                                        ADD #2,R4
                                                                                        CMP R4,#16
                                                                                        BLE B116
                                                                                E116:;;;;;;;;
                                      END WHODUNIT
                                                                                E.15:;;;;;;;
                                      IF SUCCESS NE #0
                                                                                        TST SUCCESS
                                                                                        BEQ L531
                                        MOVB  R1,MSG057+1
                                      END ;OF IF SUCCESS
                                                                                L531:;;;;;;;
                                      TYPE    MSG057
                                      TYPEIT  ,MSG057
                                      MOVB    #'?,MSG057+1
                                      POP     R4,R3,R2,R1,CSRNO
                                                                                        MOV (SP)+,R4
                                                                                        MOV (SP)+,R3
                                                                                        MOV (SP)+,R2
                                                                                        MOV (SP)+,R1
                                                                                        MOV (SP)+,CSRNO
                                      RETURN

                                ADD..0: SET     SUCCESS
                                        MOV     #-1,SUCCESS
                                        RETURN
                                ADD..1: IF #BIT2 OFF.IN ADDRESS THEN LET SUCCESS := ONES
                                                                                        BIT #BIT2,ADDRESS
                                                                                        BNE L532
                                                                                        MOV ONES,SUCCESS
                                                                                L532:;;;;;;
                                        RETURN
                                ADD..2: IF #BIT3 OFF.IN ADDRESS AND #BIT2 OFF.IN ADDRESS THEN LET SUCCESS := ONES
                                                                                        BIT #BIT3,ADDRESS
                                                                                        BNE L533
                                                                                        BIT #BIT2,ADDRESS
                                                                                        BNE L533
                                                                                        MOV ONES,SUCCESS
                                                                                L533:;;;;;;
                                        RETURN
                                ADD..3: IF #BIT3 SET.IN ADDRESS AND #BIT2 OFF.IN ADDRESS THEN LET SUCCESS := ONES
                                                                                        BIT #BIT3,ADDRESS
                                                                                        BEQ L534
                                                                                        BIT #BIT2,ADDRESS
                                                                                        BNE L534
                                                                                        MOV ONES,SUCCESS
```

```
  (5)  053052                                                                                    L534:;;;;;;
7903   053052  000207                    ;       RETURN
7904   053054  000207                    ADD..4: RETURN
7905   053056                            ADD..5: IF #BIT2 SET.IN ADDRESS THEN LET SUCCESS := ONES
  (4)  053056  032737  000004  001422                                                              BIT #BIT2,ADDRESS
  (6)  053064  001403                                                                            BEQ L535
  (5)  053066  013737  002154  002002                                                            MOV ONES,SUCCESS
  (3)  053074                                                                                    L535:;;;;;;
7906   053074  000207                    ;       RETURN
7907   053076                            ADD..6: IF #BIT3 OFF.IN ADDRESS AND #BIT2 SET.IN ADDRESS THEN LET SUCCESS := ONES
  (4)  053076  032737  000010  001422                                                              BIT #BIT3,ADDRESS
  (6)  053104  001007                                                                            BNE L536
  (6)  053106  032737  000004  001422                                                            BIT #BIT2,ADDRESS
  (8)  053114  001403                                                                            BEQ L536
  (7)  053116  013737  002154  002002                                                            MOV ONES,SUCCESS
  (5)  053124                                                                                    L536:;;;;;;
7908   053124  000207                    ;       RETURN
7909   053126                            ADD..7: IF #BIT3 SET.IN ADDRESS AND #BIT2 SET.IN ADDRESS THEN LET SUCCESS := ONES
  (4)  053126  032737  000010  001422                                                              BIT #BIT3,ADDRESS
  (6)  053134  001407                                                                            BEQ L537
  (6)  053136  032737  000004  001422                                                            BIT #BIT2,ADDRESS
  (8)  053144  001403                                                                            BEQ L537
  (7)  053146  013737  002154  002002                                                            MOV ONES,SUCCESS
  (5)  053154                                                                                    L537:;;;;;;
7910   053154  000207                            RETURN
7911
7912   053156                            DETAIL: SUBTST  <<SUBR  DETAILED ERROR REPORT>>
  (3)                                       ;*****************************************************************************
  (3)                                       ;*SUBTEST       SUBR    DETAILED ERROR REPORT
  (2)                                       ;*****************************************************************************
7913   053156  005237  001702                        INC     DETFLAG
7914   053162  022737  000003  001702                CMP     #3,DETFLAG
7915   053170  101454                                BLOS    4$
7916   053172  022737  000002  001702                CMP     #2,DETFLAG
7917   053200  001427                                BEQ     2$
7918   053202                                        PUSH    HEADER,MUT                          MOV HEADER,-(SP)
  (2)  053202  013746  002146                                                                    MOV MUT,-(SP)
  (3)  053206  013746  001500
7919   053212                                        SET     HEADER
  (1)  053212  012737  177777  002146                MOV     #-1,HEADER
7920   053220  005037  001500                        CLR     MUT
7921   053224  010037  001664                        MOV     R0,DETR0
7922   053230  012700  001666                        MOV     #DETR1,R0
7923   053234  010120                                MOV     R1,(R0)+
7924   053236  010220                                MOV     R2,(R0)+
7925   053240  010320                                MOV     R3,(R0)+
7926   053242  010420                                MOV     R4,(R0)+
7927   053244  010520                                MOV     R5,(R0)+
7928   053246  010620                                MOV     SP,(R0)+
7929   053250  013700  001664                        MOV     DETR0,R0
7930   053254  104013                                ERROR   +13
7931   053256  000412                                BR      1$
7932   053260                            2$:         PUSH    HEADER,MUT                          MOV HEADER,-(SP)
  (2)  053260  013746  002146                                                                    MOV MUT,-(SP)
  (3)  053264  013746  001500
7933   053270                                        SET     HEADER
```

```
  (1)  053270  012737  177777  002146          MOV     #-1,HEADER
7934   053276  005037  001500                  CLR     MUT
7935   053302  104031                          ERROR   +31
7936   053304                          1$:     POP     MUT,HEADER                              MOV (SP)+,MUT
  (2)  053304  012637  001500                                                                  MOV (SP)+,HEADER
  (3)  053310  012637  002146
7937
7938   053314  004737  053156                  ;WARNING RECURSIVE
                                               CALL    DETAIL
7939   053320  000207                          RETURN
7940   053322                          4$:     PUSH    R0                                      MOV R0,-(SP)
  (2)  053322  010046
7941   053324                                  TYPE    MSG088          ;KERNEL STACK
  (1)  053324  104401  072260                TYPEIT  ,MSG088
7942   053330                                  FOR     R0 := SP TO #KERSTK-2 BY #2             MOV SP,R0
  (3)  053330  010600
  (3)  053332                                                                                 L.17:;;;;;;;
7943   053332                                    TYPE  $CRLF
  (1)  053332  104401  002222                TYPEIT  ,$CRLF
7944   053336                                    TYPOCT      R0
  (1)  053336  010046                          MOV     R0,-(SP)        ;;SAVE R0 FOR TYPEOUT
  (1)  053340  104402                        ; TYPOC                   ;;GO TYPE--OCTAL ASCII(ALL DIGITS)
7945   053342                                    TYPE          MSG018          ;2 SPACES
  (1)  053342  104401  067076                TYPEIT  ,MSG018
7946   053346                                    TYPOCT        (R0)
  (1)  053346  011046                          MOV     (R0),-(SP)      ;;SAVE (R0) FOR TYPEOUT
  (1)  053350  104402                          TYPOC                   ;;GO TYPE--OCTAL ASCII(ALL DIGITS)
7947   053352                                  END ;OF FOR R0
  (6)  053352  062700  000002                                                                 ADD #2,R0
  (5)  053356  020027  001376                                                                 CMP R0,#KERSTK-2
  (5)  053362  003763                                                                         BLE B117
  (5)  053364                                                                               E117:;;;;;;;
7948
7949   053364  042737  030000  177776          ;SET PREVIOUS MODE TO SUPERVISOR
7950   053372  052737  010000  177776          BIC     #BIT13!BIT12,PSW
7951   053400  006506                          BIS     #BIT12,PSW
7952   053402                                  MFPI    SSP
  (2)  053402  012600                          POP     R0                                      MOV (SP)+,R0
7953   053404                                  TYPE    MSG089          ;SUPERVISOR STACK
  (1)  053404  104401  072276                TYPEIT  ,MSG089
7954   053410                                  IF R0 LT #SUPSTK
  (4)  053410  020027  000700                                                                 CMP R0,#SUPSTK
  (6)  053414  002016                                                                         BGE L540
7955   053416                                    FOR R0 := R0 TO #SUPSTK-2 BY #2
  (3)  053416                                                                               B120:;;;;;;;
7956   053416                                      TYPE        $CRLF
  (1)  053416  104401  002222                TYPEIT  ,$CRLF
7957   053422                                      TYPOCT      R0
  (1)  053422  010046                          MOV     R0,-(SP)        ;;SAVE R0 FOR TYPEOUT
  (1)  053424  104402                          TYPOC                   ;;GO TYPE--OCTAL ASCII(ALL DIGITS)
7958   053426                                      TYPE          MSG018          ;2 SPACES
  (1)  053426  104401  067076                TYPEIT  ,MSG018
7959   053432                                      TYPOCT        (R0)
  (1)  053432  011046                          MOV     (R0),-(SP)      ;;SAVE (R0) FOR TYPEOUT
  (1)  053434  104402                          TYPOC                   ;;GO TYPE--OCTAL ASCII(ALL DIGITS)
7960   053436                                    END ;OF FOR R0
  (6)  053436  062700  000002                                                                 ADD #2,R0
```

```
  (5)  053442  020027  000676                                                                  CMP R0,#SUPSTK-2
  (5)  053446  003763                                                                          BLE B120
  (5)  053450                                                                          E120:;;;;;;;
 7961  053450                                  ELSE
  (3)  053450  000402                                                                          BR L541
  (3)  053452                                                                          L540:;;;;;;
 7962  053452                                    TYPE  MSG091          ;IS EMPTY
  (1)  053452  104401  072334                  TYPEIT  ,MSG091
 7963  053456                                  END ;OF IF R0
  (3)  053456                                                                          L541:;;;;;;
 7964                                          ;SET PREVIOUS MODE TO USER
 7965  053456  052737  030000  177776          BIS     #BIT13!BIT12,PSW
 7966  053464  006506                          MFPI    USP
 7967  053466                                  POP     R0
  (2)  053466  012600                                                                          MOV (SP)+,R0
 7968  053470                                  TYPE    MSG090          ;USER STACK
  (1)  053470  104401  072320                  TYPEIT  ,MSG090
 7969  053474                                  IF R0 LT #USESTK
  (4)  053474  020027  000600                                                                  CMP R0,#USESTK
  (6)  053500  002016                                                                          BGE L542
 7970  053502                                    FOR R0 := R0 TO #USESTK-2 BY #2
  (3)  053502                                                                          B121:;;;;;;
 7971  053502                                      TYPE        $CRLF
  (1)  053502  104401  002222                  TYPEIT  ,$CRLF
 7972  053506                                      TYPOCT      R0
  (1)  053506  010046                          MOV     R0,-(SP)        ;;SAVE R0 FOR TYPEOUT
  (1)  053510  104402                          TYPOC                   ;;GO TYPE--OCTAL ASCII(ALL DIGITS)
 7973  053512                                      TYPE        MSG018          ;2 SPACES
  (1)  053512  104401  067076                  TYPEIT  ,MSG018
 7974  053516                                      TYPOCT      (R0)
  (1)  053516  011046                          MOV     (R0),-(SP)      ;;SAVE (R0) FOR TYPEOUT
  (1)  053520  104402                          TYPOC                   ;;GO TYPE--OCTAL ASCII(ALL DIGITS)
 7975  053522                                    END ;OF FOR R0
  (6)  053522  062700  000002                                                                  ADD #2,R0
  (5)  053526  020027  000576                                                                  CMP R0,#USESTK-2
  (5)  053532  003763                                                                          BLE B121
  (5)  053534                                                                          E121:;;;;;;;
 7976  053534                                  ELSE
  (3)  053534  000402                                                                          BR L543
  (3)  053536                                                                          L542:;;;;;;
 7977  053536                                    TYPE  MSG091          ;IS EMPTY
  (1)  053536  104401  072334                  TYPEIT  ,MSG091
 7978  053542                                  END ;OF IF R0
  (3)  053542                                                                          L543:;;;;;;
 7979  053542  005037  001702                  CLR     DETFLAG
 7980  053546                                  POP     R0
  (2)  053546  012600                                                                          MOV (SP)+,R0
 7981  053550  000207                          RETURN
```

```
                                        .TITLE  POWER FAIL AUTO RESTART

                                        .SBTTL  ROUTINE POWER DOWN AND UP
                                ;*******************************************************************************
                                ;POWER DOWN ROUTINE
053552                          $PWRDN: ;SAVE CACHE & MARGIN STATUS
053552                                  PUSH    CONTRL,MAINT
  (2) 053552  013746  177746                                                   MOV CONTRL,-(SP)
  (3) 053556  013746  177750                                                   MOV MAINT,-(SP)
053562  104423                        CACHON
053564  012737  054472  000024        MOV     #$ILLUP,PWRVEC ;;SET FOR FAST UP
053572  012737  000340  000026        MOV     #340,PWRVEC+2   ;;PRIO:7
053600                                  PUSH    R0,R1,R2,R3,R4,R5,CSRNO
  (2) 053600  010046                                                           MOV R0,-(SP)
  (3) 053602  010146                                                           MOV R1,-(SP)
  (4) 053604  010246                                                           MOV R2,-(SP)
  (5) 053606  010346                                                           MOV R3,-(SP)
  (6) 053610  010446                                                           MOV R4,-(SP)
  (7) 053612  010546                                                           MOV R5,-(SP)
  (8) 053614  013746  001634                                                   MOV CSRNO,-(SP)
                                        ;SAVE USER PAR'S & PDR7
053620  012700  177700                  MOV     #177700,R0
053624  012701  000021                  MOV     #17.,R1
053630                          1$:     PUSH    -(R0)
  (2) 053630  014046                                                           MOV -(R0),-(SP)
053632  077102                          SOB     R1,1$
                                        ;SAVE SUPERVISOR PAR'S
053634  012700  172300                  MOV     #172300,R0
053640  012701  000020                  MOV     #16.,R1
053644                          2$:     PUSH    -(R0)
  (2) 053644  014046                                                           MOV -(R0),-(SP)
053646  077102                          SOB     R1,2$
053650                                  IF RLFLAG NE #0 THEN $CALL WOOPS
  (4) 053650  005737  001516                                                   TST RLFLAG
  (6) 053654  001402                                                           BEQ L544
  (4) 053656  004737  054500                                                   JSR PC,WOOPS
  (3) 053662                                                                 L544:::::::
                                ;COPY KERNEL MAP TO USER & SUPERVISOR
053662  012700  172300                  MOV     #KIPDR0,R0
053666  012701  177600                  MOV     #UIPDR0,R1
053672  012702  172200                  MOV     #SIPDR0,R2
053676  012703  000040                  MOV     #32.,R3
053702  011021                  3$:     MOV     (R0),(R1)+
053704  012022                          MOV     (R0)+,(R2)+
053706  077303                          SOB     R3,3$
                                        ;SAVE USER & SUPERVISOR STACK POINTERS
053710                                  USER
  (1) 053710  052737  140000  177776    BIS     #BIT15!BIT14,PSW                ;DO IT TO IT!
053716  010600                          MOV     USP,R0
053720  104417                          KERNEL
053722                                  PUSH    R0
  (2) 053722  010046                                                           MOV R0,-(SP)
053724                                  SUPERVISOR
  (1) 053724  052737  040000  177776    BIS     #BIT14,PSW              ;DO IT TO IT!
053732  010600                          MOV     SSP,R0
053734  104417                          KERNEL
```

```
                                        PUSH    R0                              MOV R0,-(SP)

                                        ;SAVE MK11 REGISTERS
                                        MOV     MKCSRS,R1       ;GET CSR'S BYTE
                                        BEGIN LCSRSAVE
                                                                                B122:;;;;;;;
                                          FOR CSRNO := #0 TO #34 BY #4          CLR CSRNO
                                                                                B123:;;;;;;;
                                            ASLB        R1                      BCC L545
                                            ON.ERROR
                                              READCSR
                                              PUSH      CSR,CSR+2               MOV CSR,-(SP)
                                                                                MOV CSR+2,-(SP)
                                            END ;OF ON.ERROR
                                                                                L545:;;;;;;;
                                            IFB R1 EQ #0 THEN LEAVE LCSRSAVE    TSTB R1
                                                                                BEQ E122
                                          END ;OF FOR CSRNO                     ADD #4,CSRNO
                                                                                CMP CSRNO,#34
                                                                                BLE B123
                                                                                E123:;;;;;;;
                                        END LCSRSAVE
                                                                                E122:;;;;;;;
                                        ;SAVE MMR0,1,2,3
                                        PUSH    MMR0,MMR1,MMR2,MMR3             MOV MMR0,-(SP)
                                                                                MOV MMR1,-(SP)
                                                                                MOV MMR2,-(SP)
                                                                                MOV MMR3,-(SP)
                                        ;SAVE KERNEL PAR'S
                                        MOV     #172400,R0
                                        MOV     #16.,R1
                                4$:     PUSH    -(R0)                           MOV -(R0),-(SP)
                                        SOB     R1,4$
                                        ;SAVE UNIBUS MAP REGISTERS
                                        PUSH    MAPH36,MAPL36,MAPH0,MAPL0       MOV MAPH36,-(SP)
                                                                                MOV MAPL36,-(SP)
                                                                                MOV MAPH0,-(SP)
                                                                                MOV MAPL0,-(SP)
                                        ;SAVE POSSIBLE SOFTWARE SWITCH REGISTER
                                        PUSH    @SWR                            MOV @SWR,-(SP)

                                        ;SAVE STACK POINTER
                                        MOV     SP,$SAVR6       ;;SAVE SP
                                        ;NOW SET UP REAL VECTOR
                                        MOV     #$PWRUP,PWRVEC  ;;SET UP VECTOR
                                $DOWN:  HALT
                                        BR      .-2             ;;HANG UP
                                ;*******************************************************************************
```

```
                                        ;POWER UP ROUTINE
8053   054106  012737  054472  000024 $PWRUP: MOV     #$ILLUP,PWRVEC ;;SET FOR FAST DOWN
                                        ;RESTORE STACK POINTER
8055   054114  013706  054476              MOV     $SAVR6,SP       ;;GET SP
8056   054120  005037  054476              CLR     $SAVR6          ;;WAIT LOOP FOR THE TTY
8057   054124  005237  054476        1$:   INC     $SAVR6          ;;WAIT FOR THE INC
8058   054130  001375                      BNE     1$              ;;OF A WORD
                                        ;RESTORE POSSIBLE SOFTWARE SWITCH REGISTER
8060   054132                          POP     @SWR
  (2)  054132  012677  126042                                                  MOV (SP)+,@SWR
                                        ;RESTORE UNIBUS MAP
8062   054136                          POP     MAPLO,MAPHO,MAPL36,MAPH36
  (2)  054136  012637  170200                                                  MOV (SP)+,MAPLO
  (3)  054142  012637  170202                                                  MOV (SP)+,MAPHO
  (4)  054146  012637  170370                                                  MOV (SP)+,MAPL36
  (5)  054152  012637  170372                                                  MOV (SP)+,MAPH36
8063   054156  004737  035524              CALL    LOWMAP          ;SETUP LOWER 16K OF UNIBUS MAP
                                        ;RESTORE KERNEL PAR'S & PDR'S
8065   054162  012700  172340              MOV     #172340,R0
8066   054166  012702  172300              MOV     #KIPDR0,R2
8067   054172  012701  000020              MOV     #16.,R1
8068   054176                        6$:   POP     (R0)+
  (2)  054176  012620                                                          MOV (SP)+,(R0)+
8069   054200  012722  077406              MOV     #77406,(R2)+
8070   054204  077104                      SOB     R1,6$
                                        ;RESTORE MMR3,2,1,0
8072   054206                          POP     MMR3,MMR2,MMR1,MMR0
  (2)  054206  012637  172516                                                  MOV (SP)+,MMR3
  (3)  054212  012637  177576                                                  MOV (SP)+,MMR2
  (4)  054216  012637  177574                                                  MOV (SP)+,MMR1
  (5)  054222  012637  177572                                                  MOV (SP)+,MMR0
                                        ;RESTORE MK11 REGISTERS
8074   054226  013701  001706              MOV     MKCSRS,R1       ;GET CSR'S BYTE
8075   054232  042701  177400              BIC     #177400,R1
8076   054236                          BEGIN LCSRRESTORE
  (3)  054236                                                                  B124::::::;
8077   054236
  (3)  054236  012737  000034  001634        FOR CSRNO := #34 DOWNTO #0 BY #4
                                                                               MOV #34,CSRNO
  (3)  054244                                                                  B125::::::;
8078   054244  106201                        ASRB    R1
8079   054246                                ON.ERROR
  (8)  054246  103005                                                          BCC L546
8080   054250                                  POP     CSR+2,CSR
  (2)  054250  012637  001530                                                  MOV (SP)+,CSR+2
  (3)  054254  012637  001526                                                  MOV (SP)+,CSR
8081   054260  104425                        LOADCSR
8082   054262                                END ;OF ON.ERROR
  (3)  054262                                                                  L546::::::;
8083   054262                                IFB R1 EQ #0 THEN LEAVE LCSRRESTORE
  (6)  054262  105701                                                          TSTB R1
  (6)  054264  001407                                                          BEQ E124
8084   054266                          END ;OF FOR CSRNO
  (6)  054266  162737  000004  001634        SUB #4,CSRNO
  (5)  054274  023727  001634  000000        CMP CSRNO,#0
  (5)  054302  002360                        BGE B125
  (5)  054304                           E125:::::::;
```

```
                        END LCSRRESTORE                                                 E124:::::::

                        ;COPY KERNEL MAP TO USER & SUPERVISOR
                        MOV     #KIPDR0,R0
                        MOV     #UIPDR0,R1
                        MOV     #SIPDR0,R2
                        MOV     #32.,R3
                3$:     MOV     (R0),(R1)+
                        MOV     (R0)+,(R2)+
                        SOB     R3,3$
                        ;RESTORE SUPERVISOR & USER STACK POINTERS
                        POP     R0                                                                      MOV (SP)+,R0
                        SUPERVISOR
                        BIS     #BIT14,PSW              ;DO IT TO IT!
                        MOV     R0,SSP
                        KERNEL
                        POP     R0                                                                      MOV (SP)+,R0
                        USER
                        BIS     #BIT15!BIT14,PSW                ;DO IT TO IT!
                        MOV     R0,USP
                        KERNEL
                        ;RESTORE SUPERVISOR PAR'S
                        MOV     #172240,R0
                        MOV     #16.,R1
                7$:     POP     (R0)+                                                                   MOV (SP)+,(R0)+
                        SOB     R1,7$
                        ;RESTORE USER PAR'S & PDR7
                        MOV     #177636,R0
                        MOV     #17.,R1
                8$:     POP     (R0)+                                                                   MOV (SP)+,(R0)+
                        SOB     R1,8$
                        ;RESTORE POSSIBLE SOFTWARE DISPLAY REGISTER
                        MOV     $PATMAR,@DISPLAY
                        MOV     $PATMAR,DISPREG
                        POP     CSRNO,R5,R4,R3,R2,R1,R0                                                 MOV (SP)+,CSRNO
                                                                                                        MOV (SP)+,R5
                                                                                                        MOV (SP)+,R4
                                                                                                        MOV (SP)+,R3
                                                                                                        MOV (SP)+,R2
                                                                                                        MOV (SP)+,R1
                                                                                                        MOV (SP)+,R0
                        MOV     #$PWRDN,PWRVEC  ;;SET UP THE POWER DOWN VECTOR
                        TYPE    MSG051          ;REPORT THE POWER FAILURE
                        TYPEIT  ,MSG051
                        ;RESTORE MARGIN & CACHE STATUS
                        POP     MAINT,CONTRL                                                            MOV (SP)+,MAINT
                                                                                                        MOV (SP)+,CONTRL
                        RTI
                $ILLUP: HALT                    ;;THE POWER UP SEQUENCE WAS STARTED
                        BR      .-2             ;; BEFORE THE POWER DOWN WAS COMPLETE
```

F12

```
POWER FAIL AUTO RESTART MACY11 30A(1052)  28-FEB-78  09:20  PAGE 60-4
CEMKAA.MA2    24-FEB-78 08:17          POUTINE POWER DOWN AND UP                                    SEQ 0355

  8124  054476  000000                 $SAVR6: 0                        ;;PUT THE SP HERE
  8125                                         .EVEN
```

```
 8127  054500                          WOOPS:  SUBTST  <<POWER FAIL WHILE RELOCATED>>
  (3)                                ;*****************************************************************************
  (3)                                ;*SUBTEST       POWER FAIL WHILE RELOCATED
  (2)                                ;*****************************************************************************
 8128  054500                                  PUSH    BANK                                    MOV BANK,-(SP)
  (2)  054500  013746  001470
 8129  054504  005037  001470                  CLR     BANK
 8130  054510                                  MAP     BANK                                    MOV R3,-(SP)
  (3)  054510  010346
  (1)  054512  013703  001470                  MOV     BANK,R3
  (1)  054516  004737  034510                  CALL    MAPPER                                  MOV (SP)+,R3
  (3)  054522  012603
 8131  054524                                  SUPERVISOR
  (1)  054524  052737  040000  177776         BIS     #BIT14,PSW              ;DO IT TO IT!
 8132  054532  013737  060024  055062         MOV     FIRST+PWRVEC,WOOPSAV
 8133  054540  013737  060026  055064         MOV     FIRST+PWRVEC+2,WOOPSAV+2
 8134  054546                                  BMOV    FIRST+WOOPUP,WOOPSAV+4,WOOPEND-WOOPUP/2+12.
  (1)  054546  004537  037342                      JSR R5,BLOCK3
  (1)  054552  000102                              WOOPEND-WOOPUP/2+12.
  (1)  054554  055066                              WOOPSAV+4
  (1)  054556  134656                              FIRST+WOOPUP
 8135  054560  012737  054656  060024         MOV     #WOOPUP,FIRST+PWRVEC
 8136  054566  012737  000340  060026         MOV     #340,FIRST+PWRVEC+2
 8137  054574                                  BMOV    WOOPUP,FIRST+WOOPUP,WOOPEND-WOOPUP/2
  (1)  054574  004537  037342                      JSR R5,BLOCK3
  (1)  054600  000066                              WOOPEND-WOOPUP/2
  (1)  054602  134656                              FIRST+WOOPUP
  (1)  054604  054656                              WOOPUP
 8138  054606  012700  172340                  MOV     #KIPAR0,R0
 8139  054612  012701  135032                  MOV     #FIRST+WOOPEND,R1
 8140  054616  012702  000010                  MOV     #8.,R2
 8141  054622  012021                  1$:     MOV     (R0)+,(R1)+
 8142  054624  077202                          SOB     R2,1$
 8143  054626  013721  172516                  MOV     MMR3,(R1)+
 8144  054632  013721  177576                  MOV     MMR2,(R1)+
 8145  054636  013721  177574                  MOV     MMR1,(R1)+
 8146  054642  013721  177572                  MOV     MMR0,(R1)+
 8147  054646  104417                          KERNEL
 8148  054650                                  POP     BANK                                    MOV (SP)+,BANK
  (2)  054650  012637  001470
 8149  054654  000207                          RETURN
 8150
 8151  054656                          WOOPUP: SUBTST  <<POWER UP FROM BANK 0 TO RELOCATION>>
  (3)                                ;*****************************************************************************
  (3)                                ;*SUBTEST       POWER UP FROM BANK 0 TO RELOCATION
  (2)                                ;*****************************************************************************
 8152  054656  012700  055032                  MOV     #WOOPEND,R0
 8153  054662  012701  172340                  MOV     #KIPAR0,R1
 8154  054666  012703  172300                  MOV     #KIPDR0,R3
 8155  054672  012702  000010                  MOV     #8.,R2
 8156  054676  012021                  1$:     MOV     (R0)+,(R1)+
 8157  054700  012723  077406                  MOV     #77406,(R3)+
 8158  054704  077204                          SOB     R2,1$
 8159  054706  012037  172516                  MOV     (R0)+,MMR3
 8160  054712  012037  177576                  MOV     (R0)+,MMR2
 8161  054716  012037  177574                  MOV     (R0)+,MMR1
```

```
 8162  054722  012037  177572                  MOV     (R0)+,MMR0
 8163  054726  013706  054476                  MOV     $SAVR6,SP
 8164  054732                                  PUSH    BANK                                    MOV BANK,-(SP)
   (2) 054732  013746  001470
 8165  054736  005037  001470                  CLR     BANK
 8166  054742                                  MAP     BANK                                    MOV R3,-(SP)
   (3) 054742  010346
   (1) 054744  013703  001470                  MOV     BANK,R3
   (1) 054750  004737  034510                  CALL    MAPPER                                  MOV (SP)+,R3
   (3) 054754  012603
 8167  054756                                  SUPERVISOR
   (1) 054756  052737  040000  177776          BIS     #BIT14,PSW              ;DO IT TO IT!
 8168  054764  013737  055062  060024          MOV     WOOPSAV,FIRST+PWRVEC
 8169  054772  013737  055064  060026          MOV     WOOPSAV+2,FIRST+PWRVEC+2
 8170                                          ;SIMULATE THE FOLLOWING BLOCK MOV BUT WITH NO STACK ACCESSES
 8171                                          ;BMOV   WOOPSAV+4,FIRST+WOOPUP,WOOPEND-WOOPUP/2+12.
 8172  055000  012700  055066                  MOV     #WOOPSAV+4,R0
 8173  055004  012701  000102                  MOV     #WOOPEND-WOOPUP/2+12.,R1
 8174  055010  012702  134656                  MOV     #FIRST+WOOPUP,R2
 8175  055014  012022          2$:             MOV     (R0)+,(R2)+
 8176  055016  077102                          SOB     R1,2$
 8177
 8178  055020  104417                          KERNEL
 8179  055022                                  POP     BANK                                    MOV (SP)+,BANK
   (2) 055022  012637  001470
 8180  055026  000137  054106                  JMP     $PWRUP
 8181  055032  000014          WOOPEND:        .BLKW   12.
 8182  055062  000104          WOOPSAV:        .BLKW   WOOPEND-WOOPUP/2+12.+2
 8183  055272      001         KERLIMIT:
 8184  055272  002506                          .PRINT  60000-KERLIMIT  ;SUPERVISOR ADDRESSES LEFT
```

```
                                                .TITLE  IO SUBROUTINES

                                                .SBTTL  ROUTINE TYPE

                                        ;*****************************************************************************
                                        ;*ROUTINE TO TYPE ASCIZ MESSAGE. MESSAGE MUST TERMINATE WITH A 0 BYTE.
                                        ;*THE ROUTINE WILL INSERT A NUMBER OF NULL CHARACTERS AFTER A LINE FEED.
                                        ;*NOTE1:        $NULL CONTAINS THE CHARACTER TO BE USED AS THE FILLER CHARACTER.
                                        ;*NOTE2:        $FILLS CONTAINS THE NUMBER OF FILLER CHARACTERS REQUIRED.
                                        ;*NOTE3:        $FILLC CONTAINS THE CHARACTER TO FILL AFTER.
                                        ;*
                                        ;*CALL:
                                        ;*1) USING A TRAP INSTRUCTION
                                        ;*      TYPE    MESADR          ;;MESADR IS FIRST ADDRESS OF AN ASCIZ STRING
                                        ;*OR
                                        ;*      TYPE
                                        ;*      MESADR
                                        ;*

055272  105737  002030                  $TYPE:  TSTB    $TPFLG          ;;IS THERE A TERMINAL?
055276  100407                                  BMI     6$              ;BR IF NO
055300  010046                          1$:     MOV     R0,-(SP)        ;;SAVE R0
055302  017600  000002                          MOV     @2(SP),R0       ;;GET ADDRESS OF ASCIZ STRING
055306  112046                          4$:     MOVB    (R0)+,-(SP)     ;;PUSH CHARACTER TO BE TYPED ONTO STACK
055310  001005                                  BNE     7$              ;;BR IF IT ISN'T THE TERMINATOR
055312  005726                                  TST     (SP)+           ;;IF TERMINATOR POP IT OFF THE STACK
055314  012600                          5$:     MOV     (SP)+,R0        ;;RESTORE R0
055316  062716  000002                  6$:     ADD     #2,(SP)         ;;ADJUST RETURN PC
055322  000002                                  RTI                     ;;RETURN
055324  122716  000011                  7$:     CMPB    #HT,(SP)        ;BRANCH IF NOT <HT>
055330  001002                                  BNE     11$
055332  112716  000040                          MOVB    #' ,(SP)        ;REPLACE TAB WITH SPACE
055336  122716  000200                  11$:    CMPB    #CRLF,(SP)      ;;BRANCH IF NOT <CRLF>
055342  001006                                  BNE     8$
055344  005726                                  TST     (SP)+           ;;POP  <CR><LF> EQUIV
055346                                          TYPE                    ;;TYPE A CR AND LF
 (1)    055346  104401                          TYPEIT
055350  002222                                  $CRLF
055352  105037  055462                          CLRB    $CHARCNT        ;;CLEAR CHARACTER COUNT
055356  000753                                  BR      4$              ;;GET NEXT CHARACTER
055360  004737  055416                  8$:     CALL    $TYPEC          ;;GO TYPE THIS CHARACTER
055364  123726  002214                  9$:     CMPB    $FILLC,(SP)+    ;;IS IT TIME FOR FILLER CHARS.?
055370  001346                                  BNE     4$              ;;IF NO GO GET NEXT CHAR.
055372  013746  002026                          MOV     $NULL,-(SP)     ;;GET # OF FILLER CHARS. NEEDED
                                                                        ;;AND THE NULL CHAR.
055376  105366  000001                  10$:    DECB    1(SP)           ;;DOES A NULL NEED TO BE TYPED?
055402  002770                                  BLT     9$              ;;BR IF NO--GO POP THE NULL OFF OF STACK
055404  004737  055416                          CALL    $TYPEC          ;;GO TYPE A NULL
055410  105337  055462                          DECB    $CHARCNT        ;;DO NOT COUNT AS A COUNT
055414  000770                                  BR      10$             ;;LOOP
055416  105777  124566                  $TYPEC: TSTB    @$TPS           ;;WAIT UNTIL PRINTER IS READY
055422  100375                                  BPL     $TYPEC
055424  116677  000002  124560                  MOVB    2(SP),@$TPB     ;;LOAD CHAR TO BE TYPED INTO DATA REG.
055432  122766  000015  000002                  CMPB    #CR,2(SP)       ;;IS CHARACTER A CARRIAGE RETURN?
055440  001003                                  BNE     1$              ;;BRANCH IF NO
055442  105037  055462                          CLRB    $CHARCNT        ;;YES--CLEAR CHARACTER COUNT
```

```
 8244  055446  000406                        BR      $TYPEX          ;;EXIT
 8245  055450  122766  000012  000002  1$:   CMPB    #LF,2(SP)       ;;IS CHARACTER A LINE FEED?
 8246  055456  001402                        BEQ     $TYPEX          ;;BRANCH IF YES
 8247  055460  105227                        INCB    (PC)+           ;;COUNT THE CHARACTER
 8248  055462  000000              $CHARCNT:.WORD  0               ;;CHARACTER COUNT STORAGE
 8249  055464  000207              $TYPEX: RETURN
```

```
                        .SBTTL  ROUTINE BINARY TO OCTAL (ASCII) AND TYPE

                        ;*************************************************************************
                        ;*THIS ROUTINE IS USED TO CHANGE A 16-BIT BINARY NUMBER TO A 6-DIGIT
                        ;*OCTAL (ASCII) NUMBER AND TYPE IT.
                        ;*$TYPOS---ENTER HERE TO SETUP SUPPRESS ZEROS AND NUMBER OF DIGITS TO TYPE
                        ;*CALL:
                        ;*      MOV     NUM,-(SP)       ;;NUMBER TO BE TYPED
                        ;*      TYPOS                   ;;CALL FOR TYPEOUT
                        ;*      .BYTE   N               ;;N=1 TO 6 FOR NUMBER OF DIGITS TO TYPE
                        ;*      .BYTE   M               ;;M=1 OR 0
                        ;*                                      ;;1=TYPE LEADING ZEROS
                        ;*                                      ;;0=SUPPRESS LEADING ZEROS
                        ;*
                        ;*$TYPON----ENTER HERE TO TYPE OUT WITH THE SAME PARAMETERS AS THE LAST
                        ;*$TYPOS OR $TYPOC
                        ;*CALL:
                        ;*      MOV     NUM,-(SP)       ;;NUMBER TO BE TYPED
                        ;*      TYPON                   ;;CALL FOR TYPEOUT
                        ;*
                        ;*$TYPOC---ENTER HERE FOR TYPEOUT OF A 16 BIT NUMBER
                        ;*CALL:
                        ;*      MOV     NUM,-(SP)       ;;NUMBER TO BE TYPED
                        ;*      TYPOC                   ;;CALL FOR TYPEOUT
                        
055466  017646  000000          $TYPOS: MOV     @(SP),-(SP)     ;;PICKUP THE MODE
055472  116637  000001  055711          MOVB    1(SP),$OFILL    ;;LOAD ZERO FILL SWITCH
055500  112637  055713          MOVB    (SP)+,$OMODE+1  ;;NUMBER OF DIGITS TO TYPE
055504  062716  000002          ADD     #2,(SP)         ;;ADJUST RETURN ADDRESS
055510  000406                  BR      $TYPON
055512  112737  000001  055711  $TYPOC: MOVB    #1,$OFILL       ;;SET THE ZERO FILL SWITCH
055520  112737  000006  055713          MOVB    #6,$OMODE+1     ;;SET FOR SIX(6) DIGITS
055526  112737  000005  055710  $TYPON: MOVB    #5,$OCNT        ;;SET THE ITERATION COUNT
055534  010346                  MOV     R3,-(SP)        ;;SAVE R3
055536  010446                  MOV     R4,-(SP)        ;;SAVE R4
055540  010546                  MOV     R5,-(SP)        ;;SAVE R5
055542  113704  055713          MOVB    $OMODE+1,R4     ;;GET THE NUMBER OF DIGITS TO TYPE
055546  005404                  NEG     R4
055550  062704  000006          ADD     #6,R4           ;;SUBTRACT IT FOR MAX. ALLOWED
055554  110437  055712          MOVB    R4,$OMODE       ;;SAVE IT FOR USE
055560  113704  055711          MOVB    $OFILL,R4       ;;GET THE ZERO FILL SWITCH
055564  016605  000012          MOV     12(SP),R5       ;;PICKUP THE INPUT NUMBER
055570  005003                  CLR     R3              ;;CLEAR THE OUTPUT WORD
055572  006105          1$:     ROL     R5              ;;ROTATE MSB INTO "C"
055574  000404                  BR      3$              ;;GO DO MSB
055576  006105          2$:     ROL     R5              ;;FORM THIS DIGIT
055600  006105                  ROL     R5
055602  006105                  ROL     R5
055604  010503                  MOV     R5,R3
055606  006103          3$:     ROL     R3              ;;GET LSB OF THIS DIGIT
055610  105337  055712          DECB    $OMODE          ;;TYPE THIS DIGIT?
055614  100016                  BPL     6$              ;;BR IF NO
055616  042703  177770          BIC     #177770,R3      ;;GET RID OF JUNK
055622  001002                  BNE     4$              ;;TEST FOR 0
055624  005704                  TST     R4              ;;SUPPRESS THIS 0?
055626  001403                  BEQ     5$              ;;BR IF YES
```

```
 8307   055630  005204                  4$:     INC     R4              ;;DON'T SUPPRESS ANYMORE 0'S
 8308   055632  052703  000060                  BIS     #'0,R3          ;;MAKE THIS DIGIT ASCII
 8309   055636  052703  000040          5$:     BIS     #' ,R3          ;;MAKE ASCII IF NOT ALREADY
 8310   055642  110337  055706                  MOVB    R3,8$           ;;SAVE FOR TYPING
 8311   055646                                  TYPE    8$              ;;GO TYPE THIS DIGIT
  (1)   055646  104401  055706                  TYPEIT  ,8$
 8312   055652  105337  055710          6$:     DECB    $OCNT           ;;COUNT BY 1
 8313   055656  003347                          BGT     2$              ;;BR IF MORE TO DO
 8314   055660  002402                          BLT     7$              ;;BR IF DONE
 8315   055662  005204                          INC     R4              ;;INSURE LAST DIGIT ISN'T A BLANK
 8316   055664  000744                          BR      2$              ;;GO DO THE LAST DIGIT
 8317   055666  012605          7$:             MOV     (SP)+,R5        ;;RESTORE R5
 8318   055670  012604                          MOV     (SP)+,R4        ;;RESTORE R4
 8319   055672  012603                          MOV     (SP)+,R3        ;;RESTORE R3
 8320   055674  016666  000002  000004          MOV     2(SP),4(SP)     ;;SET THE STACK FOR RETURNING
 8321   055702  012616                          MOV     (SP)+,(SP)
 8322   055704  000002                          RTI                     ;;RETURN
 8323   055706     000          8$:             .BYTE   0               ;;STORAGE FOR ASCII DIGIT
 8324   055707     000                          .BYTE   0               ;;TERMINATOR FOR TYPE ROUTINE
 8325   055710     000          $OCNT:  .BYTE   0               ;;OCTAL DIGIT COUNTER
 8326   055711     000          $OFILL: .BYTE   0               ;;ZERO FILL SWITCH
 8327   055712  000000          $OMODE: .WORD   0               ;;NUMBER OF DIGITS TO TYPE
```

```
                        .SBTTL  ROUTINE CONVERT BINARY TO DECIMAL AND TYPE
                        ;*******************************************************************************
                        ;*THIS ROUTINE IS USED TO CHANGE A 16-BIT BINARY NUMBER TO A 5-DIGIT
                        ;*SIGNED DECIMAL (ASCII) NUMBER AND TYPE IT. DEPENDING ON WHETHER THE
                        ;*NUMBER IS POSITIVE OR NEGATIVE A SPACE OR A MINUS SIGN WILL BE TYPED
                        ;*BEFORE THE FIRST DIGIT OF THE NUMBER. LEADING ZEROS WILL ALWAYS BE
                        ;*REPLACED WITH SPACES.
                        ;*CALL:
                        ;*      MOV     NUM,-(SP)       ;;PUT THE BINARY NUMBER ON THE STACK
                        ;*      TYPDS                   ;;GO TO THE ROUTINE
055714                  $TYPDS: PUSH    R0,R1,R2,R3,R5
(2) 055714  010046                                                      MOV R0,-(SP)
(3) 055716  010146                                                      MOV R1,-(SP)
(4) 055720  010246                                                      MOV R2,-(SP)
(5) 055722  010346                                                      MOV R3,-(SP)
(6) 055724  010546                                                      MOV R5,-(SP)
055726  012746  020200                MOV     #20200,-(SP)    ;;SET BLANK SWITCH AND SIGN
055732  016605  000020                MOV     20(SP),R5       ;;GET THE INPUT NUMBER
055736  100004                        BPL     1$              ;;BR IF INPUT IS POS.
055740  005405                        NEG     R5              ;;MAKE THE BINARY NUMBER POS.
055742  112766  000055  000001        MOVB    #'-,1(SP)       ;;MAKE THE ASCII NUMBER NEG.
055750  005000                1$:     CLR     R0              ;;ZERO THE CONSTANTS INDEX
055752  012703  056130                MOV     #$DBLK,R3       ;;SETUP THE OUTPUT POINTER
055756  112723  000040                MOVB    #' ,(R3)+       ;;SET THE FIRST CHARACTER TO A BLANK
055762  005002                2$:     CLR     R2              ;;CLEAR THE BCD NUMBER
055764  016001  056120                MOV     $DTBL(R0),R1    ;;GET THE CONSTANT
055770  160105                3$:     SUB     R1,R5           ;;FORM THIS BCD DIGIT
055772  002402                        BLT     4$              ;;BR IF DONE
055774  005202                        INC     R2              ;;INCREASE THE BCD DIGIT BY 1
055776  000774                        BR      3$
056000  060105                4$:     ADD     R1,R5           ;;ADD BACK THE CONSTANT
056002  005702                        TST     R2              ;;CHECK IF BCD DIGIT=0
056004  001002                        BNE     5$              ;;FALL THROUGH IF 0
056006  105716                        TSTB    (SP)            ;;STILL DOING LEADING 0'S?
056010  100407                        BMI     7$              ;;BR IF YES
056012  106316                5$:     ASLB    (SP)            ;;MSD?
056014  103003                        BCC     6$              ;;BR IF NO
056016  116663  000001  177777        MOVB    1(SP),-1(R3)    ;;YES--SET THE SIGN
056024  052702  000060                6$:     BIS     #'0,R2          ;;MAKE THE BCD DIGIT ASCII
056030  052702  000040                7$:     BIS     #' ,R2          ;;MAKE IT A SPACE IF NOT ALREADY A DIGIT
056034  110223                        MOVB    R2,(R3)+        ;;PUT THIS CHARACTER IN THE OUTPUT BUFFER
056036  005720                        TST     (R0)+           ;;JUST INCREMENTING
056040  020027  000010                CMP     R0,#10          ;;CHECK THE TABLE INDEX
056044  002746                        BLT     2$              ;;GO DO THE NEXT DIGIT
056046  003002                        BGT     8$              ;;GO TO EXIT
056050  010502                        MOV     R5,R2           ;;GET THE LSD
056052  000764                        BR      6$              ;;GO CHANGE TO ASCII
056054  105726                8$:     TSTB    (SP)+           ;;WAS THE LSD THE FIRST NON-ZERO?
056056  100003                        BPL     9$              ;;BR IF NO
056060  116663  177777  177776        MOVB    -1(SP),-2(R3)   ;;YES--SET THE SIGN FOR TYPING
056066  105013                9$:     CLRB    (R3)            ;;SET THE TERMINATOR
056070                          POP     R5,R3,R2,R1,R0
(2) 056070  012605                                                      MOV (SP)+,R5
(3) 056072  012603                                                      MOV (SP)+,R3
(4) 056074  012602                                                      MOV (SP)+,R2
(5) 056076  012601                                                      MOV (SP)+,R1
```

```
  (6)  056100  012600                                                                  MOV (SP)+,R0
 8376  056102                                  TYPE    $DBLK            ;;NOW TYPE THE NUMBER
  (1)  056102  104401  056130                  TYPEIT  ,$DBLK
 8377  056106  016666  000002  000004          MOV     2(SP),4(SP)      ;;ADJUST THE STACK
 8378  056114  012616                          MOV     (SP)+,(SP)
 8379  056116  000002                          RTI                      ;;RETURN TO USER
 8380  056120  023420                  $DTBL:  10000.
 8381  056122  001750                          1000.
 8382  056124  000144                          100.
 8383  056126  000012                          10.
 8384  056130  000004                  $DBLK:  .BLKW   4
```

```
                                        .SBTTL  ROUTINE TTY INPUT
                                ;*******************************************************************************
                                ;*SOFTWARE SWITCH REGISTER CHANGE ROUTINE.
                                ;*ROUTINE IS ENTERED FROM THE TRAP HANDLER, AND WILL
                                ;*SERVICE THE TEST FOR CHANGE IN SOFTWARE SWITCH REGISTER TRAP CALL
                                ;*WHEN OPERATING IN TTY FLAG MODE.
                                        .ENABLE LSB
056140  105777  124040          $CKSWR: TSTB    @$TKS           ;;CHAR THERE?
056144  100136                          BPL     12$             ;;IF NO, DON'T WAIT AROUND
056146  117746  124034                  MOVB    @$TKB,-(SP)     ;;SAVE THE CHAR
056152  042716  177600                  BIC     #^C177,(SP)     ;;STRIP-OFF THE ASCII
056156  022716  000006                  CMP     #6,(SP)         ;IS IT CONTROL F?
056162  001002                          BNE     1$              ;NO SKIP
056164  004737  037372                  CALL    FIELDSERVICE
056170  022716  000024          1$:     CMP     #24,(SP)        ;IS IT CONTROL T?
056174  001002                          BNE     16$             ;NO - SKIP
056176  004737  056532                  CALL    CONTT           ;YES - GO TO CONTROL T ROUTINE
056202  022716  000003          16$:    CMP     #3,(SP)         ;IS IT CONTROL C?
056206  001444                          BEQ     5$              ;YES EXIT *****NOTE***** STACK IS SCREWED UP!
056210  022716  000004                  CMP     #4,(SP)         ;IS IT CONTROL D?
056214  001002                          BNE     2$              ;NO - SKIP
056216  000137  072726          CONTP:  JMP     @#0.ODT         ;YES - GO TO ODT
056222  022737  000176  002200  2$:     CMP     #SWREG,SWR      ;;IS THE SOFT-SWR SELECTED?
056230  001105                          BNE     CKEND           ;;BRANCH IF NO
056232  022726  000007                  CMP     #7,(SP)+        ;;IS IT A CONTROL G?
056236  001102                          BNE     CKEND           ;;NO, RETURN TO USER
056240  005737  001450                  TST     $AUTO           ;ARE WE RUNNING IN AUTO-MODE?
056244  001077                          BNE     CKEND           ;BRANCH IF YES
056246                                  TYPE    $CNTLG          ;;ECHO THE CONTROL-G (^G)
 (1)    056246  104401  057266          TYPEIT  ,$CNTLG
056252                          $GTSWR: TYPE    $MSWR           ;;TYPE CURRENT CONTENTS
 (1)    056252  104401  057273          TYPEIT  ,$MSWR
056256                                  TYPOCT  @SWR            ;;OF THE SWR
 (1)    056256  017746  123716          MOV     @SWR,-(SP)      ;;SAVE @SWR FOR TYPEOUT
 (1)    056262  104402                  TYPOC                   ;;GO TYPE--OCTAL ASCII(ALL DIGITS)
056264                                  TYPE    $MNEW           ;;PROMPT FOR NEW SWR
 (1)    056264  104401  057304          TYPEIT  ,$MNEW
056270  005046                  3$:     CLR     -(SP)           ;;CLEAR COUNTER
056272  005046                          CLR     -(SP)           ;;THE NEW SWR
056274  105777  123704          4$:     TSTB    @$TKS           ;;CHAR THERE?
056300  100375                          BPL     4$              ;;IF NOT TRY AGAIN
056302  117746  123700                  MOVB    @$TKB,-(SP)     ;;PICK UP CHAR
056306  042716  177600                  BIC     #^C177,(SP)     ;;MAKE IT 7-BIT ASCII
056312  021627  000003                  CMP     (SP),#3         ;;IS IT A CONTROL-C?
056316  001015                          BNE     7$              ;;BRANCH IF NOT
056320                          5$:     TYPE    $CNTLC          ;;YES, ECHO CONTROL-C (^C)
 (1)    056320  104401  057254          TYPEIT  ,$CNTLC
056324  062706  000006                  ADD     #6,SP           ;;CLEAN UP STACK
056330  123727  001452  000001          CMPB    $INTAG,#1       ;;REENABLE TTY KEYBOARD INTERRUPTS?
056336  001003                          BNE     6$              ;;BRANCH IF NO
056340  012777  000100  123636          MOV     #100,@$TKS      ;;ALLOW TTY KEYBOARD INTERRUPTS
056346  000137  037020          6$:     JMP     BOOT            ;;CONTROL-C RESTART
056352  021627  000025          7$:     CMP     (SP),#25        ;;IS IT A CONTROL-U?
056356  001005                          BNE     9$              ;;BRANCH IF NOT
056360                                  TYPE    $CNTLU          ;;YES, ECHO CONTROL-U (^U)
 (1)    056360  104401  057261          TYPEIT  ,$CNTLU
```

SEQ 0365

```
                                ROUTINE TTY INPUT

 8435  056364  062706  000006        8$:     ADD     #6,SP           ;;IGNORE PREVIOUS INPUT
 8436  056370  000737                        BR      3$              ;;LET'S TRY IT AGAIN
 8437  056372  021627  000015        9$:     CMP     (SP),#15        ;;IS IT A <CR>?
 8438  056376  001025                        BNE     13$             ;;BRANCH IF NO
 8439  056400  005766  000004                TST     4(SP)           ;;YES, IS IT THE FIRST CHAR?
 8440  056404  001403                        BEQ     10$             ;;BRANCH IF YES
 8441  056406  016677  000002  123564        MOV     2(SP),@SWR      ;;SAVE NEW SWR
 8442  056414  062706  000006        10$:    ADD     #6,SP           ;;CLEAR UP STACK
 8443  056420                        11$:    TYPE    $CRLF           ;;ECHO <CR> AND <LF>
  (1)  056420  104401  002222                TYPEIT  ,$CRLF
 8444  056424  123727  001452  000001        CMPB    $INTAG,#1       ;;RE-ENABLE TTY KBD INTERRUPTS?
 8445  056432  001003                        BNE     12$             ;;BRANCH IF NOT
 8446  056434  012777  000100  123542        MOV     #100,@$TKS      ;;RE-ENABLE TTY KBD INTERRUPTS
 8447  056442  000002        12$:    RTI                             ;;RETURN
 8448  056444  062706  000002        CKEND:  ADD     #2,SP           ;FIX STACK
 8449  056450  000002                        RTI                     ;RETURN
 8450  056452  004737  055416        13$:    CALL    $TYPEC          ;;ECHO CHAR
 8451  056456  021627  000060                CMP     (SP),#60        ;;CHAR < 0?
 8452  056462  002420                        BLT     15$             ;;BRANCH IF YES
 8453  056464  021627  000067                CMP     (SP),#67        ;;CHAR > 7?
 8454  056470  003015                        BGT     15$             ;;BRANCH IF YES
 8455  056472  042726  000060                BIC     #60,(SP)+       ;;STRIP-OFF ASCII
 8456  056476  005766  000002                TST     2(SP)           ;;IS THIS THE FIRST CHAR
 8457  056502  001403                        BEQ     14$             ;;BRANCH IF YES
 8458  056504  006316                        ASL     (SP)            ;;NO, SHIFT PRESENT
 8459  056506  006316                        ASL     (SP)            ;;    CHAR OVER TO MAKE
 8460  056510  006316                        ASL     (SP)            ;;    ROOM FOR NEW ONE.
 8461  056512  005266  000002        14$:    INC     2(SP)           ;;KEEP COUNT OF CHAR
 8462  056516  056616  177776                BIS     -2(SP),(SP)     ;;SET IN NEW CHAR
 8463  056522  000664                        BR      4$              ;;GET THE NEXT ONE
 8464  056524                        15$:    TYPE    $QUES           ;;TYPE ?<CR><LF>
  (1)  056524  104401  002221                TYPEIT  ,$QUES
 8465  056530  000715                        BR      8$              ;;SIMULATE CONTROL-U
 8466                                        .DSABL  LSB
 8467
 8468  056532                        CONTT:  SUBTST  <<CONTROL T>>
  (3)                           ;*************************************************************************
  (3)                           ;*SUBTEST       CONTROL T
  (2)                           ;*************************************************************************
 8469  056532                                PUSH    R0
  (2)  056532  010046                                                                   MOV R0,-(SP)
 8470  056534                                IF RLFLAG NE #0
  (4)  056534  005737  001516                                                           TST RLFLAG
  (6)  056540  001402                                                                   BEQ L547
 8471  056542                                  TYPE  MSG092          ;RELOCATED
  (1)  056542  104401  072346                TYPEIT  ,MSG092
 8472  056546                                END ;OF IF RLFLAG
  (3)  056546                                                                   L547:::::::
 8473  056546                                TYPE    MSG093          ;BANK=
  (1)  056546  104401  072362                TYPEIT  ,MSG093
 8474  056552                                TYPOCS  BANK,,3         ;TYPE 3 DIGITS
  (1)  056552  013746  001470                MOV     BANK,-(SP)      ;;SAVE BANK FOR TYPEOUT
  (1)  056556  104403                        TYPOS                   ;;GO TYPE--OCTAL ASCII
  (1)  056560     003                        .BYTE   3               ;;TYPE 3 DIGIT(S)
  (1)  056561     000                        .BYTE   0               ;;SUPPRESS LEADING ZEROS
 8475  056562                                TYPE    MSG094          ;MAR=
```

```
  (1)  056562  104401  072370                TYPEIT  ,MSG094
 8476  056566  013700  177750                MOV     MAINT,R0
 8477  056572  006200                        ASR     R0
 8478  056574  042700  177770                BIC     #↑C7,R0
 8479  056600                                TYPOCS  R0,,1           ;TYPE 1 DIGIT
  (1)  056600  010046                        MOV     R0,-(SP)        ;;SAVE R0 FOR TYPEOUT
  (1)  056602  104403                        TYPOS                   ;;GO TYPE--OCTAL ASCII
  (1)  056604     001                        .BYTE   1               ;;TYPE 1 DIGIT(S)
  (1)  056605     000                        .BYTE   0               ;;SUPPRESS LEADING ZEROS
 8480  056606                                TYPE    MSG095          ;PAT=
  (1)  056606  104401  072377                TYPEIT  ,MSG095
 8481  056612                                TYPOCS  REALPAT,,2      ;TYPE 2 DIGITS
  (1)  056612  013746  001744                MOV     REALPAT,-(SP)   ;;SAVE REALPAT FOR TYPEOUT
  (1)  056616  104403                        TYPOS                   ;;GO TYPE--OCTAL ASCII
  (1)  056620     002                        .BYTE   2               ;;TYPE 2 DIGIT(S)
  (1)  056621     000                        .BYTE   0               ;;SUPPRESS LEADING ZEROS
 8482  056622                                TYPE    $CRLF
  (1)  056622  104401  002222                TYPEIT  ,$CRLF
 8483  056626                                POP     R0
  (2)  056626  012600                                                                    MOV (SP)+,R0
 8484  056630  000207                        RETURN
```

```
                                        ;*******************************************************************************
                                        ;*THIS ROUTINE WILL INPUT A SINGLE CHARACTER FROM THE TTY
                                        ;*CALL:
                                        ;*      RDCHR                       ;;INPUT A SINGLE CHARACTER FROM THE TTY
                                        ;*      RETURN HERE                 ;;CHARACTER IS ON THE STACK
                                        ;*                                  ;;WITH PARITY BIT STRIPPED OFF
                                        ;
056632  011646                          $RDCHR: MOV     (SP),-(SP)          ;;PUSH DOWN THE PC
056634  016666  000004  000002                  MOV     4(SP),2(SP)         ;;SAVE THE PS
056642  105777  123336                  1$:     TSTB    @$TKS               ;;WAIT FOR
056646  100375                                  BPL     1$                  ;;A CHARACTER
056650  117766  123332  000004                  MOVB    @$TKB,4(SP)         ;;READ THE TTY
056656  042766  177600  000004                  BIC     #↑C<177>,4(SP)      ;;GET RID OF JUNK IF ANY
056664  026627  000004  000023                  CMP     4(SP),#23           ;;IS IT A CONTROL-S?
056672  001013                                  BNE     3$                  ;;BRANCH IF NO
056674  105777  123304                  2$:     TSTB    @$TKS               ;;WAIT FOR A CHARACTER
056700  100375                                  BPL     2$                  ;;LOOP UNTIL ITS THERE
056702  117746  123300                          MOVB    @$TKB,-(SP)         ;;GET CHARACTER
056706  042716  177600                          BIC     #↑C177,(SP)         ;;MAKE IT 7-BIT ASCII
056712  022627  000021                          CMP     (SP)+,#21           ;;IS IT A CONTROL-Q?
056716  001366                                  BNE     2$                  ;;IF NOT DISCARD IT
056720  000750                                  BR      1$                  ;;YES, RESUME
056722  026627  000004  000140          3$:     CMP     4(SP),#140          ;;IS IT UPPER CASE?
056730  002407                                  BLT     4$                  ;;BRANCH IF YES
056732  026627  000004  000175                  CMP     4(SP),#175          ;;IS IT A SPECIAL CHAR?
056740  003003                                  BGT     4$                  ;;BRANCH IF YES
056742  042766  000040  000004                  BIC     #40,4(SP)           ;;MAKE IT UPPER CASE
056750  000002                          4$:     RTI                         ;;GO BACK TO USER
                                        ;*******************************************************************************
                                        ;*THIS ROUTINE WILL INPUT A STRING FROM THE TTY
                                        ;*CALL:
                                        ;*      RDLIN                       ;;INPUT A STRING FROM THE TTY
                                        ;*      RETURN HERE                 ;;ADDRESS OF FIRST CHARACTER WILL BE ON THE STACK
                                        ;*                                  ;;TERMINATOR WILL BE A BYTE OF ALL 0'S
056752  010346                          $RDLIN: MOV     R3,-(SP)            ;;SAVE R3
056754  005046                                  CLR     -(SP)               ;;CLEAR THE RUBOUT KEY
056756  012703  057230                  1$:     MOV     #$TTYIN,R3          ;;GET ADDRESS
056762  022703  057254                  2$:     CMP     #$TTYIN+20.,R3      ;;BUFFER FULL?
056766  101467                                  BLOS    8$                  ;;BR IF YES
056770  104411                                  RDCHR                       ;;GO READ ONE CHARACTER FROM THE TTY
056772  112613                                  MOVB    (SP)+,(R3)          ;;GET CHARACTER
056774  122713  000003                          CMPB    #3,(R3)             ;;IS IT A CONTROL-C?
057000  001006                                  BNE     3$                  ;;BRANCH IF NO
057002                                          TYPE    $CNTLC              ;;TYPE A CONTROL-C (↑C)
 (1)  057002  104401  057254                          TYPEIT  ,$CNTLC
057006  005726                                  TST     (SP)+               ;;CLEAN RUBOUT KEY OFF OF THE STACK
057010  012603                                  MOV     (SP)+,R3            ;;RESTORE R3
057012  000137  037020                          JMP     BOOT                ;;GOTO CONTROL-C RESTART
057016  122713  000177                  3$:     CMPB    #177,(R3)           ;;IS IT A RUBOUT
057022  001022                                  BNE     5$                  ;;BR IF NO
057024  005716                                  TST     (SP)                ;;IS THIS THE FIRST RUBOUT?
057026  001007                                  BNE     4$                  ;;BR IF NO
057030  112737  000134  057226                  MOVB    #'\,10$             ;;TYPE A BACK SLASH
057036                                          TYPE    10$
 (1)  057036  104401  057226                          TYPEIT  ,10$
```

```
 8540  057042  012716  177777            4$:   MOV     #-1,(SP)        ;;SET THE RUBOUT KEY
 8541  057046  005303                      DEC     R3              ;;BACKUP BY ONE
 8542  057050  020327  057230              CMP     R3,#$TTYIN      ;;STACK EMPTY?
 8543  057054  103434                      BLO     8$              ;;BR IF YES
 8544  057056  111337  057226              MOVB    (R3),10$                ;;SETUP TO TYPEOUT THE DELETED CHAR.
 8545  057062                              TYPE    10$             ;;GO TYPE
   (1) 057062  104401  057226              TYPEIT  ,10$
 8546  057066  000735                      BR      2$              ;;GO READ ANOTHER CHAR.
 8547  057070  005716                5$:   TST     (SP)            ;;RUBOUT KEY SET?
 8548  057072  001406                      BEQ     6$              ;;BR IF NO
 8549  057074  112737  000134  057226      MOVB    #'\,10$         ;;TYPE A BACK SLASH
 8550  057102                              TYPE    10$
   (1) 057102  104401  057226              TYPEIT  ,10$
 8551  057106  005016                      CLR     (SP)            ;;CLEAR THE RUBOUT KEY
 8552  057110  122713  000025          6$: CMPB    #25,(R3)        ;;IS CHARACTER A CTRL U?
 8553  057114  001003                      BNE     7$              ;;BR IF NO
 8554  057116                              TYPE    $CNTLU          ;;TYPE A CONTROL "U"
   (1) 057116  104401  057261              TYPEIT  ,$CNTLU
 8555  057122  000715                      BR      1$              ;;GO START OVER
 8556  057124  122713  000022          7$: CMPB    #22,(R3)        ;;IS CHARACTER A "↑R"?
 8557  057130  001011                      BNE     9$              ;;BRANCH IF NO
 8558  057132  105013                      CLRB    (R3)            ;;CLEAR THE CHARACTER
 8559  057134                              TYPE    $CRLF           ;;TYPE A "CR" & "LF"
   (1) 057134  104401  002222              TYPEIT  ,$CRLF
 8560  057140                              TYPE    $TTYIN          ;;TYPE THE INPUT STRING
   (1) 057140  104401  057230              TYPEIT  ,$TTYIN
 8561  057144  000706                      BR      2$              ;;GO PICKUP ANOTHER CHACTER
 8562  057146                          8$: TYPE    $QUES           ;;TYPE A '?'
   (1) 057146  104401  002221              TYPEIT  ,$QUES
 8563  057152  000701                      BR      1$              ;;CLEAR THE BUFFER AND LOOP
 8564  057154  111337  057226          9$: MOVB    (R3),10$                ;;ECHO THE CHARACTER
 8565  057160                              TYPE    10$
   (1) 057160  104401  057226              TYPEIT  ,10$
 8566  057164  122723  000015              CMPB    #15,(R3)+       ;;CHECK FOR RETURN
 8567  057170  001274                      BNE     2$              ;;LOOP IF NOT RETURN
 8568  057172  105063  177777              CLRB    -1(R3)          ;;CLEAR RETURN (THE 15)
 8569  057176                              TYPE    $LF             ;;TYPE A LINE FEED
   (1) 057176  104401  002223              TYPEIT  ,$LF
 8570  057202  005726                      TST     (SP)+           ;;CLEAN RUBOUT KEY FROM THE STACK
 8571  057204  012603                      MOV     (SP)+,R3        ;;RESTORE R3
 8572  057206  011646                      MOV     (SP),-(SP)      ;;ADJUST THE STACK AND PUT ADDRESS OF THE
 8573  057210  016666  000004  000002      MOV     4(SP),2(SP)     ;;      FIRST ASCII CHARACTER ON IT
 8574  057216  012766  057230  000004      MOV     #$TTYIN,4(SP)
 8575  057224  000002                      RTI                     ;;RETURN
 8576  057226     000                10$: .BYTE   0               ;;STORAGE FOR ASCII CHAR. TO TYPE
 8577  057227     000                     .BYTE   0               ;;TERMINATOR
 8578  057230  000024                $TTYIN: .BLKB  20.           ;;RESERVE SIZE BYTES FOR TTY INPUT
 8579  057254  041536  005015     000 $CNTLC: .ASCIZ  /↑C/<15><12>    ;;CONTROL "C"
 8580  057261     136  006525  000012 $CNTLU: .ASCIZ  /↑U/<15><12>    ;;CONTROL "U"
 8581  057266  043536  005015     000 $CNTLG: .ASCIZ  /↑G/<15><12>    ;;CONTROL "G"
 8582  057273     015  051412  051127 $MSWR:  .ASCIZ  <15><12>/SWR = /
       057300  036440  000040
 8583  057304  020040  042516  020127 $MNEW:  .ASCIZ  /  NEW = /
       057312  020075     000
 8584          057316                      .EVEN
```

```
                                        .SBTTL  ROUTINE READ AN OCTAL NUMBER FROM THE TTY
                                ;*******************************************************************************
                                ;*THIS ROUTINE WILL READ AN OCTAL (ASCII) NUMBER FROM THE TTY AND
                                ;*CHANGE IT TO BINARY.
                                ;*THE INPUT CHARACTERS WILL BE CHECKED TO INSURED THEY ARE LEGAL
                                ;*OCTAL DIGITS. IF AN ILLEGAL CHARACTER IS READ A "?" WILL BE TYPED
                                ;*FOLLOWED BY A CARRIAGE RETURN-LINE FEED. THE COMPLETE NUMBER MUST
                                ;*THEN BE RETYPED. THE INPUT IS TERMINATED BY TYPING A CARRIAGE RETURN.
                                ;*CALL:
                                ;*      RDOCT                   ;;READ AN OCTAL NUMBER
                                ;*      RETURN HERE             ;;LOW ORDER BITS ARE ON TOP OF THE STACK
                                ;*                              ;;HIGH ORDER BITS ARE IN $HIOCT
057316  011646                  $RDOCT: MOV     (SP),-(SP)      ;;PROVIDE SPACE FOR THE
057320  016666  000004  000002          MOV     4(SP),2(SP)     ;;INPUT NUMBER
057326                                  PUSH    R0,R1,R2
057326  010046                                                          MOV R0,-(SP)
057330  010146                                                          MOV R1,-(SP)
057332  010246                                                          MOV R2,-(SP)
057334  104412                  1$:     RDLIN                   ;;READ AN ASCIZ LINE
057336  012600                          MOV     (SP)+,R0        ;;GET ADDRESS OF 1ST CHARACTER
057340  010037  057444                  MOV     R0,5$           ;;AND SAVE IT
057344  005001                          CLR     R1              ;;CLEAR DATA WORD
057346  005002                          CLR     R2
057350  112046                  2$:     MOVB    (R0)+,-(SP)     ;;PICKUP THIS CHARACTER
057352  001420                          BEQ     3$              ;;IF ZERO GET OUT
057354  122716  000060                  CMPB    #'0,(SP)        ;;MAKE SURE THIS CHARACTER
057360  003026                          BGT     4$              ;;IS AN OCTAL DIGIT
057362  122716  000067                  CMPB    #'7,(SP)
057366  002423                          BLT     4$
057370  006301                          ASL     R1              ;;*2
057372  006102                          ROL     R2
057374  006301                          ASL     R1              ;;*4
057376  006102                          ROL     R2
057400  006301                          ASL     R1              ;;*8
057402  006102                          ROL     R2
057404  042716  177770                  BIC     #^C7,(SP)       ;;STRIP THE ASCII JUNK
057410  062601                          ADD     (SP)+,R1        ;;ADD IN THIS DIGIT
057412  000756                          BR      2$              ;;LOOP
057414  005726                  3$:     TST     (SP)+           ;;CLEAN TERMINATOR FROM STACK
057416  010166  000012                  MOV     R1,12(SP)       ;;SAVE THE RESULT
057422  010237  057464                  MOV     R2,$HIOCT
057426                                  POP     R2,R1,R0
057426  012602                                                          MOV (SP)+,R2
057430  012601                                                          MOV (SP)+,R1
057432  012600                                                          MOV (SP)+,R0
057434  000002                          RTI                     ;;RETURN
057436  005726                  4$:     TST     (SP)+           ;;CLEAN PARTIAL FROM STACK
057440  105010                          CLRB    (R0)            ;;SET A TERMINATOR
057442                                  TYPE                    ;;TYPE UP THRU THE BAD CHAR.
057442  104401                          TYPEIT
057444  000000                  5$:     .WORD   0
057446                                  TYPE    MSG062          ;INPUT MUST BE A
057446  104401  071400                  TYPEIT  ,MSG062
057452                                  TYPE    MSG063          ;N OCTAL
057452  104401  071420                  TYPEIT  ,MSG063
057456                                  TYPE    MSG064          ;NUMBER
```

```
 (1)  057456  104401  071431                  TYPEIT  ,MSG064
8633  057462  000724                          BR      1$                  ;;TRY AGAIN
8634  057464  000000                  $HIOCT: .WORD   0                   ;;HIGH ORDER BITS GO HERE
8635                                          .SBTTL  ROUTINE READ A DECIMAL NUMBER FROM THE TTY
8636
8637                                  ;*******************************************************************************
8638                                  ;*THIS ROUTINE WILL READ A DECIMAL (ASCII) NUMBER FROM THE TTY AND
8639                                  ;*CHANGE IT TO BINARY. IF TOO MANY CHARACTERS OR ANY ILLEGAL CHARACTERS
8640                                  ;*ARE READ A "?" FOLLOWED BY A CARRIAGE RETURN-LINE FEED WILL BE TYPED.
8641                                  ;*THE COMPLETE NUMBER MUST BE RETYPED. THE INPUT IS TERMINATED BY THE
8642                                  ;*USER TYPING A CARRIAGE RETURN. THE RANGE OF THE INPUT NUMBER IS
8643                                  ;*POSITIVE 32767 TO NEGATIVE 32768.
8644                                  ;*CALL:
8645                                  ;*      RDDEC                       ;;READ A DECIMAL NUMBER
8646                                  ;*      RETURN HERE                 ;;NUMBER IS ON TOP OF THE STACK
8647                                  ;
8648
8649  057466  011646                  $RDDEC: MOV     (SP),-(SP)          ;;PROVIDE SPACE FOR
8650  057470  016666  000004  000002          MOV     4(SP),2(SP)         ;;THE INPUT NUMBER
8651  057476                                  PUSH    R0,R1,R2
 (2)  057476  010046                                                                                  MOV R0,-(SP)
 (3)  057500  010146                                                                                  MOV R1,-(SP)
 (4)  057502  010246                                                                                  MOV R2,-(SP)
8652  057504  104412                  1$:     RDLIN                       ;;READ AN ASCIZ LINE
8653  057506  012600                          MOV     (SP)+,R0            ;;ADDRESS OF 1ST CHAR.
8654  057510  010037  057634                  MOV     R0,6$               ;;SAVE INCASE OF BAD INPUT
8655  057514  005046                          CLR     -(SP)               ;;CLEAR DATA WORD
8656  057516  005002                          CLR     R2                  ;;SIGN SET POSITIVE
8657  057520  122710  000055                  CMPB    #'-,(R0)            ;;SEE IF A MINUS SIGN WAS TYPED
8658  057524  001001                          BNE     2$                  ;;BR IF NO MINUS SIGN
8659  057526  112002                          MOVB    (R0)+,R2            ;;SAVE FOR LATER USE
8660  057530  112001                  2$:     MOVB    (R0)+,R1            ;;PICKUP THIS CHARACTER
8661  057532  001424                          BEQ     3$                  ;;GET OUT IF ZERO
8662  057534  122701  000060                  CMPB    #'0,R1              ;;MAKE SURE THIS CHARACTER
8663  057540  003032                          BGT     5$                  ;;IS A DIGIT BETWEEN 0 & 9
8664  057542  122701  000071                  CMPB    #'9,R1
8665  057546  002427                          BLT     5$
8666  057550  032716  170000                  BIT     #↑C7777,(SP)        ;;DON'T LET NUMBER GET TO BIG
8667  057554  001024                          BNE     5$                  ;;BR IF NUMBER WOULD OVERFLOW
8668  057556  006316                          ASL     (SP)                ;;*2
8669  057560  011646                          MOV     (SP),-(SP)          ;;SAVE FOR LATER
8670  057562  006316                          ASL     (SP)                ;;*4
8671  057564  006316                          ASL     (SP)                ;;*8.
8672  057566  062616                          ADD     (SP)+,(SP)          ;;*10.
8673  057570  102416                          BVS     5$                  ;;OVERFLOW ISN'T ALLOWED
8674  057572  162701  000060                  SUB     #'0,R1              ;;STRIP AWAY THE ASCII JUNK
8675  057576  060116                          ADD     R1,(SP)             ;;ADD IN THIS DIGIT
8676  057600  102412                          BVS     5$                  ;;OVERFLOW ISN'T ALLOWED
8677  057602  000752                          BR      2$                  ;;LOOP
8678  057604  005702                  3$:     TST     R2                  ;;CHECK IF NUMBER IS NEG
8679  057606  001401                          BEQ     4$                  ;;BR IF NO
8680  057610  005416                          NEG     (SP)                ;;YES--NEGATE THE NUMBER
8681  057612  012666  000012          4$:     MOV     (SP)+,12(SP)        ;;SAVE THE RESULT
8682  057616                                  POP     R2,R1,R0
 (2)  057616  012602                                                                                  MOV (SP)+,R2
 (3)  057620  012601                                                                                  MOV (SP)+,R1
```

```
                                                                                MOV (SP)+,R0
  (4)  057622  012600
8683   057624  000002                      RTI                 ;;RETURN
8684
8685   057626  005726              5$:     TST     (SP)+       ;;CLEAN PARTIAL NUMBER FROM STACK
8686   057630  105010                      CLRB    (R0)        ;;SET A TERMINATOR
8687   057632                              TYPE                ;;TYPE THE INPUT UP TO BAD CHAR.
  (1)  057632  104401                      TYPEIT
8688   057634  000000              6$:     .WORD   0           ;;POINTER GOES HERE
8689   057636                              TYPE    MSG062      ;INPUT MUSST BE A
  (1)  057636  104401  071400              TYPEIT  ,MSG062
8690   057642                              TYPE    MSG065      ;DECIMAL
  (1)  057642  104401  071441              TYPEIT  ,MSG065
8691   057646                              TYPE    MSG064      ;NUMBER
  (1)  057646  104401  071431              TYPEIT  ,MSG064
8692   057652  000714                      BR      1$          ;;TRY AGAIN
```

```
                                        .SBTTL  ROUTINE SAVE AND RESTORE R0-R5

                                ;*******************************************************************************
                                ;*SAVE R0-R5
                                ;*CALL:
                                ;*      SAVREG
                                ;*UPON RETURN FROM $SAVREG THE STACK WILL LOOK LIKE:
                                ;*
                                ;*TOP---(+16)
                                ;* +2---(+18)
                                ;* +4---R5
                                ;* +6---R4
                                ;* +8---R3
                                ;*+10---R2
                                ;*+12---R1
                                ;*+14---R0

057654                          $SAVREG:
057654                                  PUSH    R0,R1,R2,R3,R4,R5
  (2) 057654  010046                                                    MOV R0,-(SP)
  (3) 057656  010146                                                    MOV R1,-(SP)
  (4) 057660  010246                                                    MOV R2,-(SP)
  (5) 057662  010346                                                    MOV R3,-(SP)
  (6) 057664  010446                                                    MOV R4,-(SP)
  (7) 057666  010546                                                    MOV R5,-(SP)
057670  016646  000022                        MOV     22(SP),-(SP)    ;;SAVE PS OF MAIN FLOW
057674  016646  000022                        MOV     22(SP),-(SP)    ;;SAVE PC OF MAIN FLOW
057700  016646  000022                        MOV     22(SP),-(SP)    ;;SAVE PS OF CALL
057704  016646  000022                        MOV     22(SP),-(SP)    ;;SAVE PC OF CALL
057710  000002                              RTI

                                ;*RESTORE R0-R5
                                ;*CALL:
                                ;*      RESREG
057712                          $RESREG:
057712  012666  000022                MOV     (SP)+,22(SP)    ;;RESTORE PC OF CALL
057716  012666  000022                MOV     (SP)+,22(SP)    ;;RESTORE PS OF CALL
057722  012666  000022                MOV     (SP)+,22(SP)    ;;RESTORE PC OF MAIN FLOW
057726  012666  000022                MOV     (SP)+,22(SP)    ;;RESTORE PS OF MAIN FLOW
057732                                  POP     R5,R4,R3,R2,R1,R0
  (2) 057732  012605                                                    MOV (SP)+,R5
  (3) 057734  012604                                                    MOV (SP)+,R4
  (4) 057736  012603                                                    MOV (SP)+,R3
  (5) 057740  012602                                                    MOV (SP)+,R2
  (6) 057742  012601                                                    MOV (SP)+,R1
  (7) 057744  012600                                                    MOV (SP)+,R0
057746  000002                              RTI
```

```
                                .SBTTL  ROUTINE RANDOM NUMBER GENERATOR

                        ;*********************************************************************
                        ;*THIS ROUTINE IS A DOUBLE PRECISION PSEUDO RANDOM NUMBER GENERATOR
                        ;*WITH A RANGE OF 0 TO 2**(+33)-1.
                        ;*CALL:
                        ;*      CALL    $RAND           ;;CALL THE ROUTINE
                        ;*      RETURN                  ;;RETURN HERE THE RANDOM
                        ;*                              ;;NUMBER WILL BE IN
                        ;*                              ;;$HINUM,$LONUM

057750                  $RAND:  PUSH    R0,R1,R2
  (2) 057750  010046                                                    MOV R0,-(SP)
  (3) 057752  010146                                                    MOV R1,-(SP)
  (4) 057754  010246                                                    MOV R2,-(SP)
057756  013700  002144                MOV     SEEDLO,R0       ;SET R0 WITH LOW
057762  013701  002142                MOV     SEEDHI,R1       ;SET R1 WITH HIGH
057766  012702  000007                MOV     #7,R2           ;SET SHIFT COUNT
057772  006300                  1$:   ASL     R0              ;;SHIFT R0 LEFT AND
057774  006101                        ROL     R1              ;;ROTATE CARRY INTO R1 AND
057776  077203                        SOB     R2,1$
060000  063700  002144                ADD     SEEDLO,R0       ;ADD NUMBER TO MAKE X 129
060004  005501                        ADC     R1              ;;PROPOGATE CARRY
060006  063701  002142                ADD     SEEDHI,R1       ;ADD NUMBER TO MAKE X 129
060012  062700  001057                ADD     #1057,R0        ;;ADD LOW CONSTANT
060016  005501                        ADC     R1              ;;PROPOGATE CARRY
060020  062701  047401                ADD     #47401,R1       ;;ADD HIGH CONSTANT
060024  010037  002144                MOV     R0,SEEDLO       ;SAVE R0
060030  010137  002142                MOV     R1,SEEDHI       ;SAVE R1
060034                                POP     R2,R1,R0
  (2) 060034  012602                                                    MOV (SP)+,R2
  (3) 060036  012601                                                    MOV (SP)+,R1
  (4) 060040  012600                                                    MOV (SP)+,R0
060042  000207                        RETURN
```

```
                                .SBTTL  ROUTINE DOUBLE LENGTH BINARY TO OCTAL ASCII CONVERT
                        ;*******************************************************************************
                        ;*THIS ROUTINE WILL CONVERT A 32-BIT UNSIGNED BINARY NUMBER TO AN
                        ;*UNSIGNED OCTAL ASCIZ NUMBER.
                        ;*CALL
                        ;*      MOV     #PNTR,-(SP)     ;;POINTER TO LOW WORD OF BINARY NUMBER
                        ;*      CALL    $DB20           ;;CALL THE ROUTINE
                        ;*      RETURN                  ;;THE ADDRESS OF THE FIRST ASCIZ CHAR. IS ON THE STACK
                        
060044  104415          $DB20:  SAVREG                  ;;SAVE ALL REGISTERS
060046  016601  000002          MOV     2(SP),R1        ;;PICKUP THE POINTER TO LOW WORD
060052  012705  060163          MOV     #$OCTVL+13.,R5  ;;POINTER TO DATA TABLE
060056  012704  000014          MOV     #12.,R4         ;;DO ELEVEN CHARACTERS
060062  012703  177770          MOV     #^C7,R3         ;;MASK
060066  012100                  MOV     (R1)+,R0        ;;LOWER WORD
060070  012101                  MOV     (R1)+,R1        ;;HIGH WORD
060072  005002                  CLR     R2              ;;TERMINATOR
060074  110245          1$:     MOVB    R2,-(R5)        ;;PUT CHARACTER IN DATA TABLE
060076  010002                  MOV     R0,R2           ;;GET THIS DIGIT
060100  005304                  DEC     R4              ;;COUNT THIS CHARACTER
060102  003007                  BGT     3$              ;;BR IF NOT THE LAST DIGIT
060104  001405                  BEQ     2$              ;;BR IF IT IS THE LAST DIGIT
060106  005205                  INC     R5              ;;ALL DIGITS DONE-ADJUST POINTER FOR FIRST
060110  010566  000002          MOV     R5,2(SP)        ;;ASCIZ CHAR. & PUT IT ON THE STACK
060114  104416                  RESREG                  ;;RESTORE ALL REGISTERS
060116  000207                  RETURN                  ;;RETURN TO USER
060120  006203          2$:     ASR     R3              ;;POSITION THE MASK FOR THE LAST DIGIT
060122  006001          3$:     ROR     R1              ;;POSITION THE BINARY NUMBER FOR
060124  006000                  ROR     R0              ;;      THE NEXT OCTAL DIGIT
060126  006001                  ROR     R1
060130  006000                  ROR     R0
060132  006001                  ROR     R1
060134  006000                  ROR     R0
060136  040302                  BIC     R3,R2           ;;MASK OUT ALL JUNK
060140  062702  000060          ADD     #'0,R2          ;;MAKE THIS CHAR. ASCII
060144  000753                  BR      1$              ;;GO PUT IT IN THE DATA TABLE
060146  000016          $OCTVL: .BLKB   14.             ;;RESERVE DATA TABLE
060152                  $OCTB=$OCTVL+4                  ;POINTER TO 11 DIGIT NUMBER
```

```
                                        .TITLE  TABLES
                                        .SBTTL  APT MAILBOX-ETABLE
060164                          $MAIL:
060164  000000                  $MSGTY: .WORD   0       ;;MESSAGE TYPE CODE
060166  000000                  $FATAL: .WORD   0       ;;FATAL ERROR NUMBER (ERROR PC)
060170  000000                  $TESTN: .WORD   0       ;;TEST NUMBER (DISPLAY REGISTER)
060172  000000                  $PASS:  .WORD   0       ;;PASS COUNT
060174  000000                  $DEVCT: .WORD   0       ;;DEVICE COUNT
060176  000000                  $UNIT:  .WORD   0       ;;I/O UNIT NUMBER
060200  000000                  $MSGAD: .WORD   0       ;;MESSAGE ADDRESS
060202  000000                  $MSGLG: .WORD   0       ;;MESSAGE LENGTH
060204                          $ETABLE:                ;;APT ENVIRONMENT TABLE
060204      000                 $ENV:   .BYTE   0       ;;ENVIRONMENT BYTE              ;SET TO A 1 FOR APT AUTO MODE
060205      000                 $ENVM:  .BYTE   0       ;ENVIRONMENT MODE
                                                        ;BIT7(200)=APT SWR ;BIT5(40)=NO CONSOLE
060206  000001                  $SWREG: .WORD   1       ;;APT SWITCH REGISTER
060210  000000                  $USWR:  .WORD   0       ;USED TO LIMIT THE NUMBER OF PASSES
060212  030400                  $CPUOP: .WORD   30400   ;;CPU TYPE,OPTIONS
                                                        BITS 15-11=CPU TYPE
                                                        ;*                  11/04=01,11/05=02,11/20=03,11/40=04,11/45=05
                                                        ;*                  11/70=06,PDQ=07,Q=10
                                                        ;*              BIT 10=REAL TIME CLOCK
                                                        ;*              BIT  9=FLOATING POINT PROCESSOR
                                                        ;*              BIT  8=MEMORY MANAGEMENT
060214      001                 $MAMS1: .BYTE   1       ;;HIGH ADDRESS,M.S. BYTE ;DEFAULT = 64K
060215      004                 $MTYP1: .BYTE   4       ;;MEM. TYPE,BLK#1
                                                        MEM.TYPE BYTE   --  (HIGH BYTE)
                                                        ;*                  900 NSEC CORE=001
                                                        ;*                  300 NSEC BIPOLAR=002
                                                        ;*                  500 NSEC MOS=003
                                                        ;*                  MK11 ERROR CORRECTING MOS=004
060216  177776                  $MADR1: .WORD   177776  ;;HIGH ADDRESS,BLK#1
                                                        ;*  MEM.LAST ADDR.=3 BYTES,THIS WORD AND LOW OF "TYPE" ABOVE
060220      000                 $MAMS2: .BYTE   0       ;;HIGH ADDRESS,M.S. BYTE
060221      000                 $MTYP2: .BYTE   0       ;;MEM.TYPE,BLK#2
060222  000000                  $MADR2: .WORD   0       ;;MEM.LAST ADDRESS,BLK#2
060224      000                 $MAMS3: .BYTE   0       ;;HIGH ADDRESS,M.S.BYTE
060225      000                 $MTYP3: .BYTE   0       ;;MEM.TYPE,BLK#3
060226  000000                  $MADR3: .WORD   0       ;;MEM.LAST ADDRESS,BLK#3
060230      000                 $MAMS4: .BYTE   0       ;;HIGH ADDRESS,M.S.BYTE
060231      000                 $MTYP4: .BYTE   0       ;;MEM.TYPE,BLK#4
060232  000000                  $MADR4: .WORD   0       ;;MEM.LAST ADDRESS,BLK#4
060234  000000                  $VECT1: .WORD   0       ;;INTERRUPT VECTOR#1,BUS PRIORITY#1
060236  000000                  $VECT2: .WORD   0       ;;INTERRUPT VECTOR#2BUS PRIORITY#2
060240  000000                  $BASE:  .WORD   0       ;;BASE ADDRESS OF EQUIPMENT UNDER TEST
060242  000000                  $DEVM:  .WORD   0       ;;DEVICE MAP

                                ;IF EITHOR OF THESE 2 WORDS ARE NON ZERO
                                ;THEY WILL BE LOADED INTO ALL CSR'S & THE PROGRAM WILL STOP
060244  000000                  $CDW1:  .WORD   0       ;CSR
060246  000000                  $CDW2:  .WORD   0       ;CSR+2

                                ;THE FOLLOWING LOCATIONS SPECIFY WHICH PATTERNS
                                ;ARE TO BE RUN FOR PARTICULAR MEMORIES
                                ;
```

```
 8855                                ;REFERENCE THE TABLE LISTED BELOW TO RELATE BITS TO PATTERNS.
 8856                                ;BIT0 SET WILL RUN THE FIRST ENTRY IN THE TABLE, BIT0 SET
 8857                                ;IN THE SECOND WORD WILL RUN THE 17TH ENTRY IN THE TABLE ...
 8858                                ;
 8859                                ;NOTE** NULL TESTS DO NOT TAKE ANY TIME
 8860                                ;
 8861  060250  177777                $DDW0:  .WORD   177777  ;MK11 CSR TESTS          TABLE = MKCSRT:
 8862  060252  177777                $DDW1:  .WORD   177777  ;MK11 CSR TESTS          TABLE = MKCSRT:
 8863  060254  161377                $DDW2:  .WORD   161377  ;MK11 PATTERNS           TABLE = MKPAT:
 8864  060256  177777                $DDW3:  .WORD   177777  ;MK11 PATTERNS           TABLE = MKPAT:
 8865  060260  177677                $DDW4:  .WORD   177677  ;MJ11 PATTERNS           TABLE = MJPAT:
 8866  060262  177777                $DDW5:  .WORD   177777  ;MJ11 PATTERNS           TABLE = MJPAT:
 8867  060264                        $ETEND:
 8868                                ;SETUP APT PARAMETER BLOCK AS DEFINED IN THE APT-PDP11 DIAGNOSTIC
 8869                                ;INTERFACE SPEC.
 8870
 8871                                .???????????SOMEHOW FIGURE OUT WHAT TO SET THERE VARIABLES TO!
 8872  060264                        $APTHD:
 8873  060264  000000                $HIBTS: .WORD   0       ;;TWO HIGH BITS OF 18 BIT MAILBOX ADDR.
 8874  060266  060164                $MBADR: .WORD   $MAIL   ;;ADDRESS OF APT MAILBOX (BITS 0-15)
 8875  060270  000010                $TSTM:  .WORD   8.      ;;RUN TIM OF LONGEST TEST
 8876  060272  000144                $PASTM: .WORD   100.    ;;RUN TIME IN SECS. OF 1ST PASS ON 1 UNIT (QUICK VERIFY)
 8877  060274  000310                $UNITM: .WORD   200.    ;;ADDITIONAL RUN TIME (SECS) OF A PASS FOR EACH ADDITIONAL UNIT
 8878  060276  000040                        .WORD   $ETEND-$MAIL/2 ;;LENGTH MAILBOX-ETABLE(WORDS)
```

```
                                        .SBTTL  ROUTINE TRAP DECODER

                                ;*************************************************************************
                                ;*THIS ROUTINE WILL PICKUP THE LOWER BYTE OF THE "TRAP" INSTRUCTION
                                ;*AND USE IT TO INDEX THROUGH THE TRAP TABLE FOR THE STARTING ADDRESS
                                ;*OF THE DESIRED ROUTINE. THEN USING THE ADDRESS OBTAINED IT WILL
                                ;*GO TO THAT ROUTINE.

060300  010046                  $TRAP:  MOV     R0,-(SP)        ;;SAVE R0
060302  016600  000002                  MOV     2(SP),R0        ;;GET TRAP ADDRESS
060306  005740                          TST     -(R0)           ;;BACKUP BY 2
060310  111000                          MOVB    (R0),R0         ;;GET RIGHT BYTE OF TRAP
060312  006300                          ASL     R0              ;;POSITION FOR INDEXING
060314  016000  060334                  MOV     $TRPAD(R0),R0   ;;INDEX TO TABLE
060320  000200                          RTS     R0              ;;GO TO ROUTINE

                                ;;THIS IS USE TO HANDLE THE "GETPRI" MACRO

060322  011646                  $TRAP2: MOV     (SP),-(SP)      ;;MOVE THE PC DOWN
060324  016666  000004  000002          MOV     4(SP),2(SP)     ;;MOVE THE PSW DOWN
060332  000002                          RTI                     ;;RESTORE THE PSW

                                        .SBTTL  TRAP TABLE

                                ;*THIS TABLE CONTAINS THE STARTING ADDRESSES OF THE ROUTINES CALLED
                                ;*BY THE "TRAP" INSTRUCTION.

                                ;       ROUTINE
                                ;       -------
060334  060322                  $TRPAD: .WORD   $TRAP2
060336  055272                          $TYPE   ;CALL=TYPEIT    TRAP+1(104401)  TTY TYPEOUT ROUTINE
060340  055512                          $TYPOC  ;CALL=TYPOC     TRAP+2(104402)  TYPE OCTAL NUMBER (WITH LEADING ZEROS)
060342  055466                          $TYPOS  ;CALL=TYPOS     TRAP+3(104403)  TYPE OCTAL NUMBER (NO LEADING ZEROS)
060344  060600                          $NOTRAP;$TYPON  ;CALL=TYPON     TRAP+4(104404)  TYPE OCTAL NUMBER (AS PER LAST C
060346  055714                          $TYPDS  ;CALL=TYPDS     TRAP+5(104405)  TYPE DECIMAL NUMBER (WITH SIGN)
060350  060600                          $NOTRAP:$TYPBN  ;CALL=TYPBN     TRAP+6(104406)  TYPE BINARY (ASCII) NUMBER

060352  056252                          $GTSWR  ;CALL=GTSWR     TRAP+7(104407)  GET SOFT-SWR SETTING
060354  056140                          $CKSWR  ;CALL=CKSWR     TRAP+10(104410) TEST FOR CHANGE IN SOFT-SWR

060356  056632                          $RDCHR  ;CALL=RDCHR     TRAP+11(104411) TTY TYPEIN CHARACTER ROUTINE
060360  056752                          $RDLIN  ;CALL=RDLIN     TRAP+12(104412) TTY TYPEIN STRING ROUTINE
060362  057316                          $RDOCT  ;CALL=RDOCT     TRAP+13(104413) READ AN OCTAL NUMBER FROM TTY
060364  057466                          $RDDEC  ;CALL=RDDEC     TRAP+14(104414) READ A DECIMAL NUMBER FROM TTY

060366  057654                          $SAVREG ;CALL=SAVREG    TRAP+15(104415) SAVE R0-R5 ROUTINE
060370  057712                          $RESREG ;CALL=RESREG    TRAP+16(104406) RESTORE R0-R5 ROUTINE

060372  032546                          $KERNEL  ;CALL=KERNEL   TRAP+17(104417) ENTER KERNEL MODE
060374  032556                          $ENERGIZE;CALL=ENERGIZETRAP+20(104420) TURN ON MEMORY MANAGEMENT & TRAPS
060376  032566                          $DEENERGI;CALL=DEENERGITRAP+21(104421) TURN OFF MEMORY MENAGEMENT & TRAPS

060400  034626                          $KMAP    ;CALL=KMAP     TRAP+22(104422) MAP KERNEL 1 TO 1

060402  032576                          $CACHN  ;CALL=CACHON    TRAP+23(104423) TURN CACHE ON
```

```
 8936  060404  032614                  $CACHF  ;CALL=CACHOFF   TRAP+24(104424) TURN CACHE OFF
 8937
 8938  060406  032624                  $LOADC  ;CALL=LOADCSR   TRAP+25(104425) LOAD CORRECT CSR
 8939  060410  033056                  $READC  ;CALL=READCSR   TRAP+26(104426) READ CORRECT CSR
 8940
 8941  060412  046646                  $PER01  ;CALL=PERR01    TRAP+27(104427) PROGRAM DETECTED ERROR
 8942  060414  046674                  $PER02  ;CALL=PERR02    TRAP+30(104430) PROGRAM DETECTED ERROR
 8943  060416  046722                  $PER03  ;CALL=PERR03    TRAP+31(104431) PROGRAM DETECTED ERROR
 8944  060420  046752                  $PER04  ;CALL=PERR04    TRAP+32(104432) PROGRAM DETECTED ERROR
 8945  060422  047034                  $PER07  ;CALL=PERR07    TRAP+33(104433) PROGRAM DETECTED ERROR
 8946  060424  047054                  $PER10  ;CALL=PERR10    TRAP+34(104434) PROGRAM DETECTED ERROR
 8947  060426  047102                  $PER11  ;CALL=PERR11    TRAP+35(104435) PROGRAM DETECTED ERROR
 8948  060430  047120                  $PER12  ;CALL=PERR12    TRAP+36(104436) PROGRAM DETECTED ERROR
 8949  060432  047140                  $PER13  ;CALL=PERR13    TRAP+37(104437) PROGRAM DETECTED ERROR
 8950  060434  047156                  $PER14  ;CALL=PERR14    TRAP+40(104440) PROGRAM DETECTED ERROR
 8951  060436  047176                  $PER15  ;CALL=PERR15    TRAP+41(104441) PROGRAM DETECTED ERROR
 8952  060440  047216                  $PER16  ;CALL=PERR16    TRAP+42(104442) PROGRAM DETECTED ERROR
 8953  060442  047236                  $PER17  ;CALL=PERR17    TRAP+43(104443) PROGRAM DETECTED ERROR
 8954  060444  047254                  $PER20  ;CALL=PERR20    TRAP+44(104444) PROGRAM DETECTED ERROR
 8955  060446  047272                  $PER21  ;CALL=PERR21    TRAP+45(104445) PROGRAM DETECTED ERROR
 8956  060450  047312                  $PER22  ;CALL=PERR22    TRAP+46(104446) PROGRAM DETECTED ERROR
 8957  060452  047330                  $PER23  ;CALL=PERR23    TRAP+47(104447) PROGRAM DETECTED ERROR
 8958  060454  047350                  $PER24  ;CALL=PERR24    TRAP+50(104450) PROGRAM DETECTED ERROR
 8959  060456  047370                  $PER25  ;CALL=PERR25    TRAP+51(104451) PROGRAM DETECTED ERROR
 8960  060460  047600                  $PER26  ;CALL=PERR26    TRAP+52(104452) PROGRAM DETECTED ERROR
 8961  060462  047616                  $PER27  ;CALL=PERR27    TRAP+53(104453) PROGRAM DETECTED ERROR
 8962  060464  047754                  $PER30  ;CALL=PERR30    TRAP+54(104454) PROGRAM DETECTED ERROR
 8963  060466  050154                  $PER31  ;CALL=PERR31    TRAP+55(104455) PROGRAM DETECTED ERROR
 8964  060470  050240                  $PER32  ;CALL=PERR32    TRAP+56(104456) PROGRAM DETECTED ERROR
 8965  060472  050256                  $PER33  ;CALL=PERR33    TRAP+57(104457) PROGRAM DETECTED ERROR
 8966  060474  060600                  $NOTRAP ;CALL=PERR34    TRAP+60(104460) PROGRAM DETECTED ERROR
 8967  060476  060600                  $NOTRAP ;CALL=PERR35    TRAP+61(104461) PROGRAM DETECTED ERROR
 8968  060500  060600                  $NOTRAP ;CALL=PERR36    TRAP+62(104462) PROGRAM DETECTED ERROR
 8969  060502  060600                  $NOTRAP ;CALL=PERR37    TRAP+63(104463) PROGRAM DETECTED ERROR
 8970  060504  060600                  $NOTRAP ;CALL=PERR40    TRAP+64(104464) PROGRAM DETECTED ERROR
 8971  060506  060600                  $NOTRAP ;CALL=PERR41    TRAP+65(104465) PROGRAM DETECTED ERROR
 8972  060510  060600                  $NOTRAP ;CALL=PERR42    TRAP+66(104466) PROGRAM DETECTED ERROR
 8973  060512  060600                  $NOTRAP ;CALL=PERR43    TRAP+67(104467) PROGRAM DETECTED ERROR
 8974
 8975  060514  033314                  $ECCDIS ;CALL=ECCDIS    TRAP+70(104470) DISABLE ECC ON ALL CSR'S
 8976  060516  033350                  $ECC1DIS;CALL=ECC1DIS   TRAP+71(104471) DISABLE ECC ON 1 SELECTED CSR
 8977  060520  033402                  $ECCINIT;CALL=ECCINIT   TRAP+72(104472) INITIALIZE ALL MK11 CSR'S
 8978  060522  033422                  $ECC1INIT;CALL=ECC1INIT TRAP+73(104473) INITIALIZE 1 SELECTED MK11 CSR
 8979  060524  033526                  $CBCSR  ;CALL=CBCSR     TRAP+74(104474) WRITE GENERATED CHECKBITS IN ALL CSR'S
 8980  060526  033570                  $CB1CSR ;CALL=CB1CSR    TRAP+75(104475) WRITE GENERATED CHECKBITS IN 1 SELECTED
 8981  060530  033630                  $WASSBE ;CALL=WASSBE    TRAP+76(104476) WAS THERE A SBE ON ANY CSR?
 8982  060532  033744                  $WAS1SBE;CALL=WAS1SBE   TRAP+77(104477) WAS THERE A SBE ON 1 SELECTED CSR?
 8983  060534  033774                  $WASDBE ;CALL=WASDBE    TRAP+100(104500)WAS THERE A DBE ON ANY CSR?
 8984  060536  034110                  $WAS1DBE;CALL=WAS1DBE   TRAP+101(104501)WAS THERE A DBE ON 1 SELECTED CSR?
 8985  060540  034162                  $CLRCSR ;CALL=CLRCSR    TRAP+102(104502)CLEAR ALL CSR'S
 8986  060542  034200                  $CLR1CSR;CALL=CLR1CSR   TRAP+103(104503)CLEAR 1 SELECTED CSR
 8987  060544  034214                  $CHKDIS ;CALL=CHKDIS    TRAP+104(104504)DISABLE ECC & WRITE CKBITS FROM ALL CSR'
 8988  060546  034250                  $CHK1DIS;CALL=CHK1DIS   TRAP+105(104505)DISABLE ECC & WRITE CKBITS FROM 1 SELECT
 8989  060550  033440                  $ENASBE ;CALL=ENASBE    TRAP+106(104506)ENABLE TRAPS ON SBE'S FROM ALL CSR'S
 8990  060552  033474                  $ENA1SBE;CALL=ENA1SBE   TRAP+107(104507)ENABLE TRAPS ON SBE'S FROM 1 SELECTED CS
 8991  060554  033146                  $TSTRD  ;CALL=TSTREAD   TRAP+110(104510)TEAT LOC (R1) & READ CSR CAREFULLY
```

```
  8992  060556  060600                          $NOTRAP
  8993  060560  060600                          $NOTRAP
  8994  060562  060600                          $NOTRAP
  8995  060564  060600                          $NOTRAP
  8996  060566  060600                          $NOTRAP
  8997  060570  060600                          $NOTRAP
  8998  060572  060600                          $NOTRAP
  8999  060574  060600                          $NOTRAP
  9000  060576  060600                          $NOTRAP
  9001  060600                        $NOTRAP:  TYPE    MSG006          ;UNDEFINED TRAP INSTRUCTION
    (1) 060600  104401  066556                  TYPEIT  ,MSG006
  9002  060604  000000                $HALT2:   HALT
```

```
                                        .SBTTL  TABLE   ERROR POINTER

                                ;*THIS TABLE CONTAINS THE INFORMATION FOR EACH ERROR THAT CAN OCCUR.
                                ;*THE INFORMATION IS OBTAINED BY USING THE INDEX NUMBER FOUND IN
                                ;*LOCATION $ITEMB. THIS NUMBER INDICATES WHICH ITEM IN THE TABLE IS PERTINENT.
                                ;*NOTE1:        IF $ITEMB IS 0 THE ONLY PERTINENT DATA IS (ERRORPC).
                                ;*NOTE2:        EACH ITEM IN THE TABLE CONTAINS 4 POINTERS EXPLAINED AS FOLLOWS:

                                ;*      EM              ;;POINTS TO THE ERROR MESSAGE
                                ;*      DH              ;;POINTS TO THE DATA HEADER
                                ;*      DT              ;;POINTS TO THE DATA
                                ;*      DF              ;;POINTS TO THE DATA FORMAT


060606                          $ERRTB: ;ERROR  1               ;SPARE
060606  061326                          EM1                     ;SPARE
060610  064103                          DH2                     ;SPARE
060612  065400                          DT2                     ;SPARE
060614  065742                          DF2                     ;SPARE
                                        ;ERROR  2
060616  061326                          EM2
060620  064046                          DH1
060622  065366                          DT1
060624  065742                          DF2
                                        ;ERROR  3
060626  061364                          EM3
060630  064103                          DH3
060632  065400                          DT3
060634  066070                          DF9
                                        ;ERROR  4
060636  061417                          EM4
060640  064103                          DH3
060642  065410                          DT4
060644  066070                          DF9
                                        ;ERROR  5
060646  061465                          EM5
060650  064144                          DH5
060652  065420                          DT5
060654  065742                          DF2
                                        ;ERROR  6
060656  061542                          EM6
060660  064144                          DH5
060662  065420                          DT5
060664  065742                          DF2
                                        ;ERROR  7
060666  061567                          EM7
060670  064144                          DH5
060672  065420                          DT5
060674  065742                          DF2
                                        ;ERROR  10
060676  061741                          EM10
060700  064046                          DH1
060702  065366                          DT1
060704  065742                          DF2
```

```
 9060                                           ;ERROR  11
 9061  060706  062005                           EM11
 9062  060710  064321                           DH7
 9063  060712  065460                           DT7
 9064  060714  065766                           DF3
 9065                                           ;ERROR  12
 9066  060716  062005                           EM11
 9067  060720  064321                           DH7
 9068  060722  065460                           DT7
 9069  060724  066003                           DF4
 9070                                           ;ERROR  13
 9071  060726  062027                           EM12
 9072  060730  064602                           DH10
 9073  060732  065542                           DT10
 9074  060734  065742                           DF2
 9075                                           ;ERROR  14
 9076  060736  062005                           EM11
 9077  060740  064321                           DH7
 9078  060742  065460                           DT7
 9079  060744  066020                           DF5
 9080                                           ;ERROR  15
 9081  060746  062005                           EM11
 9082  060750  064321                           DH7
 9083  060752  065460                           DT7
 9084  060754  066035                           DF6
 9085                                           ;ERROR  16
 9086  060756  062053                           EM13
 9087  060760  064733                           DH13
 9088  060762  065600                           DT13
 9089  060764  066102                           DF11
 9090                                           ;ERROR  17
 9091  060766  062105                           EM14
 9092  060770  064733                           DH13
 9093  060772  065600                           DT13
 9094  060774  066102                           DF11
 9095                                           ;ERROR  20
 9096  060776  062151                           EM15
 9097  061000  064733                           DH13
 9098  061002  065600                           DT13
 9099  061004  065742                           DF2
 9100                                           ;ERROR  21
 9101  061006  062217                           EM16
 9102  061010  064667                           DH11
 9103  061012  065562                           DT11
 9104  061014  065742                           DF2
 9105                                           ;ERROR  22
 9106  061016  062245                           EM17
 9107  061020  064321                           DH7
 9108  061022  065460                           DT7
 9109  061024  066020                           DF5
 9110                                           ;ERROR  23
 9111  061026  062305                           EM18
 9112  061030  064443                           DH8
 9113  061032  065514                           DT8
 9114  061034  066075                           DF10
```

```
 9116
 9117  061036  062364                    ;ERROR  24
 9118  061040  064733                    EM19
 9119  061042  065600                    DH13
 9120  061044  065742                    DT13
 9121                                    DF2
 9122  061046  062441                    ;ERROR  25
 9123  061050  064733                    EM20
 9124  061052  065600                    DH13
 9125  061054  065742                    DT13
 9126                                    DF2
 9127  061056  000000                    ;ERROR  26
 9128  061060  064716                    0               ;NO MESSAGE
 9129  061062  065572                    DH12
 9130  061064  065742                    DT12
 9131                                    DF2
 9132  061066  062523                    ;ERROR  27
 9133  061070  064667                    EM21
 9134  061072  065562                    DH11
 9135  061074  065742                    DT11
 9136                                    DF2
 9137  061076  062562                    ;ERROR  30
 9138  061100  064733                    EM22
 9139  061102  065600                    DH13
 9140  061104  066102                    DT13
 9141                                    DF11
 9142  061106  000000                    ;ERROR  31
 9143  061110  065041                    0               ;NO MESSAGE
 9144  061112  065624                    DH14
 9145  061114  065742                    DT14
 9146                                    DF2
 9147  061116  062607                    ;ERROR  32
 9148  061120  064144                    EM23
 9149  061122  065420                    DH5
 9150  061124  065742                    DT5
 9151                                    DF2
 9152  061126  062636                    ;ERROR  33      ;SPARE
 9153  061130  064667                    EM24            ;SPARE
 9154  061132  065562                    DH11            ;SPARE
 9155  061134  065742                    DT11            ;SPARE
 9156                                    DF2             ;SPARE
 9157  061136  062636                    ;ERROR  34
 9158  061140  065157                    EM25
 9159  061142  065652                    DH15
 9160  061144  066052                    DT16
 9161                                    DF7
 9162  061146  062663                    ;ERROR  35
 9163  061150  065301                    EM26
 9164  061152  065704                    DH16
 9165  061154  065766                    DT17
 9166                                    DF3
 9167  061156  061627                    ;ERROR  36
 9168  061160  065316                    EM8
 9169  061162  065712                    DH17
 9170  061164  065742                    DT18
                                         DF2
```

```
9172                                            ;ERROR  37
9173  061166  061664                            EM9
9174  061170  064543                            DH9
9175  061172  065530                            DT9
9176  061174  065742                            DF2
9177                                            ;ERROR  40
9178  061176  062734                            EM27
9179  061200  065301                            DH16
9180  061202  065704                            DT17
9181  061204  066066                            DF8
9182                                            ;ERROR  41
9183  061206  063023                            EM28
9184  061210  065316                            DH17
9185  061212  065726                            DT19
9186  061214  065742                            DF2
9187                                            ;ERROR  42
9188  061216  063645                            EM35
9189  061220  064321                            DH7
9190  061222  065460                            DT7
9191  061224  065766                            DF3
9192                                            ;ERROR  43
9193  061226  063056                            EM29
9194  061230  064321                            DH7
9195  061232  065460                            DT7
9196  061234  065766                            DF3
9197                                            ;ERROR  44
9198  061236  063140                            EM30
9199  061240  064321                            DH7
9200  061242  065460                            DT7
9201  061244  066020                            DF5
9202                                            ;ERROR  45
9203  061246  063257                            EM31
9204  061250  064321                            DH7
9205  061252  065460                            DT7
9206  061254  065766                            DF3
9207                                            ;ERROR  46
9208  061256  063357                            EM32
9209  061260  064321                            DH7
9210  061262  065460                            DT7
9211  061264  065766                            DF3
9212                                            ;ERROR  47
9213  061266  063464                            EM33
9214  061270  064321                            DH7
9215  061272  065460                            DT7
9216  061274  065766                            DF3
9217                                            ;ERROR  50
9218  061276  063572                            EM34
9219  061300  000000                            0
9220  061302  000000                            0
9221  061304  000000                            0
9222                                            ;ERROR  51
9223  061306  063732                            EM36
9224  061310  064252                            DH6
9225  061312  065444                            DT6
9226  061314  065742                            DF2
```

```
 9228                                          ;ERROR  52
 9229  061316  064001                          EM37
 9230  061320  065316                          DH17
 9231  061322  065712                          DT18
 9232  061324  065742                          DF2
```

```
                                                        .SBTTL  ERROR MESSAGES (EM)
                                                EM1:    ;SPARE
061326                                          EM2:    .ASCIZ  /CAN'T SET 22 BIT MODE IN MMR3/
061326  040503  023516  020124
061334  042523  020124  031062
061342  041040  052111  046440
061350  042117  020105  047111
061356  046440  051115  000063
061364  040520  044522  054524                  EM3:    .ASCIZ  /PARITY ERROR(S) IN BANK 0)/
061372  042440  051122  051117
061400  051450  020051  047111
061406  041040  047101  020113
061414  024460     000
061417     116  047117  042455                  EM4:    .ASCIZ  /NON-EXISTANT MEMORY (HOLES) IN BANK 0/
061424  044530  052123  047101
061432  020124  042515  047515
061440  054522  024040  047510
061446  042514  024523  044440
061454  020116  040502  045516
061462  030040     000
061465     111  046114  043505                  EM5:    .ASCIZ  /ILLEGAL OR RESERVED INSTRUCTION (TRAP TO 10)/
061472  046101  047440  020122
061500  042522  042523  053122
061506  042105  044440  051516
061514  051124  041525  044524
061522  047117  024040  051124
061530  050101  052040  020117
061536  030061  000051
061542  047125  054105  042520                  EM6:    .ASCIZ  /UNEXPECTED TRAP TO 4/
061550  052103  042105  052040
061556  040522  020120  047524
061564  032040     000
061567     115  046505  051117                  EM7:    .ASCIZ  /MEMORY MANAGEMENT (TRAP TO 250)/
061574  020131  040515  040516
061602  042507  042515  052116
061610  024040  051124  050101
061616  052040  020117  032462
061624  024460     000
061627     120  051117  020124                  EM8:    .ASCIZ  /PORT CONTENTION NOT BALANCED/
061634  047503  052116  047105
061642  044524  047117  047040
061650  052117  041040  046101
061656  047101  042503  000104

061664  047516  026516  054105                  EM9:    .ASCIZ  /NON-EXISTANT CSR CAUSED INCORRECT ERROR TRAP/
061672  051511  040524  052116
061700  041440  051123  041440
061706  052501  042523  020104
061714  047111  047503  051122
061722  041505  020124  051105
061730  047522  020122  051124
061736  050101     000
061741     111  041516  051117                  EM10:   .ASCIZ  /INCORRECT DATA FROM DEVICE REGISTER/
061746  042522  052103  042040
061754  052101  020101  051106
061762  046517  042040  053105
061770  041511  020105  042522
```

```
                061776  044507  052123  051105
                062004     000
 9248           062005     115  046505  051117  EM11:   .ASCIZ  /MEMORY DATA ERROR/
                062012  020131  040504  040524
                062020  042440  051122  051117
                062026     000
 9249           062027     104  052105  044501  EM12:   .ASCIZ  /DETAILED ERROR DUMP/
                062034  042514  020104  051105
                062042  047522  020122  052504
                062050  050115     000
 9250           062053     115  051511  044523  EM13:   .ASCIZ  /MISSING EXPECTED SBE FLAG/
                062060  043516  042440  050130
                062066  041505  042524  020104
                062074  041123  020105  046106
                062102  043501     000
 9251           062105     127  044522  042524  EM14:   .ASCIZ  /WRITE BYTE FAILED TO CLEAR SBE FLAG/
                062112  041040  052131  020105
                062120  040506  046111  042105
                062126  052040  020117  046103
                062134  040505  020122  041123
                062142  020105  046106  043501
                062150     000
 9252           062151     106  044501  042514  EM15:   .ASCIZ  /FAILED TO GET INTERRUPT WITH DBE FLAG/
                062156  020104  047524  043440
                062164  052105  044440  052116
                062172  051105  052522  052120
                062200  053440  052111  020110
                062206  041104  020105  046106
                062214  043501     000
 9253           062217     117  046116  020131  EM16:   .ASCIZ  /ONLY ONE DBE FLAG SET/
                062224  047117  020105  041104
                062232  020105  046106  043501
                062240  051440  052105     000
 9254           062245     115  046505  051117  EM17:   .ASCIZ  /MEMORY DATA ERROR IN CHECK BITS/
                062252  020131  040504  040524
                062260  042440  051122  051117
                062266  044440  020116  044103
                062274  041505  020113  044502
                062302  051524     000
 9255           062305     123  051531  042524  EM18:   .ASCIZ  /SYSTEM SIZE REGISTER LOW DOES NOT MATCH MEMORY/
                062312  020115  044523  042532
                062320  051040  043505  051511
                062326  042524  020122  047514
                062334  020127  047504  051505
                062342  047040  052117  046440
                062350  052101  044103  046440
                062356  046505  051117  000131
 9256           062364  041123  020105  051117  EM19:   .ASCIZ  /SBE OR DBE CAUSED PARITY TRAP WHEN INHIBITED/
                062372  042040  042502  041440
                062400  052501  042523  020104
                062406  040520  044522  054524
                062414  052040  040522  020120
                062422  044127  047105  044440
                062430  044116  041111  052111
                062436  042105     000
 9257           062441     123  042502  047440  EM20:   .ASCIZ  /SBE OR DBE DID NOT CAUSE PARITY TRAP WHEN ENABLED/
```

```
                062446  020122  041104  020105
                062454  044504  020104  047516
                062462  020124  040503  051525
                062470  020105  040520  044522
                062476  054524  052040  040522
                062504  020120  044127  047105
                062512  042440  040516  046102
                062520  042105     000
 9258           062523     123  042502  047440  EM21:   .ASCIZ  /SBE OR DBE ON MASTER TEST WORD/
                062530  020122  041104  020105
                062536  047117  046440  051501
                062544  042524  020122  042524
                062552  052123  053440  051117
                062560  000104
 9259           062562  044515  051523  047111  EM22:   .ASCIZ  /MISSING EXPECTED DBE/
                062570  020107  054105  042520
                062576  052103  042105  042040
                062604  042502     000
 9260           062607     125  042516  050130  EM23:   .ASCIZ  /UNEXPECTED PARITY TRAP/
                062614  041505  042524  020104
                062622  040520  044522  054524
                062630  052040  040522  000120
 9261           062636                          EM24:   ;SPARE
 9262           062636  044103  041505  020113  EM25:   .ASCIZ  /CHECK BIT DATA ERROR/
                062644  044502  020124  040504
                062652  040524  042440  051122
                062660  051117     000
 9263           062663     101  042104  042522  EM26:   .ASCIZ  /ADDRESS PARITY ERROR DID NOT CAUSE ABORT/
                062670  051523  050040  051101
                062676  052111  020131  051105
                062704  047522  020122  044504
                062712  020104  047516  020124
                062720  040503  051525  020105
                062726  041101  051117  000124
 9264           062734  045515  030461  050040  EM27:   .ASCIZ  /MK11 PROTECTION POINTER FAILURE - DID NOT PROTECT BANK/
                062742  047522  042524  052103
                062750  047511  020116  047520
                062756  047111  042524  020122
                062764  040506  046111  051125
                062772  020105  020055  044504
                063000  020104  047516  020124
                063006  051120  052117  041505
                063014  020124  040502  045516
                063022     000
 9268           063056  047503  051122  041505  EM29:   .ASCIZ  /CORRECTION FAILURE WITH ECC ENABLED ON FORCED SBE/
                063064  044524  047117  043040
                063072  044501  052514  042522
                063100  053440  052111  020110
                063106  041505  020103  047105
                063114  041101  042514  020104
                063122  047117  043040  051117
                063130  042503  020104  041123
                063136  000105
 9269           063140  051127  052111  020105  EM30:   .ASCII  /WRITE BYTE (MOVB) WITH ECC ENABLED FAILED TO CLEAR DATA AT/<CRLF>
                063146  054502  042524  024040
                063154  047515  041126  020051
```

```
        063162  044527  044124  042440
        063170  041503  042440  040516
        063176  046102  042105  043040
        063204  044501  042514  020104
        063212  047524  041440  042514
        063220  051101  042040  052101
        063226  020101  052101     200
 9270   063233     106  051117  042503          .ASCIZ  /FORCED SBE LOCATION/
        063240  020104  041123  020105
        063246  047514  040503  044524
        063254  047117     000
 9271   063257     101  051123  020102  EM31:   .ASCIZ  /ASRB (R3)+ WITH ECC ENABLED CHANGED DATA AT FORCED DBE LOCATION/
        063264  051050  024463  020053
        063272  044527  044124  042440
        063300  041503  042440  040516
        063306  046102  042105  041440
        063314  040510  043516  042105
        063322  042040  052101  020101
        063330  052101  043040  051117
        063336  042503  020104  041104
        063344  020105  047514  040503
        063352  044524  047117     000
 9272   063357     115  053117  020102  EM32:   .ASCIZ  /MOVB #360,(R2)+ WITH ECC ENABLED CHANGED DATA AT FORCED DBE LOCATION/
        063364  031443  030066  024054
        063372  031122  025451  053440
        063400  052111  020110  041505
        063406  020103  047105  041101
        063414  042514  020104  044103
        063422  047101  042507  020104
        063430  040504  040524  040440
        063436  020124  047506  041522
        063444  042105  042040  042502
        063452  046040  041517  052101
        063460  047511  000116
 9273   063464  047515  020126  030443  EM33:   .ASCIZ  /MOV #177400,(R1) WITH ECC ENABLED CHANGED DATA AT FORCED DBE LOCATION/
        063472  033467  030064  026060
        063500  051050  024461  053440
        063506  052111  020110  041505
        063514  020103  047105  041101
        063522  042514  020104  044103
        063530  047101  042507  020104
        063536  040504  040524  040440
        063544  020124  047506  041522
        063552  042105  042040  042502
        063560  046040  041517  052101
        063566  047511  000116
 9274   063572  041123  020105  042504  EM34:   .ASCIZ  /SBE DETECTED BUT NOT ISOLATED TO ANY ARRAY/
        063600  042524  052103  042105
        063606  041040  052125  047040
        063614  052117  044440  047523
        063622  040514  042524  020104
        063630  047524  040440  054516
        063636  040440  051122  054501
        063644     000
 9275   063645     125  042516  050130  EM35:   .ASCIZ  /UNEXPECTED CORRECTION WITH ECC DISABLE ON FORCED SBE/
        063652  041505  042524  020104
```

```
        063660  047503  051122  041505
        063666  044524  047117  053440
        063674  052111  020110  041505
        063702  020103  044504  040523
        063710  046102  020105  047117
        063716  043040  051117  042503
        063724  020104  041123  000105
 9276   063732  050101  020124  044523  EM36:    .ASCIZ  /APT SIZE DISAGREES WITH PROGRAM SIZING/
        063740  042532  042040  051511
        063746  043501  042522  051505
        063754  053440  052111  020110
        063762  051120  043517  040522
        063770  020115  044523  044532
        063776  043516     000
 9280                                           .EVEN
```

```
                                                        .SBTTL  ERROR DATA HFADERS (DH)
9284  064046  020040  041520  020040  DH1:    .ASCIZ  /  PC   DEV ADD   GOOD    BAD/
      064054  042040  053105  040440
      064062  042104  020040  043440
      064070  047517  020104  020040
      064076  041040  042101     000
9285  064103                          DH2:    ;SPARE
9286  064103     040  050040  020103  DH3:    .ASCIZ  /  PC   1ST ADD  # OF ERRORS/
      064110  020040  051461  020124
      064116  042101  020104  021440
      064124  047440  020106  051105
      064132  047522  051522     000
9287  064137     040  050040  000103  DH4:    .ASCIZ  /  PC/
9288  064144  020040  041520  020040  DH5:    .ASCII  /  PC    CPUERR  MEMERR  HIADRS  LOADRS/
      064152  020040  050103  042525
      064160  051122  020040  042515
      064166  042515  051122  020040
      064174  044510  042101  051522
      064202  020040  047514  042101
      064210  051522
9289  064212  020040  046440  051115          .ASCIZ  /   MMR0    MMR1    MMR2    MMR3/
      064220  020060  020040  046440
      064226  051115  020061  020040
      064234  046440  051115  020062
      064242  020040  046440  051115
      064250  000063
9290  064252  020040  041520  020040  DH6:    .ASCIZ  /  PC   APTCORE  MJSIZE  APTMOS  MKSIZE/
      064260  040440  052120  047503
      064266  042522  020040  045115
      064274  044523  042532  020040
      064302  050101  046524  051517
      064310  020040  045515  044523
      064316  042532     000
9291  064321     040  050040  020103  DH7:    .ASCII  /  PC    BANK  VADD     PADD     GOOD/
      064326  020040  041040  047101
      064334  020113  053040  042101
      064342  020104  020040  020040
      064350  040520  042104  020040
      064356  020040  043440  047517
      064364     104
9292  064365     040  020040  020040          .ASCIZ  /     BAD     XOR  MAR BOX MTYPE INT PAT ARRAY/
      064372  040502  020104  020040
      064400  020040  047530  020122
      064406  046440  051101  041040
      064414  054117  046440  054524
      064422  042520  044440  052116
      064430  050040  052101  040440
      064436  051122  054501     000
9293  064443     104  053105  040440  DH8:    .ASCIZ  /DEV ADD   GOOD    BAD     GDSWITCH    BDSWITCH  NOTE: (OFF = 1)/
      064450  042104  020040  043440
      064456  047517  020104  020040
      064464  041040  042101  020040
      064472  020040  043440  051504
      064500  044527  041524  020110
      064506  020040  041040  051504
      064514  044527  041524  020110
```

```
                064522  047040  052117  035105
                064530  024040  043117  020106
                064536  020075  024461     000
  9294          064543     040  050040  020103  DH9:    .ASCIZ  /  PC    CPUERR  MEMERR  CSRNUM/
                064550  020040  041440  052520
                064556  051105  020122  046440
                064564  046505  051105  020122
                064572  041440  051123  052516
                064600  000115
  9295          064602  020040  030122  020040  DH10:   .ASCIZ  /  R0      R1      R2      R3      R4      R5      SP/
                064610  020040  020040  030522
                064616  020040  020040  020040
                064624  031122  020040  020040
                064632  020040  031522  020040
                064640  020040  020040  032122
                064646  020040  020040  020040
                064654  032522  020040  020040
                064662  020040  050123     000
  9296          064667     040  050040  020103  DH11:   .ASCIZ  /  PC      CSR     CSR+2/
                064674  020040  020040  041440
                064702  051123  020040  020040
                064710  051503  025522  000062
  9297          064716  041440  051123  020040  DH12:   .ASCIZ  / CSR   CSR+2/
                064724  041440  051123  031053
                064732     000
  9298          064733     040  050040  020103  DH13:   .ASCII  /  PC    BANK  VADD     PADD    WROTE1  WROTE2/
                064740  020040  041040  047101
                064746  020113  053040  042101
                064754  020104  020040  020040
                064762  040520  042104  020040
                064770  020040  051127  052117
                064776  030505  020040  051127
                065004  052117  031105
  9299          065010  041440  045510  044502          .ASCIZ  / CHKBITS    CSR    CSR+2/
                065016  051524  020040  020040
                065024  051503  020122  020040
                065032  041440  051123  031053
                065040     000
  9300          065041     115  046505  051105  DH14:   .ASCII  /MEMERR   MAINT  CONTRL  HIADRS  LOADRS  CPUERR/
                065046  020122  020040  040515
                065054  047111  020124  041440
                065062  047117  051124  020114
                065070  044040  040511  051104
                065076  020123  046040  040517
                065104  051104  020123  041440
                065112  052520  051105     122
  9301          065117     040  020040  046515          .ASCIZ  /   MMR0    MMR1    MMR2    MMR3/
                065124  030122  020040  020040
                065132  046515  030522  020040
                065140  020040  046515  031122
                065146  020040  020040  046515
                065154  031522     000
  9302          065157     040  050040  020103  DH15:   .ASCII  /  PC    BANK    PADD    GD-WD1  GD-WD2  GD-CHK/
                065164  020040  041040  047101
                065172  020113  020040  050040
                065200  042101  020104  020040
```

```
        065206  042107  053455  030504
        065214  020040  042107  053455
        065222  031104  020040  042107
        065230  041455  045510
 9303   065234  041040  042101  053455          .ASCIZ  / BAD-WD1 BAD-WD2 BAD-CHK MAR INT PAT/
        065242  030504  041040  042101
        065250  053455  031104  041040
        065256  042101  041455  045510
        065264  046440  051101  044440
        065272  052116  050040  052101
        065300     000
 9304   065301     040  050040  020103  DH16:   .ASCIZ  /  PC    BANK/
        065306  020040  041040  047101
        065314  000113
 9308           065366                          .EVEN
```

```
                                                        .SBTTL  ERROR DATA TAGS (DT)
9312    065366  001416  001422  001432  DT1:    .WORD   ERRORPC,ADDRESS,GOOD,BAD,0
        065374  001440  000000
9313    065400                          DT2:    ;SPARE
9314    065400  001416  001424  001460  DT3:    .WORD   ERRORPC,PADDRESS,PARCNT,0
        065406  000000
9315    065410  001416  001422  001456  DT4:    .WORD   ERRORPC,ADDRESS,NEMCNT,0
        065416  000000
9316    065420  001416  177766  177744  DT5:    .WORD   ERRORPC,CPUERR,MEMERR,HIADRS,LOADRS,MMR0,MMR1,MMR2,MMR3,0
        065426  177742  177740  177572
        065434  177574  177576  172516
        065442  000000
9317    065444  001416  002104  002046  DT6:    .WORD   ERRORPC,APTCORE,MJSIZE,APTMOS,MKSIZE,0
        065452  002106  002050  000000
9318    065460  001416  001662  001422  DT7:    .WORD   ERRORPC,DUMMY,ADDRESS,DUMMY,GOOD,BAD,BADXOR
        065466  001662  001432  001440
        065474  001446
9319    065476  001662  001662  001662          .WORD   DUMMY,DUMMY,DUMMY,DUMMY,DUMMY,DUMMY,0
        065504  001662  001662  001662
        065512  000000
9320    065514  001422  001432  001440  DT8:    .WORD   ADDRESS,GOOD,BAD,GDSWITCH,BDSWITCH,0
        065522  002060  002062  000000
9321    065530  001416  177766  177744  DT9:    .WORD   ERRORPC,CPUERR,MEMERR,CSRNO,0
        065536  001634  000000
9322    065542  001664  001666  001670  DT10:   .WORD   DETR0,DETR1,DETR2,DETR3,DETR4,DETR5,DETR6,0
        065550  001672  001674  001676
        065556  001700  000000
9323    065562  001416  001526  001530  DT11:   .WORD   ERRORPC,CSR,CSR+2,0
        065570  000000
9324    065572  001526  001530  000000  DT12:   .WORD   CSR,CSR+2,0
9325    065600  001416  001662  001422  DT13:   .WORD   ERRORPC,DUMMY,ADDRESS,DUMMY,TSTDAT,TSTDAT+2,CHECK,CSR,CSR+2,0
        065606  001662  001716  001720
        065614  001762  001526  001530
        065622  000000
9326    065624  177744  177750  177746  DT14:   .WORD   MEMERR,MAINT,CONTRL,HIADRS,LOADRS,CPUERR,MMR0,MMR1,MMR2,MMR3,0
        065632  177742  177740  177766
        065640  177572  177574  177576
        065646  172516  000000
9327    065652  001416  001662  001662  DT16:   .WORD   ERRORPC,DUMMY,DUMMY,GOOD,GOOD2,GOOD3
        065660  001432  001434  001436
9328    065666  001440  001442  001444          .WORD   BAD,BAD2,BAD3,DUMMY,DUMMY,DUMMY,0
        065674  001662  001662  001662
        065702  000000
9329    065704  001416  001662  000000  DT17:   .WORD   ERRORPC,DUMMY,0
```

```
 9336                                           .SBTTL  ERROR DATA FORMATS (DF)
 9337  065742  000  000  000  DF2:    .BYTE   0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
       065745  000  000  000
       065750  000  000  000
       065753  000  000  000
       065756  000  000  000
       065761  000  000  000
       065764  000  000
 9338  065766  000  005  000  DF3:    .BYTE   0,5.,0,9.,,0,0,0,10.,,3,6,2,4,12.
       065771  011  000  000
       065774  000  012  003
       065777  006  002  004
       066002  014
 9339  066003  000  005  000  DF4:    .BYTE   0,5,0,9.,,0,8.,,8.,,10.,,3,6,2,4,12.
       066006  011  000  010
       066011  010  012  003
       066014  006  002  004
       066017  014
 9340  066020  000  005  000  DF5:    .BYTE   0,5,0,9.,,11.,,11.,,11.,,10.,,3,6,2,4,12.
       066023  011  013  013
       066026  013  012  003
       066031  006  002  004
       066034  014
 9341  066035  000  005  000  DF6:    .BYTE   0,5,0,9.,,11.,,8.,,8.,,10.,,3,6,2,4,12.
       066040  011  013  010
       066043  010  012  003
       066046  006  002  004
       066051  014
 9342  066052  000  005  011  DF7:    .BYTE   0,5,9.,,0,0,11.,,0,0,11.,,10.,,2,4
       066055  000  000  013
       066060  000  000  013
       066063  012  002  004
 9343  066066  000  005       DF8:    .BYTE   0,5
 9344  066070  000  001  001  DF9:    .BYTE   0,1,1,1,1
       066073  001  001
 9345  066075  000  000  000  DF10:   .BYTE   0,0,0,13.,,13.
       066100  015  015
 9346  066102  000  005  000  DF11:   .BYTE   0,5,0,9.,,0,0,0,0,0
       066105  011  000  000
       066110  000  000  000
 9347          066114         .EVEN
```

```
 9350                                                .SBTTL  MESSAGES
 9351  066114  041600  046505  040513  MSG000: .ASCIZ  <CRLF></CEMKAA0 1170 MAIN MEMORY DIAGNOSTIC <V03-00>/
       066122  030101  030440  033461
       066130  020060  040515  047111
       066136  046440  046505  051117
       066144  020131  044504  043501
       066152  047516  052123  041511
       066160  036040  030126  026463
       066166  030060  000076
 9352  066172  020200  020040  020040  MSG001: .ASCIZ  <CRLF>/                          MEMORY CONFIGURATION MAP/
       066200  020040  020040  020040
       066206  020040  020040  020040
       066214  020040  020040  020040
       066222  046440  046505  051117
       066230  020131  047503  043116
       066236  043511  051125  052101
       066244  047511  020116  040515
       066252  000120
 9353  066254  020200  020040  020040  MSG002: .ASCIZ  <CRLF>/                                  16K WORD BANKS/
       066262  020040  020040  020040
       066270  020040  020040  020040
       066276  020040  020040  020040
       066304  020040  020040  020040
       066312  033061  020113  047527
       066320  042122  041040  047101
       066326  051513     000
 9354  066331     200  020040  020040  MSG003: .ASCII <CRLF>/                  1       2       3/
       066336  020040  020040  020040
       066344  020040  020040  020040
       066352  020061  020040  020040
       066360  020040  020062  020040
       066366  020040  020040     063
 9355  066373     040  020040  020040          .ASCIZ  /       4       5       6       7  /
       066400  020040  020064  020040
       066406  020040  020040  020065
       066414  020040  020040  020040
       066422  020066  020040  020040
       066430  020040  020067  000040
 9356  066436  020200  020040  020040  MSG004: .ASCII <CRLF>/        01234567012345670123456 7/
       066444  020040  030040  031061
       066452  032063  033065  030067
       066460  031061  032063  033065
       066466  030067  031061  032063
       066474  033065     067
 9357  066477     060  031061  032063          .ASCIZ  /012345670123456701234567012345670123/
       066504  033065  030067  031061
       066512  032063  033065  030067
       066520  031061  032063  033065
       066526  030067  031061  032063
       066534  033065  030067  031061
       066542  000063
 9358  066544  042600  051122  051117  MSG005: .ASCIZ  <CRLF>/ERRORS  /
       066552  020123  000040
 9359  066556  052600  042116  043105  MSG006: .ASCIZ  <CRLF>/UNDEFINED TRAP INSTRUCTION/
       066564  047111  042105  052040
       066572  040522  020120  047111
```

```
                066600  052123  052522  052103
                066606  047511  000116
  9360          066612  044600  052116  046122  MSG007: .ASCIZ  <CRLF>/INTRLV  /
                066620  020126  000040
  9364          066624  040600  041503  051505  MSG008: .ASCIZ  <CRLF>/ACCESSED/
                066632  042523  000104
  9366          066636  046600  046505  054524  MSG009: .ASCIZ  <CRLF>/MEMTYPE /                ;MEMORY TYPE (0=MJ11;1=MK11)
                066644  042520  000040
  9367          066650  050200  047522  042524  MSG010: .ASCIZ  <CRLF>/PROTECT /                ;MEMORY PROTECTED
                066656  052103  000040
  9368          066662  020040  020040  020060  MSG011: .ASCIZ  /    0       1       2       3       4       5       6/
                066670  020040  020040  020040
                066676  020061  020040  020040
                066704  020040  020062  020040
                066712  020040  020040  020063
                066720  020040  020040  020040
                066726  020064  020040  020040
                066734  020040  020065  020040
                066742  020040  020040  000066
  9369          066750  032464  033466  030460  MSG012: .ASCIZ  /4567012345670123456701234567012345670123456701234567012345670123456/
                066756  031462  032464  033466
                066764  030460  031462  032464
                066772  033466  030460  031462
                067000  032464  033466  030460
                067006  031462  032464  033466
                067014  030460  031462  032464
                067022  033466  030460  031462
                067030  032464  033466  030460
                067036  031462  032464  033466
                067044     000
  9370          067045     130     000          MSG013: .ASCIZ  /X/
  9371          067047     040     000          MSG014: .ASCIZ  / /                ;SPACE
  9372          067051     000     000          MSG015: .BYTE   0,0                ;FOR SINGLE ASCII CHARACTERS & TERMINATOR
  9373          067053     200  047502  020130  MSG016: .ASCIZ  <CRLF>/BOX     /
                067060  020040  020040     000
  9374          067065     040  020040  020040  MSG017: .ASCIZ  /        /                 ;8 SPACES
                067072  020040  000040
  9375          067076  020040     000          MSG018: .ASCIZ  /  /                       ;2 SPACES
  9376          067101     040  020040     000  MSG019: .ASCIZ  /   /                      ;3 SPACES
  9377          067105     200  044506  046105  MSG020: .ASCIZ  <CRLF>/FIELD SERVICE COMMAND MODE/
                067112  020104  042523  053122
                067120  041511  020105  047503
                067126  046515  047101  020104
                067134  047515  042504     000
  9378          067141     200  047503  046515  MSG021: .ASCII  <CRLF>/COMMANDS AVAILABLE ARE:/
                067146  047101  051504  040440
                067154  040526  046111  041101
                067162  042514  040440  042522
                067170     072
  9379          067171     200  020060  020075          .ASCII  <CRLF>/0 = EXIT FIELD SERVICE COMMANDS/
                067176  054105  052111  043040
                067204  042511  042114  051440
                067212  051105  044526  042503
                067220  041440  046517  040515
                067226  042116     123
  9380          067231     200  020061  020075          .ASCII  <CRLF>/1 = READ CSR/
```

```
                067236  042522  042101  041440
                067244  051123
 9381           067246  031200  036440  046040          .ASCII  <CRLF>/2 = LOAD CSR/
                067254  040517  020104  051503
                067262     122
 9382           067263     200  020063  020075          .ASCII  <CRLF>/3 = EXAMINE MEMORY/
                067270  054105  046501  047111
                067276  020105  042515  047515
                067304  054522
 9383           067306  032200  036440  046440          .ASCII  <CRLF>/4 = MODIFY MEMORY/
                067314  042117  043111  020131
                067322  042515  047515  054522
 9384           067330  032600  036440  051440          .ASCII  <CRLF>/5 = SELECT BANK,MARGIN, & PATTERN/
                067336  046105  041505  020124
                067344  040502  045516  046454
                067352  051101  044507  026116
                067360  023040  050040  052101
                067366  042524  047122
 9385           067372  033200  036440  052040          .ASCII  <CRLF>/6 = TYPE CONFIGURATION MAP/
                067400  050131  020105  047503
                067406  043116  043511  051125
                067414  052101  047511  020116
                067422  040515     120
 9386           067425     200  020067  020075          .ASCII  <CRLF>"7 = BATTERY BACKUP - P/F TEST"
                067432  040502  052124  051105
                067440  020131  040502  045503
                067446  050125  026440  050040
                067454  043057  052040  051505
                067462     124
 9387           067463     200  020070  020075          .ASCII  <CRLF>/8 = SOB-A-LONG TEST/
                067470  047523  026502  026501
                067476  047514  043516  052040
                067504  051505     124
 9388           067507     200  020071  020075          .ASCII  <CRLF>/9 = SUPER TIGHT SCOPE LOOP/
                067514  052523  042520  020122
                067522  044524  044107  020124
                067530  041523  050117  020105
                067536  047514  050117
 9389           067542  030600  036460  042440          .ASCII  <CRLF>/10= ERROR SUMMARY/
                067550  051122  051117  051440
                067556  046525  040515  054522
 9390           067564  030600  036461  051040          .ASCII  <CRLF>/11= REFRESH TEST/
                067572  043105  042522  044123
                067600  052040  051505     124
 9391           067605     200  031061  020075          .ASCII  <CRLF>/12= SET FILL COUNT/
                067612  042523  020124  044506
                067620  046114  041440  052517
                067626  052116
 9392           067630     015     012     000          .BYTE   15,12,0
 9393           067633     200  044127  041511  MSG022: .ASCIZ  <CRLF>/WHICH CSR(0-7)? /
                067640  020110  051503  024122
                067646  026460  024467  020077
                067654     000
 9394           067655     200  051503  020122  MSG023: .ASCIZ  <CRLF>/CSR - 1ST WORD? /
                067662  020055  051461  020124
                067670  047527  042122  020077
```

```
                067676      000
 9395           067677      200     051503  020122  MSG024: .ASCIZ  <CRLF>/CSR - 2ND WORD? /
                067704  020055      047062  020104
                067712  047527      042122  020077
                067720      000
 9396           067721      200     044124  051511  MSG025: .ASCIZ  <CRLF>/THIS CSR DOES NOT EXIST/
                067726  041440      051123  042040
                067734  042517      020123  047516
                067742  020124      054105  051511
                067750  000124
 9397           067752  041600      046517  040515  MSG026: .ASCIZ  <CRLF>/COMMAND:/
                067760  042116      000072
 9398           067764  047600      042114  041440  MSG027: .ASCIZ  <CRLF>/OLD CSR WAS/
                067772  051123      053440  051501
                070000      000
 9399           070001      200     051503  020122  MSG028: .ASCIZ  <CRLF>/CSR IS NOW/
                070006  051511      047040  053517
                070014      000
 9400           070015      200     054105  046501  MSG029: .ASCIZ  <CRLF>/EXAMINE MEMORY/
                070022  047111      020105  042515
                070030  047515      054522      000
 9401           070035      200     040502  045516  MSG030: .ASCIZ  <CRLF>/BANK(0-167)? /
                070042  030050      030455  033466
                070050  037451      000040
 9402           070054  050200      054510  044523  MSG031: .ASCIZ  <CRLF>/PHYSICAL ADDRESS(0-16777776)? /
                070062  040503      020114  042101
                070070  051104      051505  024123
                070076  026460      033061  033467
                070104  033467      033067  037451
                070112  000040
 9403           070114  050200      051101  052111  MSG032: .ASCIZ  <CRLF>/PARITY ABORT/
                070122  020131      041101  051117
                070130  000124
 9404           070132  052200      046511  047505  MSG033: .ASCIZ  <CRLF>/TIMEOUT TRAP/
                070140  052125      052040  040522
                070146  000120
 9405           070150  041200      050131  051501  MSG034: .ASCII  <CRLF>/BYPASSING CSR TESTS ON BOX /
                070156  044523      043516  041440
                070164  051123      052040  051505
                070172  051524      047440  020116
                070200  047502      020130
 9406           070204      000                     MSGA34: .BYTE   0
 9407           070205      040     047503  052116          .ASCII  / CONTROLLER /
                070212  047522      046114  051105
                070220      040
 9408           070221      000                     MSGB34: .BYTE   0
 9409           070222  042200      042525  052040          .ASCIZ  <CRLF>/DUE TO LACK OF SBE FREE LOCATIONS/
                070230  020117      040514  045503
                070236  047440      020106  041123
                070244  020105      051106  042505
                070252  046040      041517  052101
                070260  047511      051516      000
 9410           070265      121     000126          MSG035: .ASCIZ  /QV/
 9411           070270  046600      042117  043111  MSG036: .ASCIZ  <CRLF>/MODIFY MEMORY/
                070276  020131      042515  047515
                070304  054522          000
```

```
 9412  070307    200   046117  020104  MSG037: .ASCIZ  <CRLF>/OLD DATA WAS /
       070314 040504   040524  053440
       070322 051501   000040
 9413  070326 042200   052101  020101  MSG038: .ASCIZ  <CRLF>/DATA IS NOW /
       070334 051511   047040  053517
       070342 000040
 9414  070344 044600   050116  052125  MSG039: .ASCIZ  <CRLF>/INPUT NEW DATA? /
       070352 047040   053505  042040
       070360 052101   037501  000040
 9415  070366 051600   046105  041505  MSG040: .ASCIZ  <CRLF>/SELECT BANK, MARGIN, & PATTERN TEST/
       070374 020124   040502  045516
       070402 020054   040515  043522
       070410 047111   020054  020046
       070416 040520   052124  051105
       070424 020116   042524  052123
       070432    000
 9416  070433    200   040502  045516  MSG041: .ASCIZ  <CRLF>/BANK NOT ACCESSABLE/
       070440 047040   052117  040440
       070446 041503   051505  040523
       070454 046102   000105
 9417  070460 050200   052101  042524  MSG042: .ASCIZ  <CRLF>/PATTERN(0-37)? /
       070466 047122   030050  031455
       070474 024467   020077     000
 9418  070501    200   040520  052124  MSG043: .ASCIZ  <CRLF>/PATTERN 0 DATA IS? /
       070506 051105   020116  020060
       070514 040504   040524  044440
       070522 037523   000040
 9419  070526 046600   051101  044507  MSG044: .ASCIZ  <CRLF>/MARGIN(0,2,3,4,5)? /
       070534 024116   026060  026062
       070542 026063   026064  024465
       070550 020077      000
 9420  070553    200   040515  043522  MSG045: .ASCIZ  <CRLF>/MARGINS INHIBITED BY SW5!/
       070560 047111   020123  047111
       070566 044510   044502  042524
       070574 020104   054502  051440
       070602 032527   000041
 9421  070606 052200   020117  051505  MSG046: .ASCIZ  <CRLF>/TO ESCAPE TYPE ANY KEY/<CRLF><12><12>
       070614 040503   042520  052040
       070622 050131   020105  047101
       070630 020131   042513  100131
       070636 005012      000
 9422  070641    200   042524  052123  MSG047: .ASCIZ  <CRLF>/TEST COMPLETE/
       070646 041440   046517  046120
       070654 052105   000105
 9423  070660 047040   052117  040440  MSG048: .ASCIZ  / NOT AVAILABLE NOW - TRY LATER!/
       070666 040526   046111  041101
       070674 042514   047040  053517
       070702 026440   052040  054522
       070710 046040   052101  051105
       070716 000041
 9424  070720 041200   047101  020113  MSG049: .ASCIZ  <CRLF>/BANK REQUIRES RELOCATION/
       070726 042522   052521  051111
       070734 051505   051040  046105
       070742 041517   052101  047511
       070750 000116
 9425  070752 041200   052101  042524  MSG050: .ASCII  <CRLF>/BATTERY BACKUP TEST/
```

```
        070760  054522  041040  041501
        070766  052513  020120  042524
        070774  052123
 9426   070776  053600  044522  044524          .ASCIZ  <CRLF>/WRITING & CHECKING ADDRESS PATTERN AS BACKGROUND/
        071004  043516  023040  041440
        071012  042510  045503  047111
        071020  020107  042101  051104
        071026  051505  020123  040520
        071034  052124  051105  020116
        071042  051501  041040  041501
        071050  043513  047522  047125
        071056  000104
 9427   071060  050200  053517  051105  MSG051: .ASCIZ  <CRLF>/POWER RECOVERY/
        071066  051040  041505  053117
        071074  051105  000131
 9428   071100  051200  046505  053117  MSG052: .ASCIZ  <CRLF>/REMOVE SYSTEM POWER FOR/
        071106  020105  054523  052123
        071114  046505  050040  053517
        071122  051105  043040  051117
        071130     000
 9429   071131     040  042523  047503  MSG053: .ASCIZ  / SECONDS MAX!/
        071136  042116  020123  040515
        071144  020530     000
 9430   071147     200  047516  020127  MSG054: .ASCIZ  <CRLF>/NOW STARTING READ TEST OF MEMORY BANKS/
        071154  052123  051101  044524
        071162  043516  051040  040505
        071170  020104  042524  052123
        071176  047440  020106  042515
        071204  047515  054522  041040
        071212  047101  051513     000
 9431   071217     200  047523  026502  MSG055: .ASCIZ  <CRLF>/SOB-A-LONG TEST/
        071224  026501  047514  043516
        071232  052040  051505  000124
 9432   071240  041200  046105  020114  MSG056: .ASCIZ  <CRLF>/BELL = EACH PASS COMPLETE/
        071246  020075  040505  044103
        071254  050040  051501  020123
        071262  047503  050115  042514
        071270  042524     000
 9433   071273     040  020077     000  MSG057: .ASCIZ  / ? /
 9434   071277                          MSG058: ;SPARE SPARE SPARE SPARE SPARE SPARE SPARE SPARE SPARE SPARE
 9435   071277     200  052523  042520  MSG059: .ASCIZ  <CRLF>/SUPER TIGHT SCOPE LOOP/
        071304  020122  044524  044107
        071312  020124  041523  050117
        071320  020105  047514  050117
        071326     000
 9436   071327     200  053523  042522  MSG060: .ASCIZ  <CRLF>/SWREG ---> MEM, MEM ---> DISPLAY/
        071334  020107  026455  037055
        071342  046440  046505  020054
        071350  042515  020115  026455
        071356  037055  042040  051511
        071364  046120  054501     000
 9437   071371     077  037477  037477  MSG061: .ASCIZ  /??????/
        071376  000077
 9438   071400  047111  052520  020124  MSG062: .ASCIZ  /INPUT MUST BE A/
        071406  052515  052123  041040
        071414  020105  000101
```

```
 9439  071420  020116  041517  040524  MSG063: .ASCIZ  /N OCTAL /
       071426  020114     000
 9440  071431     116  046525  042502  MSG064: .ASCIZ  /NUMBER/<CRLF>
       071436  100122     000
 9441  071441     040  042504  044503  MSG065: .ASCIZ  / DECIMAL /
       071446  040515  020114     000
 9442  071453     200  051105  047522  MSG066: .ASCIZ  <CRLF>/ERROR COUNT EXCEEDED 20 - ABORTING FOR XXDP CHAIN/
       071460  020122  047503  047125
       071466  020124  054105  042503
       071474  042105  042105  031040
       071502  020060  020055  041101
       071510  051117  044524  043516
       071516  043040  051117  054040
       071524  042130  020120  044103
       071532  044501  000116
 9443  071536  040506  040524  020114  MSG067: .ASCIZ  /FATAL /
       071544     000
 9444  071545     115  030512  000061  MSG068: .ASCIZ  /MJ11/
 9445  071552  045515  030461     000  MSG069: .ASCIZ  /MK11/
 9446  071557     113  053440  051117  MSG070: .ASCIZ  /K WORDS OF MEMORY/<CRLF>
       071564  051504  047440  020106
       071572  042515  047515  054522
       071600  000200
 9447  071602  020113  043117  046440  MSG071: .ASCIZ  /K OF MJ11/<CRLF>
       071610  030512  100061     000
 9448  071615     113  047440  020106  MSG072: .ASCIZ  /K OF MK11/<CRLF>
       071622  045515  030461  000200
 9449  071630  051200  043105  042522  MSG073: .ASCIZ  <CRLF>/REFRESH TEST/
       071636  044123  052040  051505
       071644  000124
 9450                                  .EVEN   ;THIS STAYS WITH MSG074
 9451  071646  037477     000  MSG074: .ASCIZ  /??/
 9452  071651     200  042522  047514  MSG075: .ASCIZ  <CRLF>/RELOCATION NOT POSSIBLE/
       071656  040503  044524  047117
       071664  047040  052117  050040
       071672  051517  044523  046102
       071700  000105
 9453  071702  020200  041040  047101  MSG076: .ASCIZ  <CRLF>/  BANK  ERRORS/<CRLF>
       071710  020113  042440  051122
       071716  051117  100123     000
 9454  071723     200  047105  020104  MSG077: .ASCIZ  <CRLF>/END PASS #/
       071730  040520  051523  021440
       071736     000
 9455  071737     200  040527  047122  MSG078: .ASCIZ  <CRLF>/WARNING!  ECC IS DISABLED ON BOX /
       071744  047111  020507  020040
       071752  041505  020103  051511
       071760  042040  051511  041101
       071766  042514  020104  047117
       071774  041040  054117  000040
 9456  072002  042440  051122  051117  MSG079: .ASCIZ  / ERROR(S) DETECTED/<CRLF>
       072010  051450  020051  042504
       072016  042524  052103  042105
       072024  000200
 9463  072161     200  044506  046114  MSG085: .ASCIZ  <CRLF>/FILL COUNT(OCTAL)? /
       072166  041440  052517  052116
       072174  047450  052103  046101
```

```
        072202  037451  000040
 9471   072260  045600  051105  042516  MSG088: .ASCIZ  <CRLF>/KERNEL STACK/
        072266  020114  052123  041501
        072274  000113
 9472   072276  051600  050125  051105  MSG089: .ASCIZ  <CRLF>/SUPERVISOR STACK/
        072304  044526  047523  020122
        072312  052123  041501  000113
 9473   072320  052600  042523  020122  MSG090: .ASCIZ  <CRLF>/USER STACK/
        072326  052123  041501  000113
 9474   072334  044440  020123  046505  MSG091: .ASCIZ  / IS EMPTY/
        072342  052120  000131
 9475   072346  042522  047514  040503  MSG092: .ASCIZ  /RELOCATED  /
        072354  042524  020104  000040
 9476   072362  040502  045516  000075  MSG093: .ASCIZ  /BANK=/
 9477   072370  020040  040515  036522  MSG094: .ASCIZ  /  MAR=/
        072376     000
 9478   072377     040  050040  052101  MSG095: .ASCIZ  /  PAT=/
        072404  000075
 9479   072406  046200  040517  042504  MSG096: .ASCIZ  <CRLF>/LOADERS BEING DESTROYED/
        072414  051522  041040  044505
        072422  043516  042040  051505
        072430  051124  054517  042105
        072436     000
 9480   072437     200  040527  047122  MSG097: .ASCIZ  <CRLF>/WARNING!  ILLEGAL INTERLEAVE SELECTED (FIELD SERVICE ONLY)/
        072444  047111  020507  020040
        072452  046111  042514  040507
        072460  020114  047111  042524
        072466  046122  040505  042526
        072474  051440  046105  041505
        072502  042524  020104  043050
        072510  042511  042114  051440
        072516  051105  044526  042503
        072524  047440  046116  024531
        072532     000
 9481           072534                          .EVEN
 9482                                           .LIST   BEX
 9483                                           .TITLE ODT (CEMKA)
 9484
 9485           072726                          .=.+144+26      ;RESERVES SPACE FOR STACK
 9486
 9487           000000                  R0      =       %0      ; REGISTER
 9488           000001                  R1      =       %1      ;  NAMING
 9489           000002                  R2      =       %2      ;   CONVENTIONS
 9490           000003                  R3      =       %3
 9491           000004                  R4      =       %4
 9492           000005                  R5      =       %5
 9493           000006                  SP      =       %6
 9494           000007                  PC      =       %7
 9495           177776                  ST      =       177776          ;STATUS REGISTER
 9496
 9497           000016                  O.BKP   =       16              ;NUMBER OF BREAKPOINTS-1 MULT. BY 2
 9498           000014                  O.TVEC  =       14              ;TRT VECTOR LOCATION
 9499           000340                  O.STM   =       340             ;PRIORITY MASK - STATUS REGISTER
 9500           000020                  O.TBT   =       20              ;T-BIT MASK - STATUS REGISTER
 9501           000003                  TRT     =       000003          ;TRT INSTRUCTION
 9502
```

```
9503
9504    177562          0.RDB   =       177562  ;R DATA BUFFER
9505    177560          0.RCSR  =       177560  ;R C/SR
9506    177566          0.TDB   =       177566  ;T DATA BUFFER
9507    177564          0.TCSR  =       177564  ;T C/SR
```

```
9509                                    ; INITIALIZE ODT
9510                                    ;  USE 0.ODT FOR A NORMAL ENTRY
9511                                    ;  USE 0.ODT+2 TO RESTART ODT - WIPING OUT ALL BREAKPOINTS
9512                                    ;  USE 0.ODT+4 TO RE-ENTER (I.E. - FAKE A BREAKPOINT)
9513
9514  072726  000421                  0.ODT:  BR      2$                      ;NORMAL ENTRY
9515  072730  000440                          BR      3$                      ;RESTART
9516  072732  004737  076124          1$:     JSR     PC,0.RRST               ;RE-ENTER  --  SAVE STATUS
9517  072736  013746  000016                  MOV     0.TVEC+2,-(SP)          ;SET UP LOCAL STATUS
9518  072742  004737  076134                  JSR     PC,0.WST
9519  072746  010746                          MOV     PC,-(SP)                ;FAKE THE PC
9520  072750  012637  072600                  MOV     (SP)+,0.UPC
9521  072754  112737  177777  076245          MOVB    #-1,0.P                 ;DISALLOW PROCEED
9522  072762  105037  076243                  CLRB    0.S
9523  072766  000137  075012                  JMP     0.BK1
9524
9525                                          ;RUN AT CURRENT STATUS
9526  072772  004537  075432          2$:     JSR     5,0.SVTT                ;SAVE TTY STATUS
9527  072776  012737  072732  056220          MOV     #1$,@#CONTP+2
9528  073004  004037  075312                  JSR     0,0.SVR                 ;SAVE REGISTERS (MAINLY SP)
9529  073010  012706  072562                  MOV     #0.URO,SP               ;SET UP STACK
9530  073014  012704  076253                  MOV     #0.ID,R4                ;TYPE ID
9531  073020  012703  076270                  MOV     #0.IDND,R3
9532  073024  004537  076060                  JSR     5,0.TYPE
9533  073030  000414                          BR      4$
9534  073032  004037  075312          3$:     JSR     0,0.SVR                 ;SAVE REGISTERS
9535  073036  004537  075534                  JSR     5,0.REM                 ;REMOVE ALL BREAKPOINTS
9536  073042  113704  072604                  MOVB    0.PRI,R4                ;GET ODT PRIORITY
9537  073046  106004                          RORB    R4                      ;SHIFT
9538  073050  106004                          RORB    R4                      ;  INTO
9539  073052  106004                          RORB    R4                      ;    POSITION
9540  073054  010446                          MOV     R4,-(SP)                ;STORE IN STATUS
9541  073056  004737  076134                  JSR     PC,0.WST
9542  073062  105037  076243          4$:     CLRB    0.S                     ;DISABLE SINGLE INSTRUCTION FOR NOW
9543  073066  112737  177777  076245          MOVB    #-1,0.P                 ;DISALLOW PROCEED
9544  073074  012737  000340  000016          MOV     #0.STM,0.TVEC+2         ;STATUS WORD TO TRT VECTOR+2
9545  073102  012737  074774  000014          MOV     #0.BRK,0.TVEC           ;PC TO TRT VECTOR
9546  073110  000137  074362                  JMP     0.RALL                  ;CLEAR BRK PT TABLES
9547                                    ;
9548                                    ;0.CTLC
9549                                    ;       ↑C PROCESSING. RETURN TO KEYBOARD MONITOR
9550                                    ;
9551                                    ;
9552  073114  013706  072576          0.CTLC: MOV     0.USP,SP                ;RESTORE USER STACK
9553  073120  012704  073140                  MOV     #1$,R4
9554  073124  012703  073141                  MOV     #2$,R3
9555  073130  004537  076060                  JSR     R5,0.TYPE               ;ECHO "↑C"
9556  073134  000137  037020                  JMP     BOOT
9557  073140     136                  1$:     .BYTE   '↑
9558  073141     103                  2$:     .BYTE   'C
9559
9560                                    ;CONTROL F PROCESSING
9561  073142  013706  072576          0.CTLF: MOV     0.USP,SP                ;RESTORE USER STACK
9562  073146  012704  073172                  MOV     #1$,R4
9563  073152  012703  073173                  MOV     #2$,R3
9564  073156  004537  076060                  JSR     R5,0.TYPE               ;ECHO ↑F
```

```
 9565  073162  004737  037372                  CALL    FIELDSERVICE
 9566  073166  000137  072726                  JMP     0.ODT
 9567  073172     136                  1$:     .BYTE   'F
 9568  073173     106                  2$:     .BYTE   'F
 9569
 9570                                          ;CONTROL E PROCESSING
 9571  073174  012704  073220          0.CTLE: MOV     #1$,R4
 9572  073200  012703  073221                  MOV     #2$,R3
 9573  073204  004537  076060                  JSR     R5,0.TYPE       ;ECHO 'E
 9574  073210  012705  056222                  MOV     #CONTP+4,R5
 9575  073214  000137  074556                  JMP     0.GOGO
 9576  073220     136                  1$:     .BYTE   'F
 9577  073221     105                  2$:     .BYTE   'E
 9578
 9579                                          ; SPECIAL NAME HANDLER
 9580                                          ;  DEPENDS UPON THE EXPLICIT ORDER OF THE TWO TABLES 0.TL AND 0.URO
 9581
 9582  073222  004537  075740          0.REGT: JSR     5,0.GET         ;SPECIAL NAME, GET ONE MORE CHARACTER
 9583  073226  012704  076321                  MOV     #0.TL,R4        ;TABLE START ADDRESS
 9584  073232  120024                  1$:     CMPB    R0,(R4)+        ;IS THIS THE CORRECT CHARACTER?
 9585  073234  001414                          BEQ     3$              ;JUMP IF YES
 9586  073236  022704  076341                  CMP     #0.TL+0.LG,R4   ;IS THE SEARCH DONE?
 9587  073242  101373                          BHI     1$              ;BRANCH IF NOT
 9588  073244  042700  177770                  BIC     #177770,R0      ;MASK OFF OCTAL
 9589  073250  010004                          MOV     R0,R4
 9590  073252  006304                  2$:     ASL     R4
 9591  073254  062704  072562                  ADD     #0.URO,R4       ;GENERATE ADDRESS
 9592  073260  005202                          INC     R2              ;SET FOUND FLAG
 9593  073262  000137  073506                  JMP     0.SCAN          ;GO FIND NEXT CHARACTER
 9594  073266  162704  076312          3$:     SUB     #0.TL-7,R4      ;GO FIND NEXT CHARACTER
 9595  073272  000767                          BR      2$
```

```
 9597                                   ; "BACKARROW"  HANDLER - OPEN INDEXED ON THE PC (BACK ARROW)
 9598                                           .ENABL  LSB
 9599  073274  004537  073350          0.ORPC: JSR     5,2$    ;TEST WORD MODE AND CLOSE
 9600  073300  061202                          ADD     @R2,R2          ;COMPUTE
 9601  073302  005202                          INC     R2
 9602  073304  005202                          INC     R2              ;    NEW ADDRESS
 9603  073306  010237  076232          1$:     MOV     R2,0.CAD        ;UPDATE CAD
 9604  073312  000137  074160                  JMP     0.OP2A          ;GO FINISH UP
 9605  073316  004537  073350          0.ORAB: JSR     5,2$            ;TEST WORD MODE AND CLOSE
 9606  073322  011202                          MOV     @R2,R2          ;GET ABSOLUTE ADDRESS
 9607  073324  000770                          BR      1$
 9608  073326  004537  073350          0.ORRB: JSR     5,2$            ;TEST AND CLOSE
 9609  073332  011201                          MOV     @R2,R1          ;COMPUTE NEW ADDRESS
 9610  073334  110101                          MOVB    R1,R1           ;EXTEND THE SIGN
 9611  073336  006301                          ASL     R1              ;R2=2(@R2)
 9612  073340  005201                          INC     R1              ;    +2
 9613  073342  005201                          INC     R1
 9614  073344  060102                          ADD     R1,R2           ;    +PC
 9615  073346  000757                          BR      1$
 9616  073350  004737  076146          2$:     JSR     PC,0.CLSE       ;CLOSE CURRENT CELL
 9617  073354  022737  000002  076240          CMP     #2,0.BW         ;ONLY WORD MODE ALLOWED
 9618  073362  001003                          BNE     3$              ;BRANCH IF ERROR
 9619  073364  013702  076232                  MOV     0.CAD,R2        ;CURRENT ADDRESS IN R2
 9620  073370  000205                          RTS     R5
 9621  073372  005726                  3$:     TST     (SP)+
 9622  073374  000137  073442          0.TCL2: JMP     0.ERR           ;POP A WORD AND SHOW THE ERROR
 9623                                           .DSABL  LSB
 9624
 9625                                   ;       PROCESS S - SINGLE INSTRUCTION MODE
 9626  073400  105737  076251          0.SNGL: TSTB    0.SMFD  ;DONT REACT IF ; NOT TYPED
 9627  073404  001773                          BEQ     0.TCL2
 9628  073406  005702                          TST     R2              ;SEE IF TURN ON OR TURN OFF
 9629  073410  001004                          BNE     1$              ;BRANCH IF TURNING IT ON
 9630  073412  105037  076243                  CLRB    0.S             ;CLEAR THE FLAG
 9631  073416  000137  073452                  JMP     0.DCD           ;CONTINUE THE SCAN
 9632  073422  112737  177777  076243  1$:     MOVB    #-1,0.S         ;SET THE FLAG
 9633  073430  000137  073452                  JMP     0.DCD
```

```
                                        ;
                                        ;MINUS PROCESSING
                                        ;
073434  105237  076271        0.MIN:  INCB    0.MINS          ;SET MINUS FOUND SWITCH ON
073440  000420                        BR      0.DCD1
                                        ; COMMAND DECODER - ODT11X
                                        ;  ALL REGISTERS MAY BE USED (R0-R5),
073442  012700  000077        0.ERR:  MOV     #'?,R0          ;  ? TO BE TYPED
073446  004537  075662                JSR     5,0.FTYP        ; OUTPUT ?
073452  005037  076240        0.DCD:  CLR     0.BW            ;CLOSE ALL
073456  004537  076206                JSR     5,0.CRLS        ;TYPE  <CR><LF>*
073462  105037  076251        0.DCD3: CLRB    0.SMFD          ;SEMICOLON FOUND FLAG
073466  105037  076250                CLRB    0.CMFD          ;COMMA FOUND FLAG
073472  105037  076271                CLRB    0.MINS          ;MINUS SIGN FOUND FLAG
073476  005003                        CLR     R3              ;R3 IS A SAVE REGISTER FOR R2
073500  005005                        CLR     R5              ;R5 IS A SAVE REGISTER FOR R4
073502  005004        0.DCD1: CLR     R4              ; R4 CONTAINS THE CONVERTED OCTAL
073504  005002                CLR     R2              ; R2 IS THE NUMBER FOUND FLAG
073506  004537  075740        0.SCAN: JSR     5,0.GET         ;GET A CHAR, RETURN IN R0
073512  022700  000060                CMP     #'0,R0          ;COMPARE WITH ASCII 0
073516  101013                        BHI     5$              ;CHECK LEGALITY IF NON-NUMERIC
073520  022700  000067                CMP     #'7,R0          ;COMPARE WITH ASCII 7
073524  103410                        BLO     5$              ;CHECK LEGALITY IF NOT OCTAL
073526  042700  177770                BIC     #177770,R0      ;CONVERT TO BCD
073532  006304                        ASL     R4              ; MAKE ROOM
073534  006304                        ASL     R4              ;   IN
073536  006304                        ASL     R4              ;     R4
073540  060004                        ADD     R0,R4           ;PACK THREE BITS IN R4
073542  005202                        INC     R2              ;R2 HAS NUMERIC FLAG
073544  000760                        BR      0.SCAN          ;   AND TRY AGAIN
073546  005001        5$:     CLR     R1              ;CLEAR INDEX
073550  120061  076275    1$:     CMPB    R0,0.LGCH(R1)   ;DO THE CODES MATCH?
073554  001405                        BEQ     2$              ;JUMP IF YES
073556  005201                        INC     R1              ; SET INDEX FOR NEXT SEARCH
073560  020127  000024                CMP     R1,#0.CLGT      ;IS THE SEARCH DONE?
073564  103326                        BHIS    0.ERR           ;   OOPS!
073566  000770                        BR      1$              ;RE-LOOP
073570  105737  076271    2$:     TSTB    0.MINS          ;IF MINUS WAS NOT TYPED
073574  001401                        BEQ     4$              ;DO NOT NEGATE K
073576  005404                        NEG     R4              ;OTHERWISE, TAKE 2'S COMPLEMENT.
073600  105737  076250    4$:     TSTB    0.CMFD          ;IF A COMMA NOT TYPED,
073604  001402                        BEQ     3$              ;SKIP NEXT INSTRUCTION.
073606  063704  076344                ADD     0.BIAS,R4       ;OTHERWISE, ADD RELOC. BIAS TO (R4)
073612  105037  076271    3$:     CLRB    0.MINS          ;REINITIALIZE MINUS-TYPED SWITCH FOR NXT SCAN
073616  006301                        ASL     R1              ;MULTIPLY BY TWO
073620  000171  073624                JMP     @6$(R1) ;GO TO PROPER ROUTINE

073624  073674        6$:     0.SEMI  ;  ; SEMICOLON
073626  073712                0.WRD   ;  /     OPEN WORD
073630  073724                0.BYT   ;  \     OPEN BYTE -BACK SLASH
073632  074072                0.CRET  ;  CARRIAGE RETURN     CLOSE
073634  073222                0.REGT  ;  $  REGISTER OPS
073636  074530                0.GO    ;  G  GO TO ADDRESS K
```

```
 9691  073640  074112
 9692  073642  073274
 9693  073644  074106
 9694  073646  074234
 9695  073650  074420
 9696  073652  074256
 9697  073654  074664
 9698  073656  073316
 9699  073660  073326
 9700  073662  073400
 9701  073664  073434
 9702  073666  073114
 9703  073670  073142
 9704  073672  073174

O.OP1   :  <LF>  MODIFY, CLOSE, OPEN NEXT
O.ORPC  :  "BACKARROW"   OPEN RELATED, INDEX - PC (BACK ARROW)
O.OLD   :  <   RETURN TO OLD SEQUENCE AND OPEN
O.BACK  :  ↑   OPEN PREVIOUS (UP ARROW)
O.OFST  :  O   OFFSET
O.BKPT  :  B   BREAKPOINTS
O.PROC  :  P   PROCEED
O.ORAB  :  @   OPEN RELATED, ABSOLUTE
O.ORRB  :  >   OPEN RELATED, RFL. BRANCH
O.SNGL  :  S   SINGLE INSTRUCTION MODE
O.MIN   :  -   MINUS, NEGATES NUMBER TYPED IN
O.CTLC  :  ↑C  EXIT TO MONITOR
O.CTLF  :  ↑F  EXIT TO FIELD SERVICE MODE
O.CTLE  :  ↑E  PROCEED FROM ↑D
```

```
 9706
 9707                                   ; SEMI-COLON PROCESSOR
 9708
 9709  073674  010203                   O.SEMI: MOV     R2,R3    ;A SEMI-COLON HAS BEEN RECEIVED
 9710  073676  010405                           MOV     R4,R5    ;NUMERIC FLAG TO R3, CONTENTS TO R5
 9711  073700  105237  076251                   INCB    O.SMFD   ;SET SEMICOLON FOUND FLAG
 9712  073704  105037  076250                   CLRB    O.CMFD   ;RESET COMMA FOUND FLAG
 9713  073710  000674                           BR      O.DCD1           ;GO BACK FOR MORE
 9714
 9715                                   ; PROCESS / AND \ - OPEN WORD OR BYTE
 9716
 9717                                   ;INPUT - IF R2 IS NON-ZERO A NEW REXP HAS BEEN TYPED IN
 9718                                   ;INPUT - -ADDRESS OF WORD TO BE OPENED IS IN R4
 9719                                           .ENABL  LSB
 9720  073712  012737  000002  076240   O.WRD:  MOV     #2,O.BW          ;OPEN WORD
 9721  073720  000404                           BR      11$
 9722  073722  006104                   1$:     ROL     R4               ;GET THE ADDRESS BACK
 9723  073724  012737  000001  076240   O.BYT:  MOV     #1,O.BW          ;OPEN BYTE
 9724  073732  005702                   11$:    TST     R2               ;GET VALUE IF R2 IS NON-ZERO
 9725  073734  001011                           BNE     14$              ;BRANCH IF NUMBER INPUT
 9726  073736  105737  076250                   TSTB    O.CMFD           ;TEST FOR ","AND";"
 9727  073742  001402                           BEQ     12$
 9728  073744  000137  073442           13$:    JMP     O.ERR            ;ERROR IF PRESENT WITHOUT NUMBER.
 9729  073750  105737  076251           12$:    TSTB    O.SMFD
 9730  073754  001373                           BNE     13$
 9731  073756  000404                           BR      O.WRD1           ;NO NUMBER - REOPEN PREVIOUS LOCATION
 9732  073760  010437  076234           14$:    MOV     R4,O.DOT         ;PUT VALUE IN DOT
 9733  073764  010437  076232                   MOV     R4,O.CAD         ;    ALSO IN CAD
 9734  073770  022737  000001  076240   O.WRD1: CMP     #1,O.BW          ;CHECK BYTE MODE
 9735  073776  001407                           BEQ     22$              ;JUMP IF BYTE
 9736  074000  013704  076232                   MOV     O.CAD,R4
 9737  074004  006204                           ASR     R4               ;MOVE ONE BIT TO CARRY
 9738  074006  103745                           BCS     1$               ;JUMP IF ODD ADDRESS
 9739  074010  017700  002216                   MOV     @O.CAD,R0        ;GET CONTENTS OF WORD
 9740  074014  000402                           BR      23$
 9741  074016  117700  002210           22$:    MOVB    @O.CAD,R0        ;GET CONTENTS OF BYTE
 9742  074022  004537  075566           23$:    JSR     5,O.CADV         ;GO GET AND TYPE OUT @CAD
 9743  074026  022737  000001  076240           CMP     #1,O.BW          ;CHECK IF BYTE MODE.
 9744  074034  001212                           BNE     O.DCD3           ;IF NOT WE'RE DONE. ELSE:
 9745  074036  112700  000075                   MOVB    #'=,R0   ;TYP "=" AND THEN THE ASCII BYTE
 9746  074042  004537  075662                   JSR     5,O.FTYP
 9747  074046  117700  002160                   MOVB    @O.CAD,R0
 9748  074052  004537  075662                   JSR     5,O.FTYP
 9749  074056  112700  000040                   MOVB    #' ,R0
 9750  074062  004537  075662                   JSR     5,O.FTYP
 9751  074066  000137  073462                   JMP     O.DCD3           ;GO BACK TO DECODER
 9752                                           .DSABL  LSB
 9753
 9754                                   ; PROCESS CARRIAGE RETURN
 9755  074072  004737  076146           O.CRET: JSR     PC,O.CLSE        ;CLOSE LOCATION
 9756  074076  000137  073452           O.DCDA: JMP     O.DCD            ;RETURN TO DECODER
 9757
 9758  074102  000137  073442           O.ERR3: JMP     O.ERR            ; INTERMEDIATE HELP
```

```
9760                                    ; PROCESS <LF>, OPEN NEXT WORD
9761
9762  074106  105237  076242            O.OLD:  INCB    O.SEQ   ;SET FLAG TO LATER RESTORE CAD
9763  074112  005737  076240            O.OP1:  TST     O.BW            ;<LF> RECEIVED
9764  074116  001771                    O.ERR2: BEQ     O.ERR3          ;ERROR IF NOTHING IS OPEN
9765  074120  004737  076146                    JSR     PC,O.CLSE       ;CLOSE PRESENT CELL
9766  074124  105737  076242                    TSTB    O.SEQ           ;SHOULD CAD BE RESTORED?
9767  074130  001405                            BEQ     1$              ;BRANCH IF NOT
9768  074132  013737  076234  076232            MOV     O.DOT,O.CAD     ;RESTORE PREVIOUS SEQUENCE
9769  074140  105037  076242                    CLRB    O.SEQ           ;RESET FLAG; NO LONGER NEEDED
9770  074144  063737  076240  076232  1$:       ADD     O.BW,O.CAD      ;GENERATE NEW ADDRESS
9771  074152  013737  076232  076234  O.OP2:    MOV     O.CAD,O.DOT     ;INITIALIZE DOT
9772  074160  004537  076200          O.OP2A:   JSR     5,O.CRLF        ;<CR><LF>
9773  074164  013746  076240                    MOV     O.BW,-(SP)      ;SAVE BW
9774  074170  012737  000002  076240            MOV     #2,O.BW         ;SET TO TYPE FULL WORD ADDRESS
9775  074176  013700  076232                    MOV     O.CAD,R0        ;NUMBER TO TYPE
9776  074202  004537  076224                    JSR     5,O.RORA        ; CHECK FORMAT
9777  074206  011637  076240                    MOV     @SP,O.BW        ;RESTORE BW
9778  074212  012700  027534                    MOV     #27534,R0       ;PUT "/\" IN R0
9779  074216  022726  000001                    CMP     #1,(SP)+        ;IS IT BYTE MODE?
9780  074222  001401                            BEQ     1$              ;JUMP IF YES
9781  074224  000300                            SWAB    R0              ;TYPE A /
9782  074226  004537  075662          1$:       JSR     5,O.FTYP        ;TYPE THE LOW BYTE OF R0
9783  074232  000656                            BR      O.WRD1          ;GO PROCESS IT
9784
```

```
9786
9787                                    ; PROCESS ↑, OPEN PREVIOUS WORD
9788
9789  074234  005737  076240            O.BACK: TST     O.BW            ;   ↑ RECEIVED
9790  074240  001726                            BEQ     O.ERR2          ;ERROR IF NOTHING OPEN
9791  074242  004737  076146                    JSR     PC,O.CLSE
9792  074246  163737  076240  076232            SUB     O.BW,O.CAD      ;GENERATE NEW ADDRESS
9793  074254  000736                            BR      O.OP2           ;GO DO THE REST
9794
9795                                    ; B HANDLER - SET AND REMOVE BREAKPOINTS
9796
9797  074256  012700  076346            O.BKPT: MOV     #O.TRTC,R0
9798  074262  006304                            ASL     R4              ;MULTIPLY NUMBER BY TWO
9799  074264  005703                            TST     R3
9800  074266  001423                            BEQ     3$              ;IF R3 IS ZERO GO REMOVE BREAKPOINT
9801  074270  006205                            ASR     R5              ;GET ONE BIT TO CARRY
9802  074272  103514                            BCS     O.ERR1          ;BADNESS IF ODD ADDRESS
9803  074274  006305                            ASL     R5              ;RESTORE ONE BIT
9804  074276  062704  072606                    ADD     #O.ADR1,R4
9805  074302  005702                            TST     R2
9806  074304  001007                            BNE     1$              ;JUMP IF SPECIFIC CELL
9807  074306  020014                    2$:     CMP     R0,@R4          ;IS THIS CELL FREE?
9808  074310  001405                            BEQ     1$              ;JUMP IF YES
9809  074312  020427  072624                    CMP     R4,#O.BKP+O.ADR1        ;ARE WE AT THE END OF OUR ROPE
9810  074316  103102                            BHIS    O.ERR1          ;YES, THERE IS NOTHING FREE
9811  074320  005724                            TST     (R4)+           ;INCREMENT BY TWO
9812  074322  000771                            BR      2$
9813  074324  020427  072624            1$:     CMP     R4,#O.BKP+O.ADR1
9814  074330  101075                            BHI     O.ERR1          ;ERROR IF TOO LARGE
9815  074332  010514                            MOV     R5,@R4          ;SET BREAKPOINT
9816  074334  000660                            BR      O.DCDA          ;RETURN
9817  074336  005702                    3$:     TST     R2
9818  074340  001410                            BEQ     O.RALL          ;GO REMOVE ALL
9819  074342  020427  000016                    CMP     R4,#O.BKP
9820  074346  101066                            BHI     O.ERR1          ;JUMP IF NUMBER TOO LARGE
9821  074350  010064  072606                    MOV     R0,O.ADR1(R4)   ;CLEAR BREAKPOINT
9822  074354  005064  072630                    CLR     O.CT(R4)        ;CLEAR COUNT ALSO
9823  074360  000646                            BR      O.DCDA
9824  074362  005004                    O.RALL: CLR     R4
9825  074364  012700  076346                    MOV     #O.TRTC,R0
9826  074370  020427  000020            1$:     CMP     R4,#O.BKP+2     ;ALL DONE?
9827  074374  101240                            BHI     O.DCDA          ;JUMP IF YES
9828  074376  010064  072606                    MOV     R0,O.ADR1(R4)   ;RESET BKPT
9829  074402  012764  000003  072652            MOV     #TRT,O.UIN(R4)  ;RESET CONTENTS OF TABLE
9830  074410  005064  072630                    CLR     O.CT(R4)        ;CLEAR COUNT
9831  074414  005724                            TST     (R4)+           ;INCREMENT BY TWO
9832  074416  000764                            BR      1$
9833
```

```
9835
9836                                    ; PROCESS 0, COMPUTE OFFSET
9837
9838  074420  022737  000002  076240  0.OFST: CMP     #2,0.BW         ;CHECK WORD MODE
9839  074426  001036                          BNE     0.ERR1          ;ERROR IF NOT CORRECT MODE
9840  074430  012700  000040                  MOV     #' ,R0          ;TYPE ONE BLANK
9841  074434  004537  075662                  JSR     5,0.FTYP        ;   AS A SEPARATOR
9842  074440  005703                          TST     R3              ;WAS SEMI-COLON TYPED?
9843  074442  001430                          BEQ     0.ERR1          ;NO, CALL IT AN ERROR
9844  074444  163705  076232                  SUB     0.CAD,R5        ;COMPUTE
9845  074450  005305                          DEC     R5
9846  074452  005305                          DEC     R5              ;     16 BIT OFFSET
9847  074454  010500                          MOV     R5,R0
9848  074456  004537  075566                  JSR     5,0.CADV        ;NUMBER IN R0 - WORD MODE
9849  074462  010500                          MOV     R5,R0
9850  074464  006200                          ASR     R0              ;DIVIDE BY TWO
9851  074466  103414                          BCS     1$              ;ERROR IF ODD
9852  074470  022700  177600                  CMP     #-200,R0        ;COMPARE WITH -200
9853  074474  003011                          BGT     1$              ;DO NOT TYPE IF OUT OF RANGE
9854  074476  022700  000177                  CMP     #177,R0         ;COMPARE WITH +177
9855  074502  002406                          BLT     1$              ;DO NOT TYPE IF OUT OF RANGE
9856  074504  005337  076240                  DEC     0.BW            ;SET TEMPORARY BYTE MODE
9857  074510  004537  075566                  JSR     5,0.CADV        ;NUMBER IN R0 - BYTE MODE
9858  074514  005237  076240                  INC     0.BW            ;RESTORE WORD MODE
9859  074520  000137  073462          1$:     JMP     0.DCD3          ;ALL DONE
9860
9861  074524  000137  073442          0.ERR1: JMP     0.ERR           ;INTERMEDIATE HELP
9862
9863                                    ; PROCESS G - GO
9864
9865  074530  105737  076251          0.GO:   TSTB    0.SMFD  ;WAS ";" TYPED?
9866  074534  001773                          BEQ     0.ERR1  ;BR IF NOT TYPED
9867  074536  005703                          TST     R3              ;WAS K; TYPED?
9868  074540  001771                          BEQ     0.ERR1          ; TYPE ?<CR,LF> IF NOT
9869  074542  112737  000021  076245          MOVB    #0.BKP+3,0.P    ;CLEAR PROCEED
9870  074550  006205                          ASR     R5              ;CHECK LOW ORDER BIT
9871  074552  103764                          BCS     0.ERR1          ;ERROR IF ODD NUMBER
9872  074554  006305                          ASL     R5              ;RESTORE WORD
9873  074556  010537  072600          0.GOGO: MOV     R5,0.UPC        ;SET UP NEW PC
9874  074562  012746  000340                  MOV     #0.STM,-(SP)    ;SET HIGH PRIORITY
9875  074566  004737  076134                  JSR     PC,0.WST
9876  074572  004537  075466                  JSR     5,0.RSTT        ;RESTORE TELETYPE
9877  074576  105037  076244          0.TBIT: CLRB    0.T             ;CLEAR
9878  074602  052737  000020  072602          BIS     #0.TBT,0.UST    ;     BOTH T-BIT FLAGS
9879  074610  105737  076243                  TSTB    0.S             ;SEE IF WE NEED A T BIT
9880  074614  001005                          BNE     0.GO2           ;IF NOT GO NOW
9881  074616  042737  000020  072602          BIC     #0.TBT,0.UST    ;SET THE T BIT  ;??????????????????
9882  074624  004537  075402                  JSR     5,0.RSB         ;RESTORE BREAKPOINTS
9883  074630  004037  075350          0.GO2:  JSR     0,0.RSR         ;RESTORE REGISTERS
9884  074634  013746  072602                  MOV     0.UST,-(SP)     ;     AND STATUS
9885  074640  013746  072600                  MOV     0.UPC,-(SP)     ;     AND PC
9886  074644  000240                          NOP                     ; CHANGE TO HALT FOR DEBUGGING
9887  074646  013746  072602                  MOV     0.UST,-(SP)     ; MOV IN STATUS FIRST W/O T BIT
9888  074652  042716  177420                  BIC     #0.TBT!177400,(SP)      ;  SO INTERRUPTS CAN HAPPEN BEFORE
9889  074656  004737  076134                  JSR     PC,0.WST        ;  RTT TURNS ON THE T BIT.
9890  074662  000006                          RTT
```

```
9891
9892                                    ; PROCESS P - PROCEED
9893                                    ;   ONLY ALLOWED AFTER A BREAKPOINT
9894
9895  074664  105737  076251            0.PROC: TSTB    0.SMFD  ;WAS ";" TYPED?
9896  074670  001715                            BEQ     0.ERR1  ;BR IF NOT TYPED
9897  074672  113700  076245                    MOVB    0.P,R0
9898  074676  105700                            TSTB    R0              ;CHECK LEGALITY OF PROCEED
9899  074700  002711                            BLT     0.ERR1          ;NOT LEGAL
9900  074702  005702                            TST     R2              ;CHECK FOR ILLEGAL COUNT
9901  074704  001307                            BNE     0.ERR1          ;JUMP IF ILLEGAL
9902  074706  005703                            TST     R3              ;WAS COUNT SPECIFIED?
9903  074710  001402                            BEQ     0.PR1           ;NO
9904  074712  010560  072630                    MOV     R5,0.CT(R0)     ;YES, PUT AWAY COUNT
9905  074716  012746  000340            0.PR1:  MOV     #0.STM,-(SP)    ;FORCE HIGH PRIORITY
9906  074722  004737  076134                    JSR     PC,0.WST
9907  074726  004537  075466                    JSR     5,0.RSTT        ;RESTORE TTY
9908  074732  123727  076245  000016    0.C1:   CMPB    0.P,#0.BKP      ;SEE IF A REAL ONE OR A FAKE
9909  074740  003316                            BGT     0.TBIT          ;BRANCH IF FAKE
9910  074742  105737  076243                    TSTB    0.S             ;SEE IF SINGLE INSTRUCTION MODE
9911  074746  001313                            BNE     0.TBIT          ;IF SO EXIT NOW
9912  074750  012746  000340                    MOV     #0.STM,-(SP)    ;SET HIGH PRIORITY
9913  074754  004737  076134                    JSR     PC,0.WST
9914  074760  105237  076244                    INCB    0.T             ;SET T-BIT FLAG
9915  074764  052737  000020  072602            BIS     #0.TBT,0.UST    ;SET T-BIT
9916  074772  000716                            BR      0.GO2
9917
9918
9919                                    ; BREAKPOINT HANDLER
9920  074774  012637  072600            0.BRK:  MOV     (SP)+,0.UPC     ;PRIORITY IS 7 UPON ENTRY
9921  075000  012637  072602                    MOV     (SP)+,0.UST     ;SAVE STATUS AND PC
9922  075004  112737  000021  076245            MOVB    #0.BKP+3,0.P    ;TELL ;P THAT WE CAN CONTINUE
9923  075012  004037  075312            0.BK1:  JSR     0,0.SVR         ;SAVE VARIOUS REGISTERS
9924  075016  105737  076244                    TSTB    0.T             ;CHECK FOR T-BIT SET
9925  075022  001265                            BNE     0.TBIT          ;JUMP IF SET
9926  075024  004537  075534                    JSR     5,0.REM         ;REMOVE BREAKPOINTS
9927  075030  105737  072604                    TSTB    0.PRI           ;CHECK IF PRIORITY
9928  075034  100003                            BPL     2$              ;  IS AS SAME AS USER PGM
9929  075036  113705  072602                    MOVB    0.UST,R5        ;PICK UP USER UST IF SO
9930  075042  000407                            BR      3$
9931  075044  113705  072604            2$:     MOVB    0.PRI,R5        ;OTHERWISE PICK UP ACTUAL PRIORITY
9932  075050  000257                            CCC                     ;CLEAR CARRY
9933  075052  106005                            RORB    R5              ;SHIFT LOW ORDER BITS
9934  075054  106005                            RORB    R5              ;  INTO
9935  075056  106005                            RORB    R5              ;    HIGH ORDER
9936  075060  106005                            RORB    R5              ;      POSITION
9937  075062  042705  177420            3$:     BIC     #0.TBT!177400,R5        ;CLEAR POSSIBLE T BIT (S/I MODE)
9938  075066  010546                            MOV     R5,-(SP)                ;PUT THE STATUS AWAY WHERE IT BELONGS
9939  075070  004737  076134                    JSR     PC,0.WST
9940  075074  013705  072600                    MOV     0.UPC,R5        ;GET PC, IT POINTS TO THE TRT
9941  075100  105737  076243                    TSTB    0.S             ;SEE IF IT WAS SINGLE INSTRUCTION FUN
9942  075104  100432                            BMI     14$             ;IF SO HANDLE THERE
9943  075106  005745                            TST     -(R5)
9944  075110  010537  072600                    MOV     R5,0.UPC
9945  075114  012704  000016                    MOV     #0.BKP,R4       ;GET A COUNTER
9946  075120  020564  072606            11$:    CMP     R5,0.ADR1(R4)   ;COMPARE WITH LIST
```

```
 9947  075124  001426                        BEQ     12$             ;JUMP IF FOUND
 9948  075126  005304                        DEC     R4
 9949  075130  005304                        DEC     R4
 9950  075132  002372                        BGE     11$             ;RE-LOOP UNTIL FOUND
 9951  075134  004537  075432                JSR     5,0.SVTT        ;SAVE TELETYPE STATUS
 9952  075140  004537  076200                JSR     5,0.CRLF
 9953  075144  012704  076342                MOV     #0.BD,R4        ;ERROR, NOTHING FOUND
 9954  075150  012703  076343                MOV     #0.BD+1,R3
 9955  075154  004537  076060                JSR     5,0.TYPE        ;OUTPUT "BE" FOR BAD ENTRY
 9956  075160  010500                        MOV     R5,R0
 9957  075162  062737  000002  072600        ADD     #2,0.UPC        ;POP OVER THE ADJUSTMENT ABOVE
 9958  075170  000444                        BR      13$             ; OR CONTINUE
 9959  075172  112704  000020          14$:  MOVB    #0.BKP+2,R4     ;SET BREAK POINT HIGH + 1
 9960  075176  010564  072606                MOV     R5,0.ADR1(R4)   ;STORE NEXT PC VALUE FOR TYPE OUT
 9961  075202  110437  076245          12$:  MOVB    R4,0.P  ;ALLOW PROCEED
 9962  075206  005364  072630                DEC     0.CT(R4)
 9963  075212  003247                        BGT     0.C1            ;JUMP IF REPEAT
 9964  075214  012764  000001  072630        MOV     #1,0.CT(R4)     ;RESET COUNT TO 1
 9965  075222  004537  075432                JSR     5,0.SVTT        ;SAVE TELETYPE STATUS, R4 IS SAFE
 9966  075226  012700  000102                MOV     #'B,R0
 9967  075232  004537  075662                JSR     5,0.FTYP        ;TYPE "B"
 9968  075236  113700  076245                MOVB    0.P,R0          ;CONVERT BREAKPOINT NUMBER TO ASCII
 9969  075242  062700  000140                ADD     #140,R0
 9970  075246  006200                        ASR     R0
 9971  075250  004537  075662                JSR     5,0.FTYP
 9972  075254  012700  000073                MOV     #';,R0
 9973  075260  004537  075662                JSR     5,0.FTYP        ; TYPE
 9974  075264  012737  000002  076240        MOV     #2,0.BW         ; SET WORD MODE
 9975  075272  113704  076245                MOVB    0.P,R4
 9976  075276  016400  072606                MOV     0.ADR1(R4),R0   ;GET ADDRESS OF BREAK
 9977  075302  004537  076224          13$:  JSR     5,0.RORA        ; CHECK FORMAT
 9978  075306  000137  073452                JMP     0.DCD           ;GO TO DECODER
 9979
 9980
 9981                                  ; SAVE REGISTERS R0-R6
 9982                                  ;   INTERNAL STACK
 9983
 9984  075312  012637  076236    0.SVR:  MOV     (SP)+,0.XXX     ;PICK REGISTER FROM STACK AND SAVE
 9985  075316  010637  072576            MOV     SP,0.USP        ;SAVE USER STACK ADDRESS
 9986  075322  012706  072576            MOV     #0.USP,SP       ;SET TO INTERNAL STACK
 9987  075326  010546                    MOV     R5,-(SP)        ;SAVE
 9988  075330  010446                    MOV     R4,-(SP)        ; REGISTERS
 9989  075332  010346                    MOV     R3,-(SP)        ;1
 9990  075334  010246                    MOV     R2,-(SP)        ; THRU
 9991  075336  010146                    MOV     R1,-(SP)        ;    5
 9992  075340  013746  076236            MOV     0.XXX,-(SP)     ;PUT SAVED REGISTER ON STACK
 9993  075344  005746                    TST     -(SP)
 9994  075346  000200                    RTS     R0
 9995
 9996                                  ; RESTORE REGISTERS R0-R6
 9997
 9998  075350  005726    0.RSR:  TST     (SP)+           ;POP THE EXTRA CELL
 9999  075352  012637  076236            MOV     (SP)+,0.XXX     ;GET R0 FROM STACK
10000  075356  012601                    MOV     (SP)+,R1        ;RESTORE
10001  075360  012602                    MOV     (SP)+,R2        ; REGISTERS
10002  075362  012603                    MOV     (SP)+,R3        ;  1
```

```
10003  075364  012604                        MOV     (SP)+,R4        ;  THRU
10004  075366  012605                        MOV     (SP)+,R5        ;    5
10005  075370  013706  072576                MOV     @.USP,SP        ;RESTORE USER STACK
10006  075374  013746  076236                MOV     @.XXX,-(SP)     ;PUT R0 ON USER STACK
10007  075400  000200                        RTS     R0
10008
10009                                ; RESTORE BREAKPOINTS 0-7
10010
10011  075402  012704  000016        @.RSB:  MOV     #@.BKP,R4       ;RESTORE ALL BREAKPOINTS
10012  075406  017464  072606  072652 1$:     MOV     @@.ADR1(R4),@.UIN(R4)   ;SAVE CONTENTS
10013  075414  013774  076346  072606        MOV     @.TRTC,@@.ADR1(R4)      ;REPLACE WITH TRAP
10014  075422  005304                        DEC     R4
10015  075424  005304                        DEC     R4
10016  075426  002367                        BGE     1$              ;RE-LOOP UNTIL DONE
10017  075430  000205                        RTS     R5              ;  THEN QUIT
10018
10019                                ; SAVE TELETYPE STATUS
10020
10021  075432  113737  177560  076246 @.SVTT: MOVB    @.RCSR,@.CSR1   ;SAVE R C/SR
10022  075440  113737  177564  076247        MOVB    @.TCSR,@.CSR2   ;SAVE T C/SR
10023  075446  105037  177560                CLRB    @.RCSR          ;CLEAR ENABLE AND MAINTENANCE
10024  075452  105037  177564                CLRB    @.TCSR          ;  BITS IN BOTH C/SR
10025  075456  105737  177564        1$:     TSTB    @.TCSR          ;LOOP UNTIL READY BIT COMES ON
10026  075462  100375                        BPL     1$              ;BR IF BIT NOT ON
10027  075464  000205                        RTS     R5
10028                                ; RESTORE TELETYPE STATUS
10029
10030  075466  004537  076200        @.RSTT: JSR     5,@.CRLF
10031  075472  105737  177564                TSTB    @.TCSR          ;WAIT READY
10032  075476  100375                        BPL     .-4             ;  ON PRINTER
10033  075500  032737  004000  177560        BIT     #4000,@.RCSR    ;CHECK BUSY FLAG
10034  075506  001403                        BEQ     1$              ;SKIP READY LOOP IF NOT BUSY
10035  075510  105737  177560                TSTB    @.RCSR          ;WAIT READY
10036  075514  100375                        BPL     .-4             ;    ON READER
10037  075516  113737  076246  177560 1$:     MOVB    @.CSR1,@.RCSR   ;RESTORE
10038  075524  113737  076247  177564        MOVB    @.CSR2,@.TCSR   ;  THE STATUS REGISTERS
10039  075532  000205                        RTS     R5
10040
10041                                ; REMOVE BREAKPOINTS 0-7
10042                                ;   IN THE OPPOSITE ORDER OF SETTING
10043
10044  075534  105737  076243        @.REM:  TSTB    @.S             ;SEE IF SINGLE INSTRUCTION IS GOING
10045  075540  001011                        BNE     2$              ;EXIT IF SO
10046  075542  005004                        CLR     R4              ;REMOVE ALL BREAKPOINTS
10047  075544  016474  072652  072606 1$:     MOV     @.UIN(R4),@@.ADR1(R4)   ;CLEAR BREAKPOINT
10048  075552  005204                        INC     R4
10049  075554  005204                        INC     R4
10050  075556  020427  000016                CMP     R4,#@.BKP
10051  075562  003770                        BLE     1$              ;RE-LOOP UNTIL DONE
10052  075564  000205                2$:     RTS     R5              ;THEN QUIT
10053
10054                                ; TYPE OUT CONTENTS OF WORD OR BYTE WITH ONE TRAILING SPACE
10055                                ; WORD IS IN R0
10056
10057  075566  012703  000006        @.CADV: MOV     #6,R3           ;# OF DIGITS
10058  075572  012704  177776                MOV     #-2,R4          ;# OF BITS FIRST-3
```

```
10059  075576  022737  000001  076240          CMP     #1,0.BW         ;SEE IF WORD MODE
10060  075604  001004                          BNE     3$              ;BRANCH IF SO
10061  075606  162703  000003                  SUB     #3,R3           ;ONLY DO 3 DIGITS
10062  075612  005204                          INC     R4              ;DO 2 BITS FIRST
10063  075614  000300                          SWAB    R0              ;AND TURN R0 AROUND
10064  075616  010046                  3$:     MOV     R0,-(SP)        ;SAVE R0
10065  075620  062704  000003          2$:     ADD     #3,R4           ;COMPUTE THE NUMBER OF BITS TO DO
10066  075624  005000                          CLR     R0
10067  075626  006116                  1$:     ROL     (SP)            ;GET A BIT
10068  075630  006100                          ROL     R0              ;STORE IT AWAY
10069  075632  005304                          DEC     R4              ;DECREMENT COUNTER
10070  075634  003374                          BGT     1$              ;LOOP IF MORE BITS NEEDED
10071  075636  062700  000060                  ADD     #'0,R0          ;CONVERT TO ASCII
10072  075642  004537  075662                  JSR     R5,0.FTYP       ;TYPE IT
10073  075646  005303                          DEC     R3              ;SEE IF MORE DIGITS TO DO
10074  075650  003363                          BGT     2$              ;LOOP IF SO
10075  075652  112700  000040                  MOVB    #' ,R0          ;SET UP FOR TRAILING SPACE
10076  075656  005726                          TST     (SP)+           ;GET RID OF JUNK
10077  075660  000400                          BR      0.FTYP
10078
10079
10080                                  ; TYPE ONLY ONE CHARACTER (CONTAINED IN R0)
10081
10082  075662  105737  177564          0.FTYP: TSTB    0.TCSR
10083  075666  100375                          BPL     .-4
10084  075670  042700  177400                  BIC     #177400,R0              ;CLEAR HIGH BYTE,SHOULD NOT
10085                                                                  ;CONTAIN INPORTANT INFO.
10086  075674  001416                          BEQ     3$
10087  075676  022700  000176                  CMP     #176,R0         ;PRINT ? FOR 177;16-37
10088  075702  103411                          BLO     2$              ; 1-10 AND 200-377.
10089  075704  022700  000037                  CMP     #37,R0
10090  075710  103410                          BLO     3$
10091  075712  022700  000015                  CMP     #15,R0
10092  075716  103403                          BLO     2$
10093  075720  022700  000010                  CMP     #10,R0
10094  075724  103402                          BLO     3$
10095  075726  012700  000077          2$:     MOV     #'?,R0
10096  075732  110037  177566          3$:     MOVB    R0,0.TDB
10097  075736  000205                  0.TYP1: RTS     R5
10098                                  ; GENERAL CHARACTER INPUT ROUTINE -- ODT11X
10099                                  ; CHARACTER INPUT GOES TO R0
10100
10101  075740  105737  177560          0.GET:  TSTB    0.RCSR          ;WAIT FOR
10102  075744  100375                          BPL     .-4             ;   INPUT FROM KBD
10103  075746  113700  177562                  MOVB    0.RDB,R0        ;GET CHARACTER - STRIP OFF PARITY
10104  075752  042700  177600                  BIC     #177600,R0      ;STRIP OFF PARITY FROM CHARACTER
10105  075756  122700  000003                  CMPB    #3,R0           ;IS IT ↑C?
10106  075762  001435                          BEQ     3$              ;IF SO, DO NOT ECHO
10107  075764  122700  000006                  CMPB    #6,R0           ;IS IT ↑F?
10108  075770  001432                          BEQ     3$              ;IF SO, DO NOT ECHO
10109  075772  122700  000005                  CMPB    #5,R0           ;IS IT ↑E?
10110  075776  001427                          BEQ     3$              ;IF SO, DO NOT ECHO
10111  076000  105737  076252                  TSTB    0.SCRN          ;SHOULD WE ECHO <LF>?
10112  076004  001003                          BNE     2$              ;BR IF YES
10113  076006  120027  000012                  CMPB    R0,#012         ;SEE IF A <LF>
10114  076012  001421                          BEQ     3$              ;IF SO SAVE THE PAPER
```

```
10115  076014  120027  000173          2$:     CMPB    R0,#173
10116  076020  002005                          BGE     1$
10117  076022  122700  000140                  CMPB    #140,R0         ;TEST FOR LOWER CASE
10118  076026  002002                          BGE     1$
10119  076030  162700  000040                  SUB     #40,R0          ;CHAR IS LC-CONVERT TO UPPER CASE
10120  076034  004537  075662          1$:     JSR     5,0.FTYP        ;ECHO CHARACTER
10121  076040  001737                          BEQ     0.GET           ;IGNORE NULLS
10122  076042  105737  076252                  TSTB    C.SCRN          ;SHOULD WE PASS ON SPACES?
10123  076046  001003                          BNE     3$              ;BR IF YES
10124  076050  122700  000040                  CMPB    #40,R0          ;CHECK FOR SPACES
10125  076054  001731                          BEQ     0.GET           ;IGNORE SPACES
10126  076056  000205                  3$:     RTS     R5
10127
10128                                  ; GENERAL CHARACTER OUTPUT ROUTINE - ODT11X
10129                                  ;  ADDRESS OF FIRST BYTE IN R4,
10130                                  ;  ADDRESS OF LAST BYTE IN R3, (R3)>(R4)
10131                                  ; EXPECTS LOCS 56,57 TO BE INITIALIZED BY MONITOR FOR FILL
10132                                  ; CHARACTERISTICS OF TERMINAL
10133                                  ;  56=CHAR TO BE FILLED AFTER
10134                                  ;  57=# OF NULLS TO FILL WITH
10135
10136  076060  020304                  0.TYPE: CMP     R3,R4           ;CHECK FOR COMPLETION
10137  076062  103725                          BLO     0.TYP1          ;  EXIT WHEN DONE
10138  076064  112400                          MOVB    (R4)+,R0        ;GET A CHARACTER
10139  076066  004537  075662                  JSR     5,0.FTYP        ;TYPE ONE CHARACTER
10140  076072  120037  002214                  CMPB    R0,@#$FILLC     ;COMPARE CHAR AGAINST FILL REQUIREMENT
10141  076076  001370                          BNE     0.TYPE          ;NO FILL NEEDED
10142  076100  113727  002027                  MOVB    @#$FILLS,(PC)+  ;FILL COUNT INTO TEMP
10143  076104  000000                  1$:     0
10144  076106  005000                          CLR     R0              ;FILL WITH NULLS
10145  076110  004537  075662          2$:     JSR     R5,0.FTYP       ;TYPE NULLS
10146  076114  005337  076104                  DEC     1$              ;DECREASE COUNT
10147  076120  003373                          BGT     2$              ;BRANCH IF NOT DONE
10148  076122  000756                          BR      0.TYPE          ;LOOP UNTIL DONE
10149
10150                                  ;SUBROUTINE TO READ THE PS INDEPENDENT OF MACHINE
10151
10152  076124  013737  177776  072602  0.RRST: MOV     ST,0.UST        ;STORE THE STATUS IN USER
10153                                                                  ;STATUS AREA.FOR AN LSI
10154                                                                  ;THIS IS CHANGED TO
10155                                                                  ;MFPS 0.UST
10156  076132  000207                          RTS     PC
10157
10158                                  ;SUBROUTINE TO WRITE PS INDEPENDENT OF MACHINE
10159                                  ;CALL ROUTINE WITH PS VALUE
10160                                  ;ON THE STACK
10161
10162  076134  016637  000002  177776  0.WST:  MOV     2(SP),ST        ;STORE NEW PS VALUE
10163                                                                  ;THIS INSTRUCTION IS CHANGED FOR LSI
10164                                                                  ;TO MTPS 2(SP)
10165  076142  012616                          MOV     (SP)+,(SP)      ;PUT RETURN PC OVER SUB. ARGUMENT
10166  076144  000207                          RTS     PC              ;TO RETURN WITHOUT IT ON THE STACK
10167                                  ; CLOSE WORD OR BYTE AND EXIT,
10168                                  ; UPON ENTERING, R2 HAS NUMERIC FLAG, R4 HAS CONTENTS
10169
10170                                          .ENABL  LSB
```

```
10171  076146  005702                  0.CLSE: TST     R2              ;IF NO NUMBER WAS TYPED THERE IS
10172  076150  001412                          BEQ     1$              ;NO CHANGE TO THE OPEN CELL
10173  076152  022737  000001  076240          CMP     #1,0.BW
10174  076160  001404                          BEQ     2$              ;JUMP IF BYTE MODE
10175  076162  101005                          BHI     1$              ;JUMP IF ALREADY CLOSED
10176  076164  010477  000042                  MOV     R4,@0.CAD       ;STORE WORD
10177  076170  000402                          BR      1$
10178  076172  110477  000034          2$:     MOVB    R4,@0.CAD       ;STORE BYTE
10179  076176  000207          1$:     RTS     PC
10180  076200  012703  076273          0.CRLF: MOV     #0.CR+1,R3      ;LWA <CR,LF>
10181  076204  000402                          BR      3$
10182  076206  012703  076274          0.CRLS: MOV     #0.CR+2,R3      ;LWA <CR,LF>*
10183  076212  012704  076272          3$:     MOV     #0.CR,R4        ;FWA
10184  076216  004537  076060                  JSR     5,0.TYPE        ;TYPE SOMETHING
10185  076222  000205                          RTS     R5
10186                                          .DSABL  LSB
10187                                  ;SUBROUTINE 0.RORA
10188                                  ;FUNCTION:                              THE
10189                                  ;ADDRESS WILL BE PRINTED IN ABSOLUTE FORM
10190                                  ;INPUT: THE ADDRESS TO BE PRINTED IS IN R0.
10191                                  ;DATA SAVED:  R0 CONTAINING THE ADDRESS TO BE
10192                                  ;PRINTED, AND LOCATION 0.CAD CONTAINING
10193                                  ;THE CURRENT ADDRESS WERE SAVED AND RESTORED.
10194                                  ;CALLED:  JSR 5,0.RORA
10195
10196  076224  004537  075566          0.RORA: JSR     5,0.CADV        ;PRINT ABSOLUTE ADDRESS
10197  076230  000205                          RTS     R5
10198  076232  000000          0.CAD:  0                               ; CURRENT ADDRESS
10199  076234  000000          0.DOT:  0                               ; ORIGIN ADDRESS
10200  076236  000000          0.XXX:  .WORD   0                       ;TEMPORARY STORAGE
10201  076240  000000          0.BW:   .WORD   0                       ; =0 - ALL CLOSED
10202                                                                  ; =1 - BYTE OPEN,
10203                                                                  ; =2 - WORD OPEN
10204  076242      000         0.SEQ:  .BYTE   0                       ;CHANGE SEQUENCE INDICATOR
10205  076243      000         0.S:    .BYTE   0                       ;SINGLE INSTRUCTION FLAG
10206                                                                  ;0 IF NOT ACTIVE
10207                                                                  ;-1 IF ACTIVE
10208                                                                  ;NO BREAK BOINTS MAY BE SET WHILE IN
10209                                                                  ;SINGLE INSTRUCTION MODE
10210  076244      000         0.T:    .BYTE   0                       ;  T-BIT FLAG
10211  076245      000         0.P:    .BYTE   0                       ;PROCEED FLAG = -2 IF MANUAL ENTRY
10212                                                                  ;               -1 IF NO PROCEED ALLOWED
10213                                                                  ;               0-7 IF PCEED ALLOWED
10214  076246      000         0.CSR1: .BYTE   0                       ;SAVE CELL - R C/SR
10215  076247      000         0.CSR2: .BYTE   0                       ;SAVE CELL - T C/SR
10216  076250      000         0.CMFD: .BYTE   0       ;COMMA FOUND SWITCH, =0 NO COMMA FOUND
10217                                                                  ;               =1 COMMA FOUND
10218  076251      000         0.SMFD: .BYTE   0                       ;SEMICOLON FOUND SWITCH
10219                                                                  ;=0 NO SEMICOLON FOUND
10220                                                                  ;=1 SEMICOLON FOUND
10221  076252      000         0.SCRN: .BYTE   0                       ;FLAG; 1=PASS SPACES ON FROM TTY
10222                                                                  ;ALSO, IF =1, <LF> IS ECHOED
10223  076253      012         0.ID:   .BYTE   012
10224  076254      015                 .BYTE   015
10225  076255      040                 .BYTE   40
10226  076256  042117  020124  041450          .ASCII /ODT (CEMKA)/
```

```
        076264  046505  040513      051
10227           076270                  O.IDND=.-1
10228   076271      000                 O.MINS: .BYTE   0       ;MINUS SIGN TYPED  (SWITCH)
10229                                                           ;0=NO MINUS TYPED;  1=MINUS SIGN TYPED
10230   076272      015                 O.CR:   .BYTE   015     ;  <CR>
10231   076273      012                         .BYTE   012     ;  <LF>
10232   076274      052                         .BYTE   '*      ;  *
10233
10234   076275      073                 O.LGCH: .BYTE   ';      ;
10235   076276      057                         .BYTE   '/      ;  /
10236   076277      134                         .BYTE   '\      ;  \      (BACK SLASH)
10237   076300      015                         .BYTE   015     ;  CARRIAGE RETURN
10238   076301      044                         .BYTE   '$      ;  $
10239   076302      107                         .BYTE   'G      ;  G
10240   076303      012                         .BYTE   012     ;  <LF>
10241   076304      137                         .BYTE   '_      ;  _      (BACK ARROW)
10242   076305      074                         .BYTE   '<      ;  <
10243   076306      136                         .BYTE   '^      ;  ^      (UP ARROW)
10244   076307      117                         .BYTE   'O      ;  O
10245   076310      102                         .BYTE   'B      ;  B
10246   076311      120                         .BYTE   'P      ;  P
10247   076312      100                         .BYTE   '@      ;  @
10248   076313      076                         .BYTE   '>      ;  >
10249   076314      123                         .BYTE   'S      ;  S
10250   076315      055                         .BYTE   '-      ;  -
10251   076316      003                         .BYTE   003     ;CTRL C
10252   076317      006                         .BYTE   006     ;CTRL F
10253   076320      005                         .BYTE   005     ;CTRL E
10254           000024                  O.CLGT  =       .-O.LGCH                ;TABLE LENGTH
10255
10256   076321      123                 O.TL:   .BYTE   'S      ;DO        1
10257   076322      120                         .BYTE   'P      ;NOT       2
10258   076323      115                         .BYTE   'M      ;CHANGE    3
10259   076324      000                         .BYTE   0       ;THE       4
10260   076325      000                         .BYTE   0       ;ORDER     5
10261   076326      103                         .BYTE   'C      ;          6
10262   076327      106                         .BYTE   'F      ;          7
10263   076330      122                         .BYTE   'R      ;         10
10264   076331      000                         .BYTE   0       ;         11
10265   076332      000                         .BYTE   0       ;         12
10266   076333      000                         .BYTE   0       ;         13
10267   076334      000                         .BYTE   0       ;         14
10268   076335      000                         .BYTE   0       ;         15
10269   076336      000                         .BYTE   0       ;         16
10270   076337      000                         .BYTE   0       ;         17
10271   076340      102                         .BYTE   'B      ;         20
10272           000020                  O.LG    =       .-O.TL
10273           076342                          .EVEN
10274   076342  042502                  O.BD:   .WORD   "BE
10275   076344  000000                  O.BIAS: .WORD   0       ;CURRENT RELOCATION BIAS
10276   076346  000003                  O.TRTC: TRT                     ;TRACE TRAP PROTOTYPE
10277                                   ;THE ORDER OF THE FOLLOWING ENTRIES IS CRITICAL
10278
10279           072562                          .=O.ODT-144     ;ODT'S STACK IMMEDIATELY PRECEDES ODT
10280
10281   072562  000000                  O.URO:  0       ;USER R0
```

```
10282  072564  000000                          0       ;       R1
10283  072566  000000                          0       ;       R2
10284  072570  000000                          0       ;       R3
10285  072572  000000                          0       ;       R4
10286  072574  000000                          0       ;       R5
10287  072576  000000                  0.USP:  0       ;USER SP
10288  072600  000000                  0.UPC:  0       ;USER PC
10289  072602  000000                  0.UST:  0       ;USER ST
10290  072604  000007                  0.PRI:  7       ;ODT PRIORITY
10291                                  ; BREAK POINT LISTS, ADR1 = ADDRESS OF BREAKPOINT,CT = COUNT,
10292                                  ;   UIN = CONTENTS
10293
10294  072606                          0.ADR1:
10295          072630                          .=.+0.BKP+4
10296  072630                          0.CT:
10297          072652                          .=.+0.BKP+4
10298  072652                          0.UIN:
10299          072674                          .=.+0.BKP+4
10300          076350                          .=0.TRTC+2
10301  076350                          $$END
10302  076350     001                  END:
10303  076350  001430                          .PRINT  100000-END      ;ADDRESSES LEFT
10307          003614                          .END START
```

```
ABORTF  001524  1031# 5178* 7307 7321 7326 7336 7341 7393 7467*
ACFLAG  001506  1024# 1723 1752 1786* 1787 2030 2120 2153 2186 2226 2271 2312 2360
                2401 2475 3274 3292 3323 5644 5835* 5850* 6410 6462 6604 6678 6743
                6786 6821
ACTFLA  002020  1102# 1371* 1546 1709 2857 3058 3067 3075 3085 3095 3104 3117 3156
                3162 3169 3175 3233 6014 7601
ADDPAR  011462  2027 2043#
ADDRES  001422  998# 1540* 1799* 3769* 3783* 3841* 3855* 3927* 5209* 5224* 5225* 5226 5228
                7202* 7203* 7208* 7209* 7214* 7215* 7220* 7221* 7228* 7236* 7241* 7246* 7251*
                7256* 7261* 7266* 7271* 7276* 7281* 7286* 7291* 7296* 7301* 7306* 7370* 7392*
                7395 7416* 7448* 7463 7767 7898 7900 7902 7905 7907 7909 9312 9315
                9318 9320 9325
ADD..0  052744  7872 7896#
ADD..1  052754  7873 7898#
ADD..2  052774  7874 7900#
ADD..3  053024  7875 7902#
ADD..4  053054  7876 7904#
ADD..5  053056  7877 7905#
ADD..6  053076  7878 7907#
ADD..7  053126  7879 7909#
ALLCPU  002074  1128# 6692* 6694* 6695* 6740 6783 6818
APTCOR  002104  1133# 1986* 1994 9317
APTDOW  037236  1364 6042# 6045 6047*
APTFLA  002022  1103# 1360* 1965 3058 3067 3075 3085 3095 3104 3117 3233 6014 7601
APTHAN  011720  1926 2103# 2109
APTHLT  037304  6049#
APTMOS  002106  1134# 1989* 1994 9317
A3    = 000002  7047#
BACKGN  031124  3013* 3024* 3041* 3191* 4935# 5978*
BAD     001440  1004# 1801* 5228* 7204* 7210* 7217* 7222* 7227* 7237* 7242* 7247* 7252* 7257*
                7262* 7267* 7272* 7277* 7282* 7287* 7292* 7297* 7302* 7309* 7373* 7386 7395*
                7417* 7449* 7450* 7466* 9312 9318 9320 9328
BADPC   001420  997# 5179* 5193* 5217* 5221* 5226* 7230* 7322* 7337* 7577 7578 7579*
BADXOR  001446  1007# 7386* 7387* 9318
BAD2    001442  1005# 7374* 9328
BAD3    001444  1006# 7376* 7377* 9328
BAFPAF  011754  2005 2116# 2145*
BAFPAR  012070  2006 2149# 2178*
BAKPAT  002174  1166# 1279* 3190 4611 4645* 4683 4690 4699 4723* 7298 7303
BALANC= 000077  6799# 6808
BANK    001470  1017# 1553* 1560* 1561 1563 1569 1584* 1651* 1652 1657 1705* 1707* 1721*
                1728* 1729* 1732 1750* 1758* 1785* 1788* 1791 2028* 2037* 2117* 2150* 2183*
                2223* 2266* 2307* 2355* 2396* 2463 2473* 2476 2498* 2512* 3272* 3275 3287*
                3290* 3293 3306* 3320* 3329* 3339* 3353 3371 3404 5642* 5645 5653* 5705
                5706* 5710* 5839 5907* 5908 6169 6174* 6178* 6181 6187 6202* 6207 6212*
                6216* 6219 6225 6251* 6256 6262* 6263 6265 6293 6368* 6369 6380 6408*
                6411 6418* 6420* 6425 6460* 6465* 6472* 6473 6478 6488* 6492* 6495 6514
                6539* 6540 6545 6550* 6551 6555 6561* 6563* 6567 6602* 6607* 6614* 6615
                6676* 6688* 6701* 6705* 6708* 6739* 6777* 6782* 6813* 6817* 6849* 6909 6967*
                6980* 6998* 7040 7403 7525 7716 7740 7769 7866 8128 8129* 8130 8148*
                8164 8165* 8166 8179* 8474
BANKIN  001472  1018# 1468* 1469 1486* 1565* 1613 1654* 1696 1733 5649 5708 5841* 6684
                6703 6710
BANKMO  035570  1739* 2101* 5648* 5704* 5741# 6037*
BANKOK  036334  2269* 2310* 2358* 2399* 5882#
BAWPAF  012204  2007 2182# 2210* 2218*
```

```
BAWPAR  012342          2008    2222#   2250*   2258*
BDSWIT  002062          1122#   1808*   9320
BEGIN   004676          1249    1281    1282    1383    1385#
BITC  = 000001           249#   1446    1815    1816    5167    5240    5244    5249    5254    5345    5347    5373    5378
                        5383    5389    5430    5432    5438    5441    5463    5465    5480    5482    5693    5695    5717
                        5719    6183    6221    6497    7036    7464    7470    7809
BIT1  = 000002           248#   1816    1820    2043    2567    2629    5361    5367    5383    5389    5396    5403    5496
                        5503
BIT10 = 002000           239#
BIT11 = 004000           238#   1620    1638    5011    5276    5278    5299    5301    5658    5871    6724    7405    7465
                        7469
BIT12 = 010000           237#   1401    1402    1407    1473    1697    2641    5138    5663    5842    7949    7950    7965
BIT13 = 020000           236#   1401    1407    1471    1473    1656    1934    1935    2642    5116    5661    5662    5735
                        6007    7949    7965
BIT14 = 040000           235#   1472    1579    1592    1609    1630    1656    1659    1934    1935    2540    2834    3354
                        3393    3426    4571    4572    5352    5354    5661    5662    5681    5714    5765    5781    6044
                        6188    6226    6243    6528    6876    6884    6901    6908    6917    6933    6966    6979    6991
                        6997    7017    7046    7057    7308    7372    7394    8014    8018    8096    8100    8131    8167
BIT15 = 100000           234#   1380    1381    1719    1734    1843    1852    2559    2571    2574    2578    3146    5354
                        5452    5650    5658    5659    5681    5688    5709    5714    5765    5781    5862    6685    6704
                        6711    6717    6902    7538    7571    7772    7805    8014    8100
BIT2  = 000004           247#   1815    1817    1850    2629    5254    5396    5403    5496    5503    7898    7900    7902
                        7905    7907    7909
BIT3  = 000010           246#   1475    1818    1819    2628    2634    5112    5254    5362    5368    5384    5390    5397
                        5404    5497    5504    7900    7902    7907    7909
BIT4  = 000020           245#    964    1336    1492    1493    1718    1817    1819    1867    1878    2097    2559    2578
                        5094    5344    5419    5439    5847    6671    6673    6691    6692    6695    6742    6785    6820
                        6862    7805
BIT5  = 000040           244#   1353    1426    1820    1821    1829    2097    5094    5684    5692    5721    5725    5998
                        6039
BIT6  = 000100           243#   1821    1822    1833    4954    4989    5094    6305    6363    6398    6404    6451    6470
                        6522    6537    6593    6612    7007
BIT7  = 000200           242#   1356    1818    1822    1837    2636    4953    5094    6304    6397    6450    6521    6592
                        6632
BIT8  = 000400           241#   1446    5254
BIT9  = 001000           240#   2851    2852    5240    5244    5254    5693    5695    5717    5719    7036    7809
BLOCK1  037306          1557    2027    2962    2971    2980    2999    3016    3027    3149    3154    3160    3167    3173
                        3179    3197    3200    3204    3207    3250    3255    3267    3317    4420    4940    5330    6058#
                        6963    7001
BLOCK2  037326          6063#
BLOCK3  037342          2995    3000    3001    3017    3028    3251    3252    3355    5682    5715    5766    5782    6067#
                        8134    8137
BMFLAG  001520          1029#   1723    2188    2228    2314    2403    5644    5838*   5873*   5877*
BOOT    037020          5984#   8431    8533    9556
BOOT1   037062          5993    5999#
BO      004304          1319#   1325
B1      006532          1651#   1705
B10     010340          1883#
B100    045004          6800#   6810
B101    045076          6817#   6849
B102    045140          6823#   6828
B103    045174          6830#   6835
B104    045216          6836#
B105    045220          6837#   6846
B106    045276          6853#   6859
B107    045372          6871#   6880
```

```
B11     010350          1886#   1897
B110    046554          7042#   7056
B111    046604          7047#   7054
B112    052220          7800#
B113    052224          7801#   7814
B114    052426          7838#   7846
B115    052536          7861#
B116    052540          7862#   7886
B117    053332          7942#   7947
B12     010352          1887#   1896
B120    053416          7955#   7960
B121    053502          7970#   7975
B122    053744          8024#
B123    053750          8025#   8032
B124    054236          8076#
B125    054244          8077#   8084
B13     010452          1905#   1911
B14     010546          1919#   1924
B15     010614          1932#   1939
B16     011112          1968#   1993
B17     011404          2028#   2037
B2      007074          1720#
B20     013462          2466#   2510
B21     013504          2469#   2505
B22     013510          2471#
B23     013522          2473#   2498
B24     013602          2480#   2495
B25     013622          2485#   2490
B26     017632          3124#   3138
B27     020716          3271#   3288
B3      007102          1721#   1728
B30     020722          3272#   3287
B31     021050          3289#   3307
B32     021056          3290#   3306
B33     021260          3320#   3329
B34     027102          4501#   4564
B35     027140          4509#   4563
B36     027154          4515#   4527
B37     027266          4536#   4560
B4      007260          1749#
B40     027446          4568#   4570
B41     027470          4572#   4575
B42     033642          5414#
B43     033646          5415#   5425
B44     034006          5447#
B45     034012          5448#   5458
B46     034310          5512#
B47     034314          5513#   5519
B5      007264          1750#   1758
B50     034734          5641#
B51     034742          5642#   5653
B52     035044          5656#
B53     035050          5657#   5671
B54     036160          5848#   5853
B55     041670          6408#   6418
B56     042232          6460#   6465
```

```
B57     043074          6550#   6561
B6      007516          1787#   1790
B60     043432          6602#   6607
B61     043716          6660#   6669
B62     043770          6670#   6690
B63     044006          6675#
B64     044014          6676#   6688
B65     044134          6693#   6696
B66     044204          6701#   6705
B67     044366          6736#   6778
B7      010254          1866#   1877
B70     044400          6739#   6777
B71     044436          6744#   6775
B72     044516          6757#   6761
B73     044600          6767#   6774
B74     044672          6782#   6813
B75     044730          6787#   6791
B76     044760          6793#   6798
B77     045002          6799#
CACHK   002132          1147#   5248
CACHOF= 104424           105#   1530    1578    1591    1608    1629    1647    1762    1827    1882    2483    2541    2835
                        3395    3428    4535    5287    5321    5339    6095    6656    6868    6975    6988    7012
CACHON= 104423           104#   1533    1582    1595    1612    1634    1715    1784    1879    1899    2491    2582    2590
                        2849    3397    3430    4512    7990
CACHVE= 000114           264#    265
CBCSR = 104474           148#
CB1CSR= 104475           149#   2833    3743    3828    3919    3992    4068    4144    4795
CHECK   001762          1086#   2831*   5395    5402    5548*   9325
CHKDIS= 104504           156#   1660    1669    1677    1685
CHKGEN  034350          3732*   3819*   3908*   3980*   4056*   4132*   4738*   5525#
CHKTAB  034450          5535    5552#
CHK1DI= 104505           157#   4307    4328    4336    4350    4371    4380    4395    4414    7364
CKEND   056444          8409    8411    8413    8448#
CKSWR = 104410            87#   7489    7561    7593
CLRCSR= 104502           154#   1663    1672    1680    1688    2860    3266    3332    6319    6384    6442    6484    6584
                        6962
CLR1CS= 104503           155#   2525    2550    2569    2585    2838    2847    3148    3181    3715    3760    3773    3804
                        3811    3830    3882    3888    3902    3921    3949    3974    3996    4031    4050    4070    4106
                        4148    4184    4425    4772    7369
CMD11A  043236          6570#   6574    6578
CMD11B  043426          6602#   6609
CMD11C  043500          6589    6611#
CMD5A   040762          6282#   6286    6290
CMD5B   041170          6314#   6327
CMD5C   041354          6297    6301    6362#
CMD7A   041632          6394    6403#
CMD8A   042036          6428#   6432    6436
CMD8B   042226          6460#   6467
CMD8C   042300          6447    6469#
CMD9A   042524          6500#   6504    6508
CMD9B   042740          6513    6533#
CMD9C   042746          6518    6536#
CMD9LO  042726          6512    6529#   6531    6533
CONFGL  003550          1232#
CONFGT  005472          1437    1491#
CONFG1  005154          1439#   1499
```

```
CONFG3  005342  1469#  1489
CONFG4  005446  1476  1486#
CONFG5  005516  1490  1496#
CONFIE  003550  1227#  1332
CONFIG  002230  1226#  1330  1380*  1381*  1470*  1471*  1472*  1475*  1481*  1485*  1542*  1570  1620*
                1621*  1638*  1641*  1655  1656  1697*  1719*  1734*  1933  1934  1935  5011  5029
                5049  5056  5088  5094  5111  5134  5138  5650*  5658  5659  5660  5661  5662
                5663  5709*  5842  5849  5856  5862  5871  6267  6554  6556  6685*  6704*  6711*
                7405*  7407*  7409*  7410
CONFSP  005560  1438*  1491  1503#
CONTFL  001704  1060#  2000*  2452  2511*
CONTP   056216  8407#  9527*  9574
CONTRL= 177746  272#  1436  1501*  5248*  5249*  5254*  5286  5306*  5320  5324*  5338  5341*  6094
                6122*  7465*  7469*  7989  8120*  9326
CONTT   056532  8402*  8468#
COREHI  036704  5920*  5934*  5947*  5963*  5969#  6299*  6367*
COUNT   002040  1113#  4513*  4516*  4517*  4518  4525*  4533*  4537*  4538*  4539  4558*
CPUBIT  001476  1020#  1336*  1542  1621  1641  1655  1718  1829  1833  1837  1869*  1878*  1933
                5660  5849  6267  6674*  6691*  6740*  6742*  6783*  6785*  6818*  6820*  6862
CPUERR= 177766  282#  1256*  1348*  1397*  1492  1500*  1543*  1585*  1777*  6133*  6137*  6148*  6165*
                6203*  6252*  7614*  9316  9321  9326
CR    = 000015  181#  8241
CRLF  = 000200  182#  8221  9269  9351  9352  9353  9354  9356  9358  9359  9360  9364  9366
                9367  9373  9377  9378  9379  9380  9381  9382  9383  9384  9385  9386  9387
                9388  9389  9390  9391  9393  9394  9395  9396  9397  9398  9399  9400  9401
                9402  9403  9404  9405  9409  9411  9412  9413  9414  9415  9416  9417  9418
                9419  9420  9421  9422  9424  9425  9426  9427  9428  9430  9431  9432  9435
                9436  9440  9442  9446  9447  9448  9449  9452  9453  9454  9455  9456  9458
                9459  9461  9463  9465  9466  9467  9471  9472  9473  9479  9480
CSR     001526  1032#  1443  1462  1463  1473  1658*  1668*  1676*  1684*  1917*  1918*  2527  2559
                2578  2601  2628*  2629*  2634*  2635  2639  2841  4306*  4320  4322  4327*  4335*
                4341  4349*  4363  4365  4370*  4379*  4386  4394*  4405  4407  4413*  5265  5276*
                5278*  5282  5288  5299*  5301*  5305  5316*  5317*  5322*  5323*  5344  5355*  5356*
                5361*  5362*  5363*  5367*  5368*  5369*  5373*  5374*  5378*  5379*  5383*  5384*  5385*
                5389*  5390*  5391*  5395*  5396*  5397*  5398*  5402*  5403*  5404*  5405*  5419  5439
                5452  5471  5473  5477  5485*  5486*  5490*  5491*  5496*  5497*  5498*  5503*  5504*
                5505*  6151*  6154*  7366  7797  7805  7806  7809  7816*  8029  8080*  9323  9324
                9325
CSRCAS  014666  2489*  2654#
CSRFB   001572  1035#  1464*  7864
CSRFBA  001766  1088#  2498  2597*  2710*  2726*  2742*  2758*  2786*  2802*  2818*
CSRFIR  001710  1067#  2480  2704*  2712*  2714*  2720*  2728*  2730*  2736*  2744*  2746*  2752*  2760*
                2762*  2780*  2788*  2790*  2796*  2804*  2806*  2812*  2820*  2822*  2832  3142  3166
                4301  6256  6315*  6366*
CSRINC  001772  1090#  2495  2706*  2716*  2722*  2732*  2738*  2748*  2754*  2764*  2782*  2792*  2798*
                2808*  2814*  2824*  3131*  3144  4302
CSRIND  001774  1091#  2464*  2467  2508*
CSRLB   001612  1036#  1465*  7865
CSRLBA  001770  1089#  2473  2598*
CSRLOO  001776  1093#  2466*  2501  2510*  2529
CSRNO   001634  1038#  1435*  1460  1477  1436*  1497  1919*  1924*  2524*  3125*  5262  5269  5292
                5313  5332  5415*  5425*  5448*  5458*  5513*  5519*  6094  6122*  6380  6420*  6634*
                6646*  7797  7801*  7814*  7816*  7857  7893*  7993  8025*  8032*  8077*  8084*  8116*
                9321
CSROUD  001775  1092#  2465*  2509*  2519
CSROUT  034302  5364*  5375*  5386*  5399*  5487*  5499*  5509#
```

```
CSRREL  001632  1037#  1391*  1396*  5263  5314  5333
CSRSTU= 000022  2471#  2493
CSR0    001532  1033#  1462*
CSR2    001552  1034#  1463*  7868
DATBUF  001712  1068#  3583*  3584*  3585  3586  3588  3593  3597  3599*  3602*  3603*  3604  3605
                3607  3612  3616  3618*  3717*  3718*  3723  3724  3792  3794*  3812*  3813*  3816
                3817  3874  3876*  3890*  3891*  3896  3897  3962*  3963*  3968  3969  4038*  4039*
                4044  4045  4115*  4116*  4121  4122  7238  7243
DBEMSK  001726  1071#  3894*  3895*  3903  3905  3913  3915  3931  3933*  3966*  3967*  3975  3977
                3985  3987  4013  4015*  4042*  4043*  4051  4053  4061  4063  4088  4090*  4119*
                4120*  4127  4129  4137  4139  4166  4168*
DDISP = 177570   175#  1171  1342
DECMAR  036556  2163*  2238*  2368*  2411*  5939#
DEENER= 104421   100#  1399  5997  6038  6730
DETAIL  053156  7692*  7912#  7938*
DETFLA  001702  1059#  7600  7691  7913*  7914  7916  7979*
DETR0   001664  1052#  7921*  7929  9322
DETR1   001666  1053#  7922  9322
DETR2   001670  1054#  9322
DETR3   001672  1055#  9322
DETR4   001674  1056#  9322
DETR5   001676  1057#  9322
DETR6   001700  1058#  9322
DF10    066075  9114  9345#
DF11    066102  9089  9094  9140  9346#
DF2     065742  9023  9028  9043  9048  9053  9058  9074  9099  9104  9120  9125  9130  9135
                9145  9150  9155  9170  9176  9186  9226  9232  9337#
DF3     065766  9064  9165  9191  9196  9206  9211  9216  9338#
DF4     066003  9069  9339#
DF5     066020  9079  9109  9201  9340#
DF6     066035  9084  9341#
DF7     066052  9160  9342#
DF8     066066  9181  9343#
DF9     066070  9033  9038  9344#
DH1     064046  9026  9056  9284#
DH10    064602  9072  9295#
DH11    064667  9102  9133  9153  9296#
DH12    064716  9128  9297#
DH13    064733  9087  9092  9097  9118  9123  9138  9298#
DH14    065041  9143  9300#
DH15    065157  9158  9302#
DH16    065301  9163  9179  9304#
DH17    065316  9168  9184  9230  9306#
DH2     064103  9021  9285#
DH3     064103  9031  9036  9286#
DH4     064137  9287#
DH5     C64144  9041  9046  9051  9148  9288#
DH6     064252  9224  9290#
DH7     064321  9062  9067  9077  9082  9107  9189  9194  9199  9204  9209  9214  9291#
DH8     064443  9112  9293#
DH9     064543  9174  9294#
DISPLA  002202  1171#  1342*  1350*  6530*  7539*  7564*  8114*
DISPRE= 000174   482#  1350  7540*  7566*  8115*
DISPTB  011302  2003*  2005#
DSWR  = 177570   174#  1170  1341
DT1     065366  9027  9057  9312#
```

```
DT10    065542          9073    9322#
DT11    065562          9103    9134    9154    9323#
DT12    065572          9129    9324#
DT13    065600          9088    9093    9098    9119    9124    9139    9325#
DT14    065624          9144    9326#
DT16    065652          9159    9327#
DT17    065704          9164    9180    9329#
DT18    065712          9169    9231    9331#
DT19    065726          9185    9332#
DT2     065400          9022    9313#
DT3     065400          9032    9314#
DT4     065410          9037    9315#
DT5     065420          9042    9047    9052    9149    9316#
DT6     065444          9225    9317#
DT7     065460          9063    9068    9078    9083    9108    9190    9195    9200    9205    9210    9215    9318#
DT8     065514          9113    9320#
DT9     065530          9175    9321#
DUMMY   001662          1050#   9318    9319    9325    9327    9328    9329
ECCDIS= 104470          144#    1699    2858    3318    6317    6382    6440    6482    6582
ECCINI= 104472          146#    1521    1549    1712    2867    3335    3342    3346    5992    6029
ECC1DI= 104471          145#    2542    2561    2844    3741    3748    3800    3826    3878    3917    3945    3990    4027
                        4066    4102    4142    4180    4423    4777    4784    4793
ECC1IN= 104473          147#    2581    2589    4743    4786
EMTVEC= 000030          259#    1286*   1287*
EM1     061326          9020    9237#
EM10    061741          9055    9247#
EM11    062005          9061    9066    9076    9081    9248#
EM12    062027          9071    9249#
EM13    062053          9086    9250#
EM14    062105          9091    9251#
EM15    062151          9096    9252#
EM16    062217          9101    9253#
EM17    062245          9106    9254#
EM18    062305          9111    9255#
EM19    062364          9117    9256#
EM2     061326          9025    9238#
EM20    062441          9122    9257#
EM21    062523          9132    9258#
EM22    062562          9137    9259#
EM23    062607          9147    9260#
EM24    062636          9152    9261#
EM25    062636          9157    9262#
EM26    062663          9162    9263#
EM27    062734          9178    9264#
EM28    063023          9183    9266#
EM29    063056          9193    9268#
EM3     061364          9030    9239#
EM30    063140          9198    9269#
EM31    063257          9203    9271#
EM32    063357          9208    9272#
EM33    063464          9213    9273#
EM34    063572          9218    9274#
EM35    063645          9188    9275#
EM36    063732          9223    9276#
EM37    064001          9229    9278#
EM4     061417          9035    9240#
```

```
EM5     061465          9040    9241#
EM6     061542          9045    9242#
EM7     061567          9050    9243#
EM8     061627          9167    9244#
EM9     061664          9173    9246#
ENASBE= 104506           158#   1547    1710
ENA1SB= 104507           159#   4746    4779    4788
END     076350          1267   10302#  10303
ENERGI= 104420            99#   1432    2098    6732
ERRMAX  002134          1148#   7410
ERROR = 104000           169#   1494    1537    1541    1754    1809    1890    1995    2034    2050    3770    3784    3842
                        3856    3928    4751    4806    4817    5194    5218    5222    5229    5475    5479    6132    6147
                        6161    6807    6843    6882    7327    7329    7342    7344    7351    7379    7429    7432    7435
                        7438    7444    7454    7930    7935
ERRORP  001416           996#   7573*   7574*   7575    7578*   7581    7632    9312    9314    9315    9316    9317    9318
                        9321    9323    9325    9327    9329    9331    9332
ERRVEC= 000004           252#   1296*   1297*   7501    7502*   7504*   7507*
EVEN    002034          1111#   4501*   4502    4564*
EXBANK  036052          1722*   1730*   1751*   1789*   2029*   2119*   2152*   2185*   2225*   2268*   2309*   2357*   2398*
                        2474*   2513*   3273*   3291*   3321*   3340*   5643*   5707*   5819#   6284*   6370*   6409*   6461*
                        6474*   6541*   6603*   6616*   6677*   6702*   6709*   6741*   6784*   6819*
EXCMD3  040234          6192    6195    6199    6201#
EXCMD4  040542          6232    6236    6250#
EXIT    037124          6012#   7492    7603
EXIT2   037130          6014#
E0      004334          1325#
E1      007010          1705#
E10     010422          1892    1898#
E100    045052          6810#
E101    045270          6849#
E102    045166          6828#
E103    045216          6835#
E104    045254          6844    6847#
E105    045254          6846#
E106    045326          6859#
E107    045446          6880#
E11     010422          1897#
E110    046630          7056#
E111    046624          7054#
E112    052316          7810    7813    7815#
E113    052316          7814#
E114    052464          7846#
E115    052702          7881    7887#
E116    052702          7886#
E117    053364          7947#
E12     010412          1896#
E120    053450          7960#
E121    053534          7975#
E122    054010          8031    8033#
E123    054010          8032#
E124    054304          8083    8085#
E125    054304          8084#
E13     010506          1911#
E14     010572          1924#
E15     010676          1939#
E16     011232          1993#
```

```
E17     011454          2037#
E2      007166          1725    1731#
E20     014014          2510#
E21     013766          2505#
E22     013724          2493    2499#
E23     013724          2498#
E24     013710          2495#
E25     013660          2490#
E26     017716          3138#
E27     021042          3288#
E3      007154          1728#
E30     021026          3287#
E31     021200          3307#
E32     021164          3306#
E33     021336          3329#
E34     027440          4564#
E35     027424          4563#
E36     027246          4527#
E37     027410          4560#
E4      007332          1756    1759#
E40     027464          4570#
E41     027510          4575#
E42     033714          5421    5424    5426#
E43     033714          5425#
E44     034060          5454    5457    5459#
E45     034060          5458#
E46     034344          5518    5520#
E47     034344          5519#
E5      007332          1758#
E50     035044          5651    5655#
E51     035040          5653#
E52     035204          5661    5665    5676#
E53     035164          5671#
E54     036212          5853#
E55     041750          6418#
E56     042272          6465#
E57     043162          6561#
E6      007544          1790#
E60     043472          6607#
E61     043764          6669#
E62     044114          6690#
E63     044104          6686    6689#
E64     044104          6688#
E65     044156          6696#
E66     044236          6705#
E67     044662          6778#
E7      010326          1877#
E70     044652          6777#
E71     044636          6775#
E72     044542          6761#
E73     044626          6774#
E74     045066          6813#
E75     044752          6791#
E76     045002          6798#
E77     045052          6808    6811#
FASTCI= 177640           312#   1580*   1610*   2997    3396*   4424*   5335*   5340*   6059
```

```
FATAL$  001454  1011#  1537*  1541*  4751*  5194*  5218*  5222*  5229*  6882*  7601  7645
FCMD10  043032  6114  6544#
FCMD11  043176  6115  6566#
FCMD12  043570  6116  6620#
FIELDS  037372  6080#  8399*  9565*
FINDBA=  000052  5656#  5661  5665
FINDTU=  000112  7800#  7810  7813
FIRST =  060000  489#  1573  1602  1627  1649  2025  2704  2712  2714  2720  2728  2730  2736
                 2744  2746  2752  2760  2762  2780  2788  2790  2796  2804  2806  2812  2820
                 2822  2959  2967  2988  3014  3025  3042  3196  3219  3227  3247  3276  3295
                 3325  3355  3360  3361  4195  4421  4514  4534  4568  4570  4571  4938  6045*
                 6046*  6047*  6182  6220  6315  6412  6496  6877  6885  6887  6902*  6909  6918
                 6934  6967  6970  6980  6984  6992*  6998  7009  7018*  7041  7768  8132  8133
                 8134  8135*  8136*  8137  8139  8168*  8169*  8174
FIRSTB  001636  1039#  1464  1466  1487*  1488  2597  2645*  2646  2648*  2710  2726  2742  2758
                 2786  2802  2818  7864*  7866
FIRSTP=  000002  1720#  1725
FLIPLO  002160  1158#  1273*  3004  4909*  4910*  4911  4913  4915
FLIPWA  030774  2987*  4907#
FLUSH  011526  2056#  6713
FSCMD0  037544  6104  6120#
FSCMD1  037564  6105  6126#
FSCMD2  037654  6106  6140#
FSCMD3  040026  6107  6168#
FSCMD4  040272  6108  6206#
FSCMD5  040600  6109  6255#
FSCMD6  041454  6110  6373#
FSCMD7  041462  6111  6379#
FSCMD8  041776  6112  6424#
FSCMD9  042370  6113  6477#
FSINFL  002126  1142#  2068#  2773  2774*
FSPAT  041254  6325*  6329#
FSSTAC  001752  1082#  6128*  6136  6142*  6164  6258*  6362  6399*  6403  6426*  6469  6479*  6536
                 6568*  6611
FS1  037446  6096#  6102  6118
GALWOR  001742  1078#  4679*  4714*  4715  4717  4718*  4721
GDSWIT  002060  1121#  1804*  9320
GETDAT  050324  7324*  7339*  7458#
GETDA1  050442  7460*  7468  7473#
GETDIS  050650  3372*  3405*  7520*  7524#
GETSIZ  014472  1458*  2596*  2625#
GOOD  001432  1001#  1493*  1800*  7205*  7211*  7216*  7223*  7226*  7233*  7238*  7243*  7248*  7253*
                 7258*  7263*  7268*  7273*  7278*  7283*  7288*  7293*  7298*  7303*  7313*  7315*  7371*
                 7385  7391*  7419*  7421*  7451*  9312  9318  9320  9327
GOOD2  001434  1002#  7354*  7358*  9327
GOOD3  001436  1003#  7355*  7359*  9327
GTSWR =  104407  86#  1415
HEADER  002146  1153#  1274*  2018*  2021*  2446*  2459*  2486*  3768*  3771*  3782*  3785*  3840*  3843*
                 3854*  3857*  3926*  3929*  4807*  4818*  6312*  6457*  6599*  7424*  7440*  7445*  7453*
                 7455*  7643  7655  7689*  7696*  7918  7919*  7932  7933*  7936*
HIADRS=  177742  270#  9316  9326
HIBANK  002076  1130#  1793*
HIPAT  036366  2353*  2394*  5901#
HT  =  000011  179#  8218
I  002130  1143#  1319*  1325*  1953*  1954*  1956  2485*  2488  2490*  3271*  3280  3283  3288*
                 3289*  3299  3302  3307*  7003*  7016  7024
```

IIII = 177777

73# 1249# 1319# 1321# 1323# 1325# 1329# 1334# 1353# 1355# 1356# 1358# 1359#
1366# 1367# 1370# 1372# 1374# 1375# 1412# 1446# 1447# 1456# 1492# 1495# 1546#
1548# 1550# 1637# 1639# 1640# 1642# 1651# 1655# 1656# 1667# 1675# 1683# 1691#
1692# 1693# 1694# 1698# 1703# 1704# 1705# 1709# 1711# 1713# 1718# 1720# 1721#
1723# 1724# 1726# 1727# 1728# 1731# 1735# 1742# 1743# 1749# 1750# 1752# 1753#
1755# 1757# 1758# 1759# 1787# 1790# 1794# 1815# 1816# 1817# 1818# 1819# 1820#
1821# 1822# 1828# 1829# 1832# 1833# 1836# 1837# 1840# 1853# 1860# 1861# 1862#
1865# 1866# 1877# 1883# 1886# 1887# 1888# 1889# 1894# 1895# 1896# 1897# 1898#
1905# 1911# 1915# 1919# 1921# 1923# 1924# 1927# 1932# 1933# 1934# 1935# 1937#
1938# 1939# 1954# 1962# 1965# 1968# 1969# 1972# 1974# 1976# 1985# 1987# 1988#
1990# 1992# 1993# 1994# 1996# 1997# 2028# 2030# 2033# 2035# 2036# 2037# 2048#
2049# 2051# 2052# 2054# 2073# 2076# 2102# 2110# 2143# 2176# 2216# 2256# 2298#
2346# 2387# 2435# 2466# 2467# 2469# 2471# 2473# 2475# 2478# 2479# 2480# 2482#
2485# 2490# 2494# 2495# 2496# 2497# 2498# 2499# 2500# 2504# 2505# 2506# 2510#
2527# 2532# 2534# 2559# 2578# 2604# 2621# 2641# 2642# 2648# 2649# 2655# 2692#
2711# 2713# 2715# 2727# 2729# 2731# 2743# 2745# 2747# 2759# 2761# 2763# 2773#
2776# 2787# 2789# 2791# 2803# 2805# 2807# 2819# 2821# 2823# 2851# 2852# 2857#
2859# 2861# 3057# 3058# 3059# 3060# 3066# 3067# 3068# 3069# 3075# 3076# 3077#
3084# 3085# 3086# 3087# 3094# 3095# 3096# 3097# 3103# 3104# 3105# 3106# 3117#
3118# 3119# 3124# 3127# 3130# 3132# 3137# 3138# 3151# 3153# 3156# 3158# 3159#
3162# 3164# 3165# 3169# 3171# 3172# 3175# 3177# 3178# 3233# 3234# 3235# 3271#
3272# 3274# 3280# 3282# 3283# 3285# 3286# 3287# 3288# 3289# 3290# 3292# 3299#
3301# 3302# 3304# 3305# 3306# 3307# 3320# 3322# 3323# 3327# 3328# 3329# 3330#
3334# 3338# 3345# 3790# 3791# 3872# 3873# 3935# 3936# 4017# 4018# 4092# 4093#
4170# 4171# 4501# 4502# 4505# 4508# 4509# 4515# 4516# 4517# 4518# 4521# 4524#
4527# 4536# 4537# 4538# 4539# 4541# 4543# 4545# 4547# 4548# 4550# 4552# 4554#
4556# 4557# 4560# 4563# 4564# 4568# 4570# 4572# 4575# 4960# 5077# 5091# 5093#
5094# 5096# 5098# 5099# 5267# 5275# 5277# 5279# 5281# 5290# 5298# 5300# 5302#
5304# 5344# 5346# 5348# 5362# 5368# 5384# 5390# 5397# 5404# 5414# 5415# 5417#
5419# 5422# 5423# 5424# 5425# 5426# 5429# 5431# 5433# 5447# 5448# 5450# 5452#
5455# 5456# 5457# 5458# 5459# 5462# 5464# 5466# 5497# 5504# 5512# 5513# 5515#
5517# 5518# 5519# 5520# 5640# 5641# 5642# 5644# 5652# 5653# 5655# 5656# 5657#
5658# 5659# 5660# 5661# 5662# 5663# 5666# 5667# 5668# 5669# 5670# 5671# 5672#
5674# 5676# 5687# 5689# 5848# 5849# 5851# 5853# 5915# 5917# 5927# 5929# 5930#
5932# 5954# 5956# 5958# 5960# 5962# 5970# 5971# 5972# 5979# 5980# 5981# 5995#
6014# 6016# 6019# 6031# 6032# 6034# 6084# 6088# 6089# 6093# 6103# 6117# 6181#
6183# 6184# 6219# 6221# 6222# 6263# 6267# 6270# 6275# 6276# 6280# 6285# 6287#
6288# 6291# 6295# 6298# 6316# 6318# 6320# 6381# 6383# 6385# 6408# 6410# 6417#
6418# 6431# 6433# 6434# 6437# 6439# 6441# 6443# 6460# 6462# 6464# 6465# 6481#
6483# 6485# 6495# 6497# 6498# 6503# 6505# 6506# 6509# 6548# 6550# 6554# 6560#
6561# 6562# 6573# 6575# 6576# 6579# 6581# 6583# 6585# 6602# 6604# 6606# 6607#
6660# 6669# 6670# 6675# 6676# 6678# 6687# 6688# 6689# 6690# 6693# 6696# 6701#
6705# 6736# 6739# 6743# 6744# 6749# 6753# 6757# 6760# 6761# 6762# 6766# 6767#
6770# 6773# 6774# 6775# 6776# 6777# 6778# 6782# 6786# 6787# 6791# 6793# 6798#
6799# 6800# 6806# 6809# 6810# 6811# 6812# 6813# 6817# 6821# 6823# 6828# 6830#
6835# 6836# 6837# 6842# 6845# 6846# 6847# 6848# 6849# 6853# 6859# 6862# 6864#
6866# 6871# 6872# 6877# 6879# 6880# 6881# 6883# 6890# 6898# 6921# 6928# 6937#
6944# 6973# 6974# 7042# 7047# 7049# 7051# 7053# 7054# 7056# 7307# 7311# 7312#
7314# 7316# 7321# 7323# 7325# 7326# 7328# 7330# 7336# 7338# 7340# 7341# 7343#
7345# 7393# 7397# 7418# 7420# 7422# 7428# 7430# 7431# 7433# 7434# 7436# 7437#
7439# 7485# 7488# 7490# 7493# 7494# 7496# 7497# 7560# 7572# 7576# 7577# 7580#
7584# 7586# 7587# 7589# 7594# 7596# 7600# 7601# 7604# 7605# 7610# 7611# 7612#
7691# 7693# 7694# 7697# 7796# 7800# 7801# 7803# 7805# 7809# 7811# 7812# 7813#
7814# 7815# 7817# 7818# 7822# 7825# 7827# 7828# 7838# 7840# 7842# 7844# 7846#
7861# 7862# 7863# 7866# 7871# 7880# 7881# 7882# 7883# 7886# 7887# 7888# 7890#
7898# 7900# 7902# 7905# 7907# 7909# 7942# 7947# 7954# 7955# 7960# 7961# 7963#

F02


```
                    7969#  7970#  7975#  7976#  7978#  8004#  8024#  8025#  8027#  8030#  8031#  8032#  8033#
                    8076#  8077#  8079#  8082#  8083#  8084#  8085#  8470#  8472#  10301
IISTAC  002162      1159#  1253   1843*  1844   1849*  6664*  6717*  6720*  6723*  6873
IISTAD  002164      1160#  1850*  6665*  6666*  6721*  6722*  6724
IISTID  010112      1831   1835   1839   1843#
IISTIN  044266      1848*  6663*  6715#
IISTOK  010140      1848#  7037
IISTVE  002166      1161#  6657
INCBNK  036376      2136*  2169*  2204*  2244*  2282*  2325*  2371*  2414*  5905#
INCMAR  036412      2130*  2198*  2279*  2322*  5911#
INCPAT  036352      2133*  2201*  5893#
INCRPT  036352      2285*  2328*  5892#
INTERL  001642      1041#  1457*  1470
INTER0  005562      1448   1506#
INTER1  005564      1449   1508#
INTER2  005564      1450   1509#
INTER3  005572      1451   1512#
INTER4  005572      1452   1513#
INTER5  005600      1453   1517#
INTER6  005572      1454   1514#
INTER7  005600      1455   1518#
INTKRA  014344      2470*  2594#  3129#
INTOVR  002042      1114#  2630*  2632*  5267   5290   6155*  6157*
INT..0  015006      2605   2704#
INT..1  015032      2606   2710#  2777
INT..2  015076      2607   2720#
INT..3  015122      2608   2726#
INT..4  015166      2609   2736#
INT..5  015212      2610   2742#
INT..6  015256      2611   2752#
INT..7  015302      2612   2758#
INT.10  015346      2613   2769#
INT.11  015346      2614   2773#
INT.12  015370      2615   2780#
INT.13  015414      2616   2786#
INT.14  015460      2617   2796#
INT.15  015504      2618   2802#
INT.16  015550      2619   2812#
INT.17  015574      2620   2818#
IOTVEC= 000020       257#  1284*  1285*
ISECCO  014034      2468*  2516#
KDIAG = 000010      4500#  4516   4537   4563
KDPAR0= 172360       403#  3000   3017   3028   3251   3511   3548   3570   4824   4833
KDPAR7= 172376       410#
KDPDR7= 172336       390#
KERLIM  055272      8183#  8184
KERNEL= 104417        97#  1581   1594   1611   1633   1702   2580   2588   2848   3356   3398   3431   5683
                    5716   5767   5783   6048   6190   6201   6228   6246   6250   6878   6888   6903   6910
                    6919   6935   6968   6981   6993   6999   7019   7055   7059   7310   7375   7396   8016
                    8020   8098   8102   8147   8178
KERSTK= 001400       166#  7947
KIPAR0= 172340       393#  5600   5625   5789   5808   8138   8153
KIPDR0= 172300       373#  5627   5791   8006   8066   8087   8154
KMAP  = 104422       102#  1430   6731
KPFLAG  001504      1023#  1648*  1724   1753   5362   5368   5384   5390   5397   5404   5497   5504   5834*
                    5844#
```

LAST = 157776
LASTB 001640
LASTER 001410
LBLS0 = 000547

490# 2495 2976 3147 3194 4197 4305 4515 4536 4572 4603 4605 4727
4889 6184 6222 6498 7042
1040# 1465 1488 2598 2638* 2641* 2642* 2646* 2647* 2649* 7865* 7866
992# 1810* 2049 2069*
1249# 1319# 1325 1329# 1334 1353# 1355 1356# 1358 1359# 1366 1367# 1375
1412# 1446# 1447# 1456 1492# 1495 1546# 1548# 1550 1637# 1639 1640# 1642
1651# 1705 1709# 1711# 1713 1718# 1743 1749# 1759 1787# 1790 1794# 1815#
1816# 1817# 1818# 1819# 1820# 1821# 1822# 1828# 1829# 1832 1833# 1836 1837#
1840 1853# 1860# 1861# 1862# 1865 1866# 1877 1883# 1898 1905# 1911 1915#
1927 1932# 1939 1954# 1962# 1965# 1997 2028# 2037 2048# 2052 2054# 2073#
2076 2102# 2110 2143# 2176# 2216# 2256# 2298# 2346# 2387# 2435# 2466# 2510
2527# 2532# 2534 2559# 2578# 2604# 2621 2641# 2642# 2648# 2649# 2655# 2692
2711# 2713# 2715 2727# 2729# 2731 2743# 2745# 2747 2759# 2761# 2763 2773#
2776 2787# 2789# 2791 2803# 2805# 2807 2819# 2821# 2823 2851# 2852# 2857#
2859# 2861 3057# 3058# 3060 3066# 3067# 3069 3075# 3077 3084# 3085# 3087
3094# 3095# 3097 3103# 3104# 3106 3117# 3119 3124# 3138 3151# 3153 3156#
3159 3162# 3165 3169# 3172 3175# 3178 3233# 3235 3271# 3288 3289# 3307
3320# 3329 3330# 3345 3790# 3791# 3872# 3873# 3935# 3936# 4017# 4018# 4092#
4093# 4170# 4171# 4501# 4564 4568# 4570 4572# 4575 4960# 5077# 5091# 5093#
5099 5267# 5281 5290# 5304 5344# 5346# 5348 5362# 5368# 5384# 5390# 5397#
5404# 5414# 5426 5429# 5431# 5433 5447# 5459 5462# 5464# 5466 5497# 5504#
5512# 5520 5640# 5641# 5655 5656# 5676 5687# 5689 5848# 5853 5915# 5917
5927# 5929 5930# 5932 5954# 5956 5958# 5960# 5962 5970# 5981 5995# 6014#
6016# 6019 6031# 6032# 6034 6084# 6088 6089# 6093 6103# 6117 6181# 6183#
6184# 6219# 6221# 6222# 6263# 6267# 6270 6275# 6276# 6280 6285# 6287 6288#
6291 6295# 6298 6316# 6318# 6320 6381# 6383# 6385 6408# 6418 6431# 6433
6434# 6437 6439# 6441# 6443 6460# 6465 6481# 6483# 6485 6495# 6497# 6498#
6503# 6505 6506# 6509 6548# 6562 6573# 6575 6576# 6579 6581# 6583# 6585
6602# 6607 6660# 6669 6670# 6690 6693# 6696 6701# 6705 6736# 6778 6782#
6813 6817# 6849 6853# 6859 6862# 6864# 6866 6871# 6880 6881# 6883 6890#
6898 6921# 6928 6937# 6944 6973# 6974# 7042# 7056 7307# 7311 7312# 7314#
7316 7321# 7323# 7325 7326# 7328# 7330 7336# 7338# 7340 7341# 7343# 7345
7393# 7397 7418# 7420# 7422 7428# 7430 7431# 7433 7434# 7436 7437# 7439
7485# 7488 7490# 7493 7494# 7496 7497# 7560# 7572 7576# 7587 7589# 7612
7691# 7693 7694# 7697 7796# 7828 7838# 7846 7861# 7887 7888# 7890 7898#
7900# 7902# 7905# 7907# 7909# 7942# 7947 7954# 7961# 7963 7969# 7976# 7979
8004# 8024# 8033 8076# 8085 8470# 8472

LBLS1 = 000125
1321# 1323 1370# 1372# 1374 1546# 1655# 1704 1709# 1720# 1731 1735# 1742
1750# 1758 1787# 1790 1886# 1897 1915# 1919# 1924 1933# 1938 1968# 1993
1994# 1996 2030# 2036 2049# 2051 2073# 2467# 2506 2857# 3058# 3059# 3067#
3068# 3075# 3076# 3085# 3086# 3095# 3096# 3104# 3105# 3117# 3118# 3127# 3137
3151# 3156# 3158# 3162# 3164# 3169# 3171# 3175# 3177# 3233# 3234# 3272# 3287
3290# 3306 3322# 3328 3334# 3338 4502# 4505# 4508 4509# 4563 4572# 4575
5094# 5096# 5098 5275# 5277# 5279 5298# 5300# 5302 5415# 5425 5448# 5458
5513# 5519 5642# 5653 5657# 5671 5672# 5674 5849# 5851 5954# 5971# 5980
6014# 6285# 6410# 6417 6431# 6462# 6464 6503# 6550# 6561 6573# 6604# 6606
6675# 6689 6739# 6777 6786# 6812 6821# 6848 6872# 6879 7042# 7056 7577#
7580 7584# 7586 7594# 7596 7600# 7611 7800# 7815 7817# 7827 7840# 7842#
7844 7862# 7886 7955# 7960 7970# 7975 8025# 8032 8077# 8084

LBLS10= 000127
2482# 2494

LBLS11= 000025
2485# 2490

LBLS2 = 000547
1656# 1703 1721# 1728 1752# 1757 1887# 1896 1921# 1923 1934# 1935# 1937
1969# 1992 1994# 2033# 2035 2469# 2505 3130# 3132 3274# 3286 3292# 3305
3323# 3327 4515# 4527 4536# 4560 5417# 5423 5424# 5450# 5456 5457# 5515#
5517 5518# 5644# 5652 5658# 5670 5972# 5979 6554# 6560 6676# 6688 6743#
6776 6787# 6791 6793# 6798 6799# 6811 6823# 6828 6830# 6835 6836# 6847

H02


```
                    6877#  7047#  7054   7601#  7604   7605#  7610   7801#  7814   7818#  7822#  7825   7863#
                    7883   8027#  8030   8031#  8079#  8082   8083#
LBLS3 = 000527      1667#  1693   1694#  1698   1723#  1727   1753#  1755   1888#  1895   1972#  1974#  1976
                    1985#  1987   1988#  1990   2471#  2499   2500#  2504   3280#  3282   3283#  3285   3299#
                    3301   3302#  3304   4515#  4527   4536#  4560   5419#  5422   5452#  5455   5659#  5669
                    5972#  6678#  6687   6744#  6775   6800#  6810   6837#  6846   7049#  7051#  7053   7601#
                    7803#  7812   7813#  7866#  7862
LBLS4 = 000531      1675#  1692   1724#  1726   1889#  1894   2473#  2498   4516#  4517#  4518#  4521#  4524
                    4537#  4538#  4539#  4548#  4557   5660#  5668   6749#  6753   6757#  6761   6762#  6765
                    6767#  6774   6806#  6809   6842#  6845   7805#  7811   7871#  7880   7881#
LBLS5 = 000520      1683#  1691   1724#  2475#  2497   4541#  4543   4545#  4547   4550#  4552   4554#  4556
                    5661#  5662#  5667   6760#  6770#  6773   6806#  7809#
LBLS6 = 000336      2478#  2479#  2496   5663#  5666
LBLS7 = 000024      2480#  2495
LCSROU= 000046      5512#  5518
LCSRRE= 000124      8076#  8083
LCSRSA= 000122      8024#  8031
LF    = 000012       180#  8245
LKS   = 177546       176#  7007*  7013*
LOADBA  002116      1138#  5646*  5702   5706
LOADCS= 104425       107#  1922   2631   5370   5380   5392   5406   5492   5506   5516   6156   8081
LOADER= 000050      5641#  5651
LOADHO  002114      1137#  1732*  1735   1737   2099   5647   5703   6036   6708
LOADRS= 177740       269#  5173   9316   9326
LOOP    011254      1999#  2111
LOWBNK  001736      1075#  4598*  4609   4643
LOWMAP  035524      1425#  5691*  5724*  5729#  8063*
LST$$ = 000000        70#   978   1249   1251   1254   1256   1272   1274   1319   1321   1323   1325   1329
                    1334   1340   1348   1353   1354   1355   1356   1358   1359   1360   1361   1362   1363
                    1366   1367   1368   1369   1370   1371   1372   1373   1374   1375   1384   1387   1396
                    1397   1403   1408   1412   1431   1436   1437   1446   1447   1456   1479   1482   1492
                    1493   1495   1500   1501   1527   1537   1541   1543   1546   1548   1550   1556   1557
                    1569   1579   1585   1592   1609   1625   1630   1636   1637   1639   1640   1642   1651
                    1655   1656   1657   1659   1667   1675   1683   1691   1692   1693   1694   1698   1703
                    1704   1705   1706   1709   1711   1713   1718   1720   1721   1723   1724   1725   1726
                    1727   1728   1731   1735   1742   1743   1745   1749   1750   1752   1753   1755   1756
                    1757   1758   1759   1765   1777   1786   1787   1790   1794   1815   1816   1817   1818
                    1819   1820   1821   1822   1828   1829   1832   1833   1836   1837   1840   1853   1857
                    1859   1860   1861   1862   1864   1865   1866   1872   1873   1874   1877   1883   1886
                    1887   1888   1889   1891   1892   1894   1895   1896   1897   1898   1902   1903   1905
                    1909   1910   1911   1915   1919   1921   1923   1924   1927   1932   1933   1934   1935
                    1936   1937   1938   1939   1953   1954   1955   1956   1957   1958   1959   1960   1961
                    1962   1965   1968   1969   1972   1974   1976   1985   1987   1988   1990   1992   1993
                    1994   1996   1997   2018   2019   2021   2027   2028   2030   2033   2035   2036   2037
                    2048   2049   2051   2052   2054   2072   2073   2074   2076   2077   2083   2085   2102
                    2110   2143   2176   2216   2256   2298   2346   2387   2435   2446   2447   2459   2463
                    2466   2467   2469   2471   2473   2475   2476   2478   2479   2480   2482   2485   2486
                    2488   2490   2492   2493   2494   2495   2496   2497   2498   2499   2500   2503   2504
                    2505   2506   2510   2511   2512   2527   2528   2530   2531   2532   2534   2540   2559
                    2578   2583   2591   2595   2604   2621   2622   2627   2630   2641   2642   2648   2649
                    2650   2655   2692   2704   2705   2706   2710   2711   2712   2713   2714   2715   2716
                    2720   2721   2722   2726   2727   2728   2729   2730   2731   2732   2736   2737   2738
                    2742   2743   2744   2745   2746   2747   2748   2752   2753   2754   2758   2759   2760
                    2761   2762   2763   2764   2773   2774   2775   2776   2777   2780   2781   2782   2786
                    2787   2788   2789   2790   2791   2792   2796   2797   2798   2802   2803   2804   2805
                    2806   2807   2808   2812   2813   2814   2818   2819   2820   2821   2822   2823   2824
```

2834 2851 2852 2853 2857 2859 2861 2962 2971 2980 2995 2999 3000
3001 3016 3017 3027 3028 3057 3058 3059 3060 3066 3067 3068 3069
3075 3076 3077 3084 3085 3086 3087 3094 3095 3096 3097 3103 3104
3105 3106 3117 3118 3119 3124 3127 3130 3132 3134 3136 3137 3138
3149 3151 3152 3153 3154 3156 3158 3159 3160 3162 3164 3165 3167
3169 3171 3172 3173 3175 3177 3178 3179 3188 3197 3200 3204 3207
3233 3234 3235 3250 3251 3252 3255 3267 3270 3271 3272 3274 3275
3280 3282 3283 3285 3286 3287 3288 3289 3290 3292 3293 3299 3301
3302 3304 3305 3306 3307 3317 3319 3320 3322 3323 3327 3328 3329
3330 3331 3334 3338 3345 3353 3354 3355 3366 3371 3373 3393 3400
3404 3406 3426 3433 3599 3618 3768 3771 3782 3785 3788 3790 3791
3794 3840 3843 3854 3857 3870 3872 3873 3876 3926 3929 3933 3935
3936 3939 4015 4017 4018 4021 4090 4092 4093 4096 4168 4170 4171
4174 4300 4420 4501 4502 4503 4504 4505 4506 4507 4508 4509 4513
4514 4515 4516 4517 4518 4519 4520 4521 4522 4523 4524 4525 4526
4527 4533 4534 4536 4537 4538 4539 4540 4541 4543 4544 4545 4547
4548 4549 4550 4552 4553 4554 4556 4557 4558 4559 4560 4563 4564
4568 4570 4571 4572 4574 4575 4751 4807 4818 4908 4932 4940 4948
4956 4957 4958 4960 4965 4967 4968 4969 4970 4971 4972 4982 4983
4984 4985 4991 5010 5013 5015 5028 5038 5047 5076 5077 5087 5091
5093 5094 5096 5098 5099 5100 5110 5124 5133 5141 5143 5178 5194
5218 5222 5229 5260 5267 5268 5275 5277 5279 5280 5281 5283 5290
5291 5298 5300 5302 5303 5304 5307 5311 5325 5329 5330 5343 5344
5346 5348 5352 5362 5368 5384 5390 5397 5404 5411 5414 5415 5417
5419 5420 5421 5422 5423 5424 5425 5426 5428 5429 5431 5433 5444
5447 5448 5450 5452 5453 5454 5455 5456 5457 5458 5459 5461 5462
5464 5466 5497 5504 5510 5512 5513 5515 5517 5518 5519 5520 5521
5533 5549 5599 5620 5623 5635 5640 5641 5642 5644 5651 5652 5653
5654 5655 5656 5657 5658 5659 5660 5661 5662 5663 5665 5666 5667
5668 5669 5670 5671 5672 5673 5674 5675 5676 5679 5681 5682 5687
5689 5696 5697 5701 5705 5710 5712 5714 5715 5726 5730 5738 5765
5766 5781 5782 5832 5835 5844 5845 5848 5849 5851 5853 5860 5864
5865 5873 5878 5915 5916 5917 5927 5928 5929 5930 5931 5932 5954
5955 5956 5958 5960 5962 5970 5971 5972 5977 5979 5980 5981 5993
5995 6014 6016 6019 6031 6032 6034 6043 6044 6058 6063 6067 6074
6082 6084 6085 6088 6089 6090 6093 6094 6096 6098 6101 6103 6117
6122 6129 6131 6133 6135 6137 6143 6145 6146 6148 6149 6151 6152
6154 6155 6159 6160 6163 6165 6169 6171 6172 6175 6181 6183 6184
6186 6187 6188 6191 6194 6198 6202 6203 6207 6209 6210 6213 6219
6221 6222 6224 6225 6226 6231 6235 6238 6239 6240 6242 6243 6247
6248 6251 6252 6256 6259 6260 6262 6263 6267 6268 6269 6270 6272
6274 6275 6276 6277 6279 6280 6282 6284 6285 6286 6287 6288 6289
6290 6291 6293 6295 6296 6297 6298 6300 6312 6313 6316 6318 6320
6364 6366 6368 6369 6380 6381 6383 6385 6386 6390 6391 6392 6393
6405 6407 6408 6410 6411 6417 6418 6419 6420 6425 6427 6428 6430
6431 6432 6433 6434 6435 6436 6437 6439 6441 6443 6444 6446 6457
6458 6460 6462 6464 6465 6466 6467 6472 6473 6478 6480 6481 6483
6485 6486 6489 6495 6497 6498 6500 6502 6503 6504 6505 6506 6507
6508 6509 6513 6514 6515 6517 6528 6539 6540 6545 6546 6547 6548
6549 6550 6554 6555 6558 6559 6560 6561 6562 6563 6567 6569 6570
6572 6573 6574 6575 6576 6577 6578 6579 6581 6583 6585 6586 6588
6599 6600 6602 6604 6606 6607 6608 6609 6614 6615 6621 6622 6624
6627 6637 6639 6660 6663 6670 6675 6676 6678 6686 6687 6688 6689
6690 6693 6696 6701 6705 6716 6726 6736 6739 6743 6744 6749 6750
6752 6753 6757 6760 6761 6762 6763 6765 6766 6767 6770 6773 6774
6775 6776 6777 6778 6782 6786 6787 6791 6793 6798 6799 6800 6806

```
                    6808    6809    6810    6811    6812    6813    6817    6821    6823    6828    6830    6835    6836
                    6837    6842    6844    6845    6846    6847    6848    6849    6853    6859    6862    6864    6866
                    6869    6870    6871    6872    6876    6877    6879    6880    6881    6882    6883    6884    6889
                    6890    6898    6899    6900    6901    6908    6912    6917    6920    6921    6928    6933    6936
                    6937    6944    6950    6963    6966    6973    6974    6979    6985    6989    6990    6991    6997
                    7001    7004    7015    7017    7020    7023    7026    7040    7042    7046    7047    7049    7051
                    7053    7054    7056    7057    7206    7212    7218    7241    7242    7243    7246    7247    7248
                    7251    7252    7253    7256    7257    7258    7261    7262    7263    7266    7267    7268    7271
                    7272    7273    7276    7277    7278    7281    7282    7283    7286    7287    7288    7291    7292
                    7293    7296    7297    7298    7301    7302    7303    7306    7307    7308    7309    7311    7312
                    7313    7314    7315    7316    7321    7322    7323    7325    7326    7328    7330    7336    7337
                    7338    7340    7341    7343    7345    7354    7355    7356    7359    7360    7363    7370    7372
                    7380    7384    7388    7391    7392    7393    7394    7395    7397    7398    7402    7412    7413
                    7418    7420    7422    7424    7428    7430    7431    7433    7434    7436    7437    7439    7440
                    7445    7453    7455    7459    7471    7485    7488    7490    7493    7494    7496    7497    7527
                    7528    7541    7560    7569    7572    7576    7577    7578    7579    7580    7584    7585    7586
                    7587    7589    7594    7596    7600    7601    7604    7605    7606    7610    7611    7612    7627
                    7632    7647    7648    7650    7657    7659    7684    7691    7693    7694    7696    7697    7703
                    7709    7715    7718    7721    7722    7734    7740    7748    7750    7761    7776    7781    7787
                    7788    7789    7794    7796    7797    7799    7800    7801    7803    7805    7809    7810    7811
                    7812    7813    7814    7815    7816    7817    7818    7822    7825    7827    7828    7829    7830
                    7835    7836    7838    7840    7842    7844    7845    7846    7847    7854    7857    7861    7862
                    7863    7866    7871    7880    7881    7882    7883    7886    7887    7888    7890    7891    7893
                    7896    7898    7900    7902    7905    7907    7909    7918    7919    7932    7933    7936    7940
                    7941    7942    7943    7944    7945    7946    7947    7952    7953    7954    7955    7956    7957
                    7958    7959    7960    7961    7962    7963    7967    7968    7969    7970    7971    7972    7973
                    7974    7975    7976    7977    7978    7980    7989    7993    7997    8002    8004    8014    8017
                    8018    8021    8024    8025    8027    8029    8030    8031    8032    8033    8035    8039    8042
                    8044    8060    8062    8068    8072    8076    8077    8079    8080    8082    8083    8084    8085
                    8095    8096    8099    8100    8106    8111    8116    8118    8120    8128    8130    8131    8134
                    8137    8148    8164    8166    8167    8179    8224    8311    8339    8375    8376    8414    8415
                    8416    8417    8426    8434    8443    8464    8469    8470    8471    8472    8473    8474    8475
                    8479    8480    8481    8482    8483    8530    8539    8545    8550    8554    8559    8560    8562
                    8565    8569    8600    8624    8628    8630    8631    8632    8651    8682    8687    8689    8690
                    8691    8712    8727    8741    8756    9001    9236

LWDBE  =  000044     5447#   5454    5457
LWSBE  =  000042     5414#   5421    5424
L0        003636     1249#
L1        004314     1321    1323#
L10       004646     1372    1374#
L100      011216     1988    1990#
L101      011252     1994#
L102      011254     1994    1996#
L103      011440     2030    2036#
L104      011440     2033    2035#
L105      011514     2048    2052#
L106      011514     2049    2051#
L107      011526     2054#
L11       005222     1446#
L110      011604     2073#
L111      011614     2073    2076#
L112      011750     2102    2110#
L113      012056     2143#
L114      012172     2176#
L115      012330     2216#
L116      012466     2256#
```

```
L117    012624          2298#
L12     005252          1447    1456#
L120    013004          2346#
L121    013142          2387#
L122    013322          2435#
L123    013766          2467    2506#
L124    013710          2475    2497#
L125    013572          2478#
L126    013710          2479    2496#
L127    013672          2482    2494#
L13     005516          1492    1495#
L130    013752          2500    2504#
L131    014122          2527    2532#
L132    014126          2532    2534#
L133    014460          2604    2621#
L134    014570          2641#
L135    014602          2642#
L136    014644          2648#
L137    014662          2649#
L14     005756          1546#
L140    014776          2655    2692#
L141    015060          2711    2713#
L142    015066          2713    2715#
L143    015150          2727    2729#
L144    015156          2729    2731#
L145    015240          2743    2745#
L146    015246          2745    2747#
L147    015330          2759    2761#
L15     005762          1546    1548#
L150    015336          2761    2763#
L151    015366          2773    2776#
L152    015442          2787    2789#
L153    015450          2789    2791#
L154    015532          2803    2805#
L155    015540          2805    2807#
L156    015622          2819    2821#
L157    015630          2821    2823#
L16     005764          1548    1550#
L160    015746          2851#
L161    015770          2852#
L162    016012          2857#
L163    016016          2857    2859#
L164    016020          2859    2861#
L165    017122          3057#
L166    017136          3058#
L167    017146          3058    3060#
L17     006470          1637    1639#
L170    017146          3059#
L171    017200          3066#
L172    017214          3067#
L173    017224          3067    3069#
L174    017224          3068#
L175    017262          3075#
L176    017272          3075    3077#
L177    017272          3076#
L2      004360          1329    1334#
```

```
L20     006504          1640    1642#
L200    017332          3084#
L201    017346          3085#
L202    017356          3085    3087#
L203    017356          3086#
L204    017410          3094#
L205    017424          3095#
L206    017434          3095    3097#
L207    017434          3096#
L21     006774          1655    1704#
L210    017466          3103#
L211    017502          3104#
L212    017512          3104    3106#
L213    017512          3105#
L214    017572          3117#
L215    017602          3117    3119#
L216    017602          3118#
L217    017700          3127    3137#
L22     006774          1656    1703#
L220    017670          3130    3132#
L221    020000          3151#
L222    020002          3151    3153#
L223    020032          3156#
L224    020036          3156    3159#
L225    020066          3162#
L226    020072          3162    3165#
L227    020126          3169#
L23     006746          1667    1693#
L230    020132          3169    3172#
L231    020162          3175#
L232    020166          3175    3178#
L233    020456          3233#
L234    020466          3233    3235#
L235    020466          3234#
L236    021012          3274    3286#
L237    020776          3280    3282#
L24     006746          1675    1692#
L240    021012          3283    3285#
L241    021150          3292    3305#
L242    021134          3299    3301#
L243    021150          3302    3304#
L244    021322          3322    3328#
L245    021322          3323    3327#
L246    021420          3330    3345#
L247    021372          3334    3338#
L25     006746          1683    1691#
L250    027124          4502    4505#
L251    027134          4505    4508#
L252    027246          4515    4527#
L253    027176          4516#
L254    027210          4517#
L255    027226          4518    4521#
L256    027234          4521    4524#
L257    027410          4536    4560#
L26     006762          1694    1698#
L260    027310          4537#
```

```
L261    027322          4538#
L262    027354          4539    4548#
L263    027340          4541    4543#
L264    027352          4545    4547#
L265    027376          4548    4557#
L266    027364          4550    4552#
L267    027376          4554    4556#
L27     007046          1709#
L270    027510          4572    4575#
L271    031264          4960#
L272    031730          5077#
L273    032000          5091    5093#
L274    032026          5093    5099#
L275    032020          5094    5096#
L276    032026          5096    5098#
L277    032730          5267    5281#
L3      004500          1353    1355#
L30     007052          1709    1711#
L300    032716          5275    5277#
L301    032724          5277    5279#
L302    033040          5290    5304#
L303    033026          5298    5300#
L304    033034          5300    5302#
L305    033252          5344    5346#
L306    033260          5346    5348#
L307    033336          5362#
L31     007054          1711    1713#
L310    033372          5368#
L311    033462          5384#
L312    033516          5390#
L313    033556          5397#
L314    033620          5404#
L315    033672          5417    5423#
L316    033672          5419    5422#
L317    033734          5429    5431#
L32     007250          1718    1743#
L320    033742          5431    5433#
L321    034036          5450    5456#
L322    034036          5452    5455#
L323    034100          5462    5464#
L324    034106          5464    5466#
L325    034236          5497#
L326    034272          5504#
L327    034322          5515    5517#
L33     007140          1723    1727#
L330    034734          5640#
L331    035024          5644    5652#
L332    035152          5658    5670#
L333    035152          5659    5669#
L334    035152          5660    5668#
L335    035152          5662    5667#
L336    035152          5663    5666#
L337    035200          5672    5674#
L34     007136          1724#
L340    035270          5687    5689#
L341    036200          5849    5851#
```

```
L342    036432          5915    5917#
L343    036520          5927    5929#
L344    036540          5930    5932#
L345    036634          5954#
L346    036636          5954    5956#
L347    036662          5958    5960#
L35     007140          1724    1726#
L350    036670          5960    5962#
L351    037012          5970    5981#
L352    037012          5971    5980#
L353    036770          5972#
L354    037012          5972    5979#
L355    037046          5995#
L356    037144          6014#
L357    037150          6014    6016#
L36     007250          1735    1742#
L360    037156          6016    6019#
L361    037176          6031#
L362    037210          6032    6034#
L363    037416          6084    6088#
L364    037434          6089    6093#
L365    037534          6103    6117#
L366    040722          6267    6270#
L367    040762          6276    6280#
L37     007316          1752    1757#
L370    041014          6285#
L371    041016          6285    6287#
L372    041042          6288    6291#
L373    041076          6295    6298#
L374    041216          6316    6318#
L375    041220          6318    6320#
L376    041526          6381    6383#
L377    041530          6383    6385#
L4      004516          1356    1358#
L40     007314          1753    1755#
L400    041734          6410    6417#
L401    042070          6431#
L402    042072          6431    6433#
L403    042116          6434    6437#
L404    042132          6439    6441#
L405    042134          6441    6443#
L406    042256          6462    6464#
L407    042440          6481    6483#
L41     007532          1787#
L410    042442          6483    6485#
L411    042556          6503#
L412    042560          6503    6505#
L413    042604          6506    6509#
L414    043162          6548    6562#
L415    043146          6554    6560#
L416    043270          6573#
L417    043272          6573    6575#
L42     007544          1787    1790#
L420    043316          6576    6579#
L421    043332          6581    6583#
L422    043334          6583    6585#
```

```
L423    043456          6604    6606#
L424    044070          6678    6687#
L425    044636          6743    6776#
L426    044502          6749    6753#
L427    044532          6760#
L43     007570          1794#
L430    044576          6762    6766#
L431    044616          6770    6773#
L432    045052          6786    6812#
L433    045036          6806#
L434    045042          6806    6809#
L435    045254          6821    6848#
L436    045244          6842    6845#
L437    045344          6862    6864#
L44     007704          1815#
L440    045346          6864    6866#
L441    045434          6872    6879#
L442    045432          6877#
L443    045462          6881    6883#
L444    045534          6890    6898#
L445    045712          6921    6928#
L446    045774          6937    6944#
L447    046154          6973#
L45     007716          1816#
L450    046166          6974#
L451    046630          7042    7056#
L452    046616          7049    7051#
L453    046622          7051    7053#
L454    047426          7307    7311#
L455    047442          7312    7314#
L456    047446          7314    7316#
L457    047474          7321    7323#
L46     007730          1817#
L460    047500          7323    7325#
L461    047512          7326    7328#
L462    047514          7328    7330#
L463    047542          7336    7338#
L464    047546          7338    7340#
L465    047560          7341    7343#
L466    047562          7343    7345#
L467    050016          7393    7397#
L47     007742          1818#
L470    050132          7418    7420#
L471    050140          7420    7422#
L472    050172          7428    7430#
L473    050204          7431    7433#
L474    050216          7434    7436#
L475    050230          7437    7439#
L476    050462          7485    7488#
L477    050512          7490    7493#
L5      004566          1359    1366#
L50     007754          1819#
L500    050522          7494    7496#
L501    051056          7560    7572#
L502    051162          7576    7587#
L503    051124          7577    7580#
```

```
L504    051162          7584    7586#
L505    051342          7589    7612#
L506    051230          7594    7596#
L507    051342          7600    7611#
L51     007766          1820#
L510    051272          7601#
L511    051304          7601    7604#
L512    051342          7605    7610#
L513    051644          7691    7693#
L514    051664          7694    7697#
L515    052374          7796    7828#
L516    052274          7803    7812#
L517    052274          7805    7811#
L52     010000          1821#
L520    052272          7809#
L521    052374          7817    7827#
L522    052362          7818    7822#
L523    052370          7822    7825#
L524    052442          7840    7842#
L525    052450          7842    7844#
L526    052664          7863    7883#
L527    052664          7866    7882#
L53     010012          1822#
L530    052650          7871    7880#
L531    052714          7888    7890#
L532    052772          7898#
L533    053022          7900#
L534    053052          7902#
L535    053074          7905#
L536    053124          7907#
L537    053154          7909#
L54     010030          1828#
L540    053452          7954    7961#
L541    053456          7961    7963#
L542    053536          7969    7976#
L543    053542          7976    7978#
L544    053662          8004#
L545    053766          8027    8030#
L546    054262          8079    8082#
L547    056546          8470    8472#
L55     010050          1829    1832#
L56     010070          1833    1836#
L57     010110          1837    1840#
L6      004646          1367    1375#
L60     010250          1862    1865#
L61     010402          1888    1895#
L62     010402          1889    1894#
L63     010522          1915#
L64     010602          1915    1927#
L65     010554          1921    1923#
L66     010664          1933    1938#
L67     010640          1934#
L7      004640          1370    1372#
L70     010664          1935    1937#
L71     011014          1954#
L72     011072          1962#
```

```
L73     011254              1965    1997#
L74     011222              1969    1992#
L75     011140              1972    1974#
L76     011142              1974    1976#
L77     011202              1985    1987#
MAINT = 177750               273#   1418#   2043#   2445#   3265#   3315#   5975#   5994#   6030#   6310#   6455#   6526#   6597#
                            6961#   7464#   7470#   7530    7969    8120#   8476    9326
MAPHO = 170202               417#   1423#   5690#   5723#   8042    8062#
MAPH36= 170372               477#   1421#   1763#   1770#   1771    1776#   1780#   1781    1782#   8042    8062#
MAPH37= 170376               479#
MAPLO = 170200               416#   1424#   5688#   5722#   5731    8042    8062#
MAPL1 = 170204               418#   5732
MAPL36= 170370               476#   1422#   1764#   1769#   1775#   1779#   1783#   8042    8062#
MAPPER  034510              1431#   1569#   1657#   1706#   2476#   3353#   3371#   3404#   5599#   6043#   6187#   6225#   6293#
                            6369#   6473#   6514#   6540#   6615#   6869#   6900#   6912#   6989#   7015#   7040#   8130#   8166#
MARGIN  001474              1019#   2126#   2194#   2275#   2318#   2443    3264#   3314#   5918    5919#   5922    5923#   5924
                            5926#   5927    5930    5933#   5935    5942    5943#   5944    5946#   5948    5957    5959#   5961#
                            5965#   5971    5972    5973    6256    6257    6284#   6285    6288    6307    6366#   6380    6420#
                            6425    6430#   6431    6434    6453    6472#   6478    6502#   6503    6506    6524    6539#   6567
                            6572#   6573    6576    6595    6614#   6960#   7781
MASTER  043676              2054    6655#
MAST01  044364              6697#   6735#
MAST02  044664              6698#   6781#
MAST03  045070              6699#   6816#
MAST04  045272              6700#   6852#
MEMDON  011322              2004    2017#
MEMERR= 177744               271#   5167    5174#   5181#   5187#   9316    9321    9326
MJPAT   016212              1313    2913#   2921#
MJSIZE  002046              1116#   1930#   1934#   1940#   1941#   1942#   1943#   1953    1958    1994    9317
MJTEST  016156              2457#   2908#
MKCONT  013434              2454#   2462#
MKCSRS  001706              1061#   1439#   1441#   1446#   1916    2467    3123    5272    5295    5412    5445    5511    6631
                            7798    7863    8023    8074
MKCSRT  014676              1303    2660#
MKFLAG  001510              1025#   1724    1752    1787    2449    3322    5833#   5860#   5888    5927    5930    5958    5970
                            7746    7796    7851
MKPAT   016056              1308    2866#   2875#
MKSIZE  002050              1117#   1931#   1936#   1944#   1945#   1946#   1947#   1953    1960    1994    9317
MKTEST  015774              2455#   2856#
MK11AD= 172100               486#   1389    5261    5312    5331
MMR0  = 177572               286#   5240#   5244#   5693#   5695#   5717#   5719#   7036#   8035    8072#   8146    8162#   9316
                            9326
MMR1  = 177574               287#   8035    8072#   8145    8161#   9316    9326
MMR2  = 177576               288#   8035    8072#   8144    8160#   9316    9326
MMR3  = 172516               289#   1417#   1426#   2097#   5684#   5692#   5721#   5725#   5998#   6039#   8035    8072#   8143
                            8159#   9316    9326
MMTRAP  032500              1298    5221#
MMVEC = 000250               267#   1298#   1299#
MOVELO= 000063              6675#   6686
MP1     010202              1853    1857#   1860    1861
MRUNSE  046534              6822#   7039#
MSEEDH  002150              1154#   1275#   1277
MSEEDL  002152              1155#   1276#   1278
MSGA34  070204              2501#   9406#
MSGA87  072253              1904#   1908#   9468#
MSGB34  070221              2502#   9408#
```

```
MSG000  066114          1384    9351#
MSG001  066172          4956    9352#
MSG002  066254          4957    9353#
MSG003  066331          4958    9354#
MSG004  066436          4965    4984    9356#
MSG005  066544          5010    9358#
MSG006  066556          9001    9359#
MSG007  066612          5047    9360#
MSG008  066624          5028    9364#
MSG009  066636          5087    9366#
MSG010  066650          5133    9367#
MSG011  066662          4969    4983    9368#
MSG012  066750          4972    4985    9369#
MSG013  067045          5013    9370#
MSG014  067047          4982    5015    5143    7722    7789    9371#
MSG015  067051          2529*   2530    5037*   5038    5054*   5075*   5076    5092*   5095*   5097*   5100    5114*   5118*
                        5123*   5124    5136*   5140*   5141    7841*   7843*   7845    9372#
MSG016  067053          5110    9373#
MSG017  067065          4968    4971    9374#
MSG018  067076          7684    7787    7836    7945    7958    7973    9375#
MSG019  067101          9376#
MSG020  067105          6082    9377#
MSG021  067141          6101    9378#
MSG022  067633          6637    9393#
MSG023  067655          6149    9394#
MSG024  067677          6152    9395#
MSG025  067721          6135    6163    9396#
MSG026  067752          6096    9397#
MSG027  067764          6145    9398#
MSG028  070001          6159    9399#
MSG029  070015          6171    9400#
MSG030  070035          6260    9401#
MSG031  070054          6172    6210    6486    9402#
MSG032  070114          6194    6231    9403#
MSG033  070132          6198    6235    9404#
MSG034  070150          2503    9405#
MSG035  070265          2074    9410#
MSG036  070270          6209    9411#
MSG037  070307          6238    9412#
MSG038  070326          6247    9413#
MSG039  070344          6240    9414#
MSG040  070366          6259    9415#
MSG041  070433          6268    9416#
MSG042  070460          6272    9417#
MSG043  070501          6277    9418#
MSG044  070526          6282    6428    6500    6570    9419#
MSG045  070553          6289    6435    6507    6577    9420#
MSG046  070606          6300    6393    6446    6517    6588    9421#
MSG047  070641          6419    9422#
MSG048  070660          6085    6090    9423#
MSG049  070720          6296    9424#
MSG050  070752          6386    9425#
MSG051  071060          8118    9427#
MSG052  071100          6390    9428#
MSG053  071131          6392    9429#
MSG054  071147          6407    9430#
```

```
MSG055  071217          6427    9431#
MSG056  071240          6444    6586    9432#
MSG057  071273          7854    7889*   7891    7892*   9433#
MSG058  071277          9434#
MSG059  071277          6480    9435#
MSG060  071327          6515    9436#
MSG061  071371          7761    9437#
MSG062  071400          8630    8689    9438#
MSG063  071420          8631    9439#
MSG064  071431          8632    8691    9440#
MSG065  071441          8690    9441#
MSG066  071453          7606    9442#
MSG067  071536          7647    9443#
MSG068  071545          7748    9444#
MSG069  071552          7750    9445#
MSG070  071557          1957    9446#
MSG071  071602          1959    9447#
MSG072  071615          1961    9448#
MSG073  071630          6569    9449#
MSG074  071646          7795*   7826*   7829    9451#
MSG075  071651          5673    9452#
MSG076  071702          6549    9453#
MSG077  071723          2072    9454#
MSG078  071737          2528    9455#
MSG079  072002          6547    9456#
MSG080  072026          1857    9458#
MSG081  072060          1872    9459#
MSG082  072077          1874    9460#
MSG084  072117          1891    9461#
MSG085  072161          6622    9463#
MSG086  072206          1902    9465#
MSG087  072245          1909    9467#
MSG088  072260          7941    9471#
MSG089  072276          7953    9472#
MSG090  072320          7968    9473#
MSG091  072334          7962    7977    9474#
MSG092  072346          8471    9475#
MSG093  072362          8473    9476#
MSG094  072370          8475    9477#
MSG095  072377          8480    9478#
MSG096  072406          5654    9479#
MSG097  072437          2775    9480#
MTA030  021212          3313#   3341*
MTEST   013334          2128*   2161*   2196*   2236*   2277*   2320*   2366*   2409*   2442#
MTPA03  022212          2995    3484#
MTPA04  022354          3016    3544#
MTPA20  026276          3122    4297#
MTPA21  026724          3197    4429#
MTPA25  030456          4742*   4776*   4783*   4793#
MTPA26  030570          3250    4823#
MTPB03  022254          2999    3509#
MTPB04  022410          3017    3028    3557#
MTPB21  026754          3200    4446#
MTPB25  030476          4778*   4785*   4800#
MTPB26  030604          3255    4830#
MTPC03  022314          3000    3526#
```

```
MTPC21  027010          3204    4463#
MTPC25  030530          4780*   4787*   4789*   4811#
MTPC26  030640          3251    4848#
MTPD03  022334          3001    3535#
MTPD21  027044          3207    4480#
MTPD25  030404          4753*   4755*   4758*   4760*   4776#
MTPD26  030660          3252    4861#
MTPE25  030426          4764*   4766*   4769*   4771*   4783#
MTP000  022102          1557    2962    3267    3447#   3449    4940    6963
MTP001  022126          2971    3458#
MTP002  022160          2980    3470#
MTP005  022430          3027    3566#
MTP006  022464          3035    3580#
MTP007  022654          3045    3623#
MTP010  022754          3053    3662#
MTP011  023062          3062    3698#
MTP012  023472          3071    3808#
MTP013  024050          3079    3886#
MTP014  024422          3089    3953#
MTP015  025016          3099    4035#
MTP016  025404          3108    4110#
MTP017  026006          3113    4188#
MTP020  026064          4221#
MTP022  027074          3214    4498#
MTP023  027542          3220    4589#
MTP024  027750          3229    4656#
MTP025  030144          3237    4734#
MTP030  030676          3317    4420    4868#
MTP031  030706          3355    4874#   4896
MTP20A  026064          3149    4226#   4230
MTP20B  026102          3154    4234#   4243
MTP20C  026134          3160    4247#   4256
MTP20D  026166          3167    4260#   4269
MTP20E  026220          3173    4273#   4282
MTP20F  026252          3179    4286#   4293
MTV020  017720          3124*   3125    3138*   3140#
MT0000  016312          2957#   6329
MT0001  016346          2877    2921    2965#   6330
MT0002  016406          2878    2922    2974#   6331
MT0003  016450          2923    2984#   6332
MT0004  016616          2879    2924    3009#   6333
MT0005  016700          2880    2925    3020#   6334
MT0006  016760          2660    3031#   6335
MT0007  017014          2876    3038#   6336
MT0010  017056          2661    3049#   6337
MT0011  017112          2663    3056#   6338
MT0012  017170          2664    3065#   6339
MT0013  017246          2665    3074#   6340
MT0014  017322          2666    3083#   6341
MT0015  017400          2667    3093#   6342
MT0016  017456          2668    3102#   6343
MT0017  017534          2875    3111#   6344
MT0020  017556          2882    3116#   6345
MT0021  020212          2881    3187#   6346
MT0022  020334          2883    3212#   6347    6605*
MT0023  020356          2885    3217#   6348
```

```
MT0024  020404          2886    3224#   6349
MT0025  020442          2662    3232#   6350
MT0026  020510          2884    2926    3240#   6351
MT0027  020640          2020*   3262#   6352    6388*
MT0030  021206          2057*   3311#   5996*   6033*   6353
MT0031  021430          2887    2927    3350#   6354    6463*
MT020X  020204          3158    3164    3171    3177    3164#
MT020Y  020072          3152    3166#
MT020Z  017722          3135*   3142#
MT0999  021532          1322    2669    2670    2671    2672    2673    2674    2675    2676    2677    2678    2679    2680
                        2681    2682    2683    2684    2685    2686    2687    2688    2689    2690    2691    2888    2889
                        2890    2891    2892    2893    2894    2895    2896    2897    2898    2899    2900    2901    2902
                        2903    2904    2905    2906    2928    2929    2930    2931    2932    2933    2934    2935    2936
                        2937    2938    2939    2940    2941    2942    2943    2944    2945    2946    2947    2948    2949
                        2950    2951    2952    3364#   6355    6356    6357    6358    6359    6360
MULTIP= ****** U        1126    1129    1182    1184    1186    1199    1228    1242    1250    .257    1825    1912    2046
                        2055    3958    3961    5022    5846    5854    6650    6651    7061    9265    9267    9277    9279
                        9305    9307    9330    9333    9361    9457    9462    9464    9470
MUT     001500          1021#   2019*   2022*   2447*   2458*   6313*   6458*   6600*   7687    7918    7920*   7932    7934*
                        7936*
NEMCNT  001456          1012#   1525*   1538    1568*   1588    1600    1618    1640    5206*   5207    5211*   9315
NEWBAN  001756          1084#   3339    5679*   5685    5712*   5797    5809
NEWKER  036020          5694*   5718*   5807#
NOECCF  002056          1120#   2531*   2533*   3057    3066    3084    3094    3103
NOERRO  002112          1136#   6131*   6146*   6160*   7560    7576    7589    7641    7653    7694    7695*
NOFSMO  002110          1135#   3270*   3308*   3319*   3336*   3347*   6089
NONEM   001466          1016#   1526*   1555*   1626*   1635*   5204
NONEXI  032422          1527    1556    5169    5204#
NOPAR   001464          1015#   1524*   1554*   1586*   1736*   1740*   1766*   1778*   2026*   2038*   2484*   2862*   2868*
                        2909*   2914*   3080*   3121*   3316*   3352*   4739*   4773*   5165    5176    5184    5189    6169
                        6170*   6202*   6207    6208*   6251*   6321*   6326*   6425    6472*   6478    6511*   6539*   6567
                        6614*
NOSCOP  002124          1141#   3188*   3209*   7494    7594
NOTAB   002044          1115#   5077    7718*   7720*
NULLFL  002014          1100#   3366*   5970    5982*
OLDCAC  001746          1080#   5286*   5306    5320*   5324    5338*   5341
OLDMAR  001754          1083#   5918*   5922*   5942*   5957*   5971    5972    6257*
ONES    002154          1156#   1249    1606    1670    1671    2564    2566    3012    4504    4506    4918    4924    5977
                        6721    6722    6872    6881    7263    7898    7900    7902    7905    7907    7909
O.3DR1  072606          9804    9809    9813    9821*   9828*   9946    9960*   9976    10012   10013*  10047*  10294#
O.BACK  074234          9694    9789#
O.BD    076342          9953    9954    10274#
O.BIAS  076344          9680    10275#
O.BKP = 000016          9497#   9809    9813    9819    9826    9869    9908    9922    9945    9959    10011   10050   10295
                        10297   10299
O.BKPT  074256          9636    9797#
O.BK1   075012          9523    9923#
O.BRK   074774          9545    9920#
O.BW    076240          9617    9647*   9720*   9723*   9734    9743    9763    9770    9773    9774*   9777*   9789    9792
                        9838    9856*   9858*   9974*   10059   10173   10201#
O.BYT   073724          9687    9723#
O.CAD   076232          9603*   9619    9733*   9736    9739    9741    9747    9768*   9770*   9771    9775    9792*   9844
                        10176*  10178*  10198#
O.CADV  075566          9742    9848    9857    10057#  10196
O.CLGT= 000024          9672    10254#
O.CLSE  076146          9616    9755    9765    9791    10171#
```

```
O.CMFD  076250          9650*  9678   9712*  9726   10216#
O.CR    076272         10180  10182  10183  10230#
O.CRET  074072          9688   9755#
O.CRLF  076200          9772   9952   10030  10180#
O.CRLS  076206          9648  10182#
O.CSR1  076246         10021* 10037  10214#
O.CSR2  076247         10022* 10038  10215#
O.CT    072630          9822* 9830*  9904*  9962*  9964*  10296#
O.CTLC  073114          9552#  9702
O.CTLE  073174          9571#  9704
O.CTLF  073142          9561#  9703
O.C1    074732          9908#  9963
O.DCD   073452          9631   9633   9647*  9756   9978
O.DCDA  074076          9756#  9816   9823   9827
O.DCD1  073502          9639   9654#  9713
O.DCD3  073462          9649#  9744   9751   9859
O.DOT   076234          9732*  9768   9771*  10199#
O.ERR   073442          9622   9645#  9673   9728   9758   9861
O.ERR1  074524          9802   9810   9814   9820   9839   9843   9861#  9866   9868   9871   9896   9899   9901
O.ERR2  074116          9764#  9790
O.ERR3  074102          9758#  9764
O.FTYP  075662          9646   9746   9748   9750   9782   9841   9967   9971   9973   10072  10077  10082#  10120
                       10139  10145
O.GET   075740          9582   9656   10101#  10121  10125
O.GO    074530          9690   9865#
O.GOGO  074556          9575   9873#
O.GO2   074630          9880   9883#  9916
O.ID    076253          9530  10223#
O.IDND= 076270          9531  10227#
O.LG  = 000020          9586  10272#
O.LGCH  076275          9669  10234#  10254
O.MIN   073434          9638#  9701
O.MINS  076271          9638*  9651*  9675   9681*  10228#
O.ODT   072726          8407   9514#  9566   10279
O.OFST  074420          9695   9838#
O.OLD   074106          9693   9762#
O.OP1   074112          9691   9763#
O.OP2   074152          9771#  9793
O.OP2A  074160          9604   9772#
O.ORAB  073316          9605#  9698
O.ORPC  073274          9599#  9692
O.ORRB  073326          9608#  9699
O.P     076245          9521*  9543*  9869*  9857   9908   9922*  9961*  9968   9975   10211#
O.PRI   072604          9536   9927   9931   10290#
O.PROC  074664          9697   9895#
O.PRI   074716          9903   9905#
O.RALL  074362          9546   9818   9824#
O.RCSR= 177560          9505#  10021  10023*  10033  10035  10037*  10101
O.RDB = 177562          9504#  10103
O.REGT  073222          9582#  9689
O.REM   075534          9535   9926   10044#
O.RORA  076224          9776   9977   10196#
O.RRST  076124          9516  10152#
O.RSB   075402          9882  10011#
O.RSR   075350          9883   9998#
O.RSTT  075466          9876   9907   10030#
```

```
0.S     076243  9522* 9542* 9630* 9632* 9879  9910  9941  10044 10205#
0.SCAN  073506  9593  9656# 9667
0.SCRN  076252  10111 10122 10221#
0.SEMI  073674  9685  9709#
0.SEQ   076242  9762* 9766  9769* 10204#
0.SMFD  076251  9626  9649* 9711* 9729  9865  9895  10218#
0.SNGL  073400  9626# 9700
0.STM = 000340  9499# 9544  9874  9905  9912
0.SVR   075312  9528  9534  9923  9984#
0.SVTT  075432  9526  9951  9965  10021#
0.T     076244  9877* 9914* 9924  10210#
0.TBIT  074576  9877# 9909  9911  9925
0.TBT = 000020  9500# 9878  9881  9888  9915  9937
0.TCL2  073374  9622# 9627
0.TCSR= 177564  9507# 10022 10024* 10025 10031 10038* 10082
0.TDB = 177566  9506# 10096*
0.TL    076321  9583  9586  9594  10256# 10272
0.TRTC  076346  9797  9825  10013 10276# 10300
0.TVEC= 000014  9498# 9517  9544* 9545*
0.TYPE  076060  9532  9555  9564  9573  9955  10136# 10141 10148 10184
0.TYP1  075736  10097# 10137
0.UIN   072652  9829* 10012* 10047 10298#
0.UPC   072600  9520* 9873* 9885  9920* 9940  9944* 9957* 10288#
0.URD   072562  9529  9591  10281#
0.USP   072576  9552  9561  9985* 9986  10005 10287#
0.UST   072602  9878* 9881* 9884  9887  9915* 9921* 9929  10152* 10289#
0.WRD   073712  9686  9720#
0.WRD1  073770  9731  9734# 9783
0.WST   076134  9518  9541  9875  9889  9906  9913  9939  10162#
0.XXX   076236  9984* 9992  9999* 10006 10200#
PADDRE  001424  999#  5173* 9314
PAFBAF  012500  2009  2263# 2292* 2300*
PAFBAW  012636  2010  2304# 2335* 2340* 2348*
PARBAF  013016  2011  2352# 2381* 2389*
PARBAW  013154  2012  2393# 2424* 2429* 2437*
PARCNT  001460  1013# 1523* 1535  1567* 1598  1616  1637  1741* 2031* 2033  5170* 5171  9314
PARITY  032234  1292  5165#
PARTHE  001750  1081# 4741* 4800* 4811* 4814* 5186  6169  6185* 6202* 6207  6223* 6251* 6512*
PARVEC= 000114  265#  1292* 1293*
PASFLG  001734  1073# 2448* 2487* 3312* 3330  3331* 3725  3796  3798* 3898  3941  3943* 3970
                4023  4025* 4046  4098  4100* 4123  4176  4178* 4597* 4619  4634* 4636  4641
                6314*
PATERR  001462  1014# 1559  1566* 1596  1614
PATPLU  004276  1304* 1306* 1309* 1311* 1314* 1316* 1318#
PATTER  001502  1022# 2124* 2166* 2192* 2241* 2264* 2305* 2374* 2417* 2864  2911  5896* 5897
                5902* 6256  6274* 6275  6276  6323  6368* 6380  6420* 6425  6472* 6567  6614*
PCBUMP  001764  1087# 2863* 2910* 2986* 3011* 3022* 3033* 3051* 3226* 3242* 5180  6256  6322*
                6366*
PCONFI  031162  1962  4947# 6374*
PCONFS  031442  4949* 4988  4994#
PCONF1  031352  4960  4978#
PCONF2  031410  4950  4976  4988#
PDP110  032512  1294  5224#
PDP70M  002226  1183# 1249  1254* 1828  1863* 1893* 2054
PDRONE  002156  1157#
PERAOS  047004  7227# 7234
```

```
PERBNK  050020          7320*  7335*  7350*  7378*  7400#
PERECC  050104          7416#  7427*  7443*
PERRAB  047516          7249   7254   7333#
PERRAW  047450          7206   7212   7218   7224   7239   7244   7259   7264   7269   7274   7279   7284   7294
                        7299   7304   7317   7318#  7398
PERRA3  047632          7356   7360   7362#
PERRA7  047564          7289   7348#
PERR01= 104427           110#  3466
PERR02= 104430           111#  3454   3479
PERR03= 104431           112#  3530   3538
PERR04= 104432           113#  3553   3575
PERR05  047000          7226#
PERR06  047026          7233#
PERR07= 104433           114#  3590   3609
PERR10= 104434           115#  3595   3614
PERR11= 104435           116#  3631   3680   4325   4368   4410
PERR12= 104436           117#  3637   3687
PERR13= 104437           118#  3645
PERR14= 104440           119#  3651
PERR15= 104441           120#
PERR16= 104442           121#
PERR17= 104443           122#  4434   4456   4467   4490   4542   4546
PERR20= 104444           123#  4440   4450   4473   4484   4551   4555   4629
PERR21= 104445           124#  4347   4392
PERR22= 104446           125#  4210
PERR23= 104447           126#  4693   4706
PERR24= 104450           127#  4701
PERR25= 104451           128#  4836   4841
PERR26= 104452           129#  4239   4264   4290
PERR27= 104453           130#  4251   4278
PERR30= 104454           131#  4881
PERR31= 104455           132#  3752   3758   4001   4007   4079   4084   4154   4160
PERR32= 104456           133#  3764   3778
PERR33= 104457           134#  3848
PERR34= 104460           135#
PERR35= 104461           136#
PERR36= 104462           137#
PERR37= 104463           138#
PERR40= 104464           139#
PERR41= 104465           140#
PERR42= 104466           141#
PERR43= 104467           142#
PERXOR  047730          7319*  7334*  7349*  7383#  7423*  7452*
PFLAG   001512          1026#  5836*  5864*  6678   6743   6786   6821
PHYADD  001426          1000#  7767*  7768*  7769*  7770*  7772*  7773
PORTDI  003554          1238#  1847*  1889   1903   1910   6877   9332
PROTEC= 000004          1749#  1756
PSW    = 177776           171#  1401*  1402*  1407*  1579*  1592*  1609*  1630*  1659*  2540*  2834*  3354*  3393*
                        3426*  4953*  5352*  5354*  5681*  5714*  5765*  5781*  6044*  6188*  6226*  6243*  6304*
                        6397*  6450*  6521*  6528*  6592*  6876*  6884*  6901*  6908*  6917*  6933*  6966*  6979*
                        6991*  6997*  7017*  7046*  7057*  7308*  7372*  7394*  7949*  7950*  7965*  8014*  8018*
                        8096*  8100*  8131*  8167*
PTYBUF  003604          1241#  6803   6804   6826*  6840   6841   6982*  6986   6992*  7000*  7018*  7025*  9331
PTYFLA  003574          1240#
PWRVEC= 000024           258#  1290*  1291*  1364*  7991*  7992*  8048*  8053*  8117*  8132   8133   8135*  8136*
                        8168*  8169*
```

```
QUICK   002122          1140#  1363*  6032
QVFLAG  002016          1101#  1361*  1368*  2073   2075*  3151   3791   3873   3936   4018   4093   4171   5915
                        5954
RANODD  030620          3254*  4838#  4842*
RDCHR = 104411            89#  8526
RDDEC = 104414            92#  1858   6097
RDLIN = 104412            90#  8601   8652
RDOCT = 104413            91#  6150   6153   6173   6211   6241   6261   6273   6278   6283   6429   6487   6501
                        6571   6623   6638
READCS= 104426           108#  1440   2526   2600   2633   2840   4319   4340   4362   4385   4404   4750   4805
                        4816   5418   5437   5451   5470   6130   6144   6158   7365   7804   7867   8028
READON  002066          1124#  6936*  6945*  6974
REALPA  001744          1079#  2958*  2966*  2975*  2985*  3010*  3021*  3032*  3039*  3050*  3061*  3070*  3078*
                        3088*  3098*  3107*  3112*  3120*  3189*  3213*  3218*  3225*  3236*  3241*  3263*  3313*
                        3351*  3365*  6969*  6983*  7002*  7428   7431   7434   7437   7526   7734   8481
REFDEL= 000400          4499#  4578
REFRES  027442          4530*  4566#
REFSUB  027512          4569*  4573*  4578#
REGCOP  030764          2961*  2970*  2979*  4901#  6972*
RELOCA  034720          2142*  2175*  2215*  2255*  2297*  2345*  2386*  2434*  3333*  5639#
RELOC1  035212          5679#  6911*
RESREG= 104416            95#  3002   3256   4943   5748   5785   6086   6091   6123   7690   8784
RESTAR  002052           976*   978*  1118#  1329
RESVEC= 000010           253#  1294*  1295*
RLFLAG  001516          1028#  2139   2172   2212   2252   2294   2342   2383   2431   5696*  5720*  5867   5995
                        6031   6084   6913*  7536   8004   8470
RRFLAG  001514          1027#  2122   2155   2190   2230   2273   2316   2362   2405   2475   3274   3292   3323
                        5644   5837*  5845*  5865*  5869*  6295   6410   6462   6604
RSX11M= ****** U        21:2
RUNCOU= 000104          6836#  6844
SAMESA= 000010          1883#  1892
SAVREG= 104415            94#  2996   3249   4937   5746   5749   6081   7626   8769
SBEFLA  001412           993#  2048   2067*  4300*
SBEMSK  001722          1070#  3720*  3721*  3736   3738   3786   3788*  3814*  3815*  3820   3822   3831   3868
                        3870*  3892*  3893*  3903   3905   3909   3911   3937   3939*  3964*  3965*  3975   3977
                        3981   3983   4019   4021*  4040*  4041*  4051   4053   4057   4059   4094   4096*  4117*
                        4118*  4127   4129   4133   4135   4172   4174*
SBETES  014130          2481*  2537#
SCOPE = 000004           170#  1434   1522   1552   1646   1747   1999   2017   2024   3399   3432
SDPAR0= 172260           363#  3001   3252   3533
SDPAR7= 172276           370#
SDPDR7= 172236           350#
SEEDHI  002142          1151#  1277*  3244   3259*  7044   8743   8750   8755*
SEEDLO  002144          1152#  1278*  3243   3258*  8742   8748   8754*
SETMAR  036616          2159*  2234*  2364*  2407*  5951#
SETPAT  036366          2157*  2232*  5900#
SHUTUP  037156          2084*  6013*  6021#
SIDE    002000          1094#  2469*  2502   2505*  2711   2727   2743   2759   2787   2803   2819   2851   2852
                        3128*
SIDEOK  015640          2478   2828#
SIPAR0= 172240           353#  5601
SIPAR3= 172246           356#  5615
SIPDR0= 172200           333#  5603   8008   8089
SIZE  = 040000           491#  1529   1575   1604   1628   2960   2968   2977   2991   3015   3026   3248   3278
                        3297   3324   4422   4939   5682   5715   5766   5782   6414   6971
SIZELO= 177760           277#  1797   1799   1801   1805
```

```
SLAVES  002072  1127# 1859* 1860 1861 1862 1877 1884 1911 5853 6690 6696 6761 6774
                6778 6791 6798 6810 6828 6835 6846 6859
SLAVUP  044346  6658 6729#
SLPAT   046050  6949 6951 6953 6955 6957 6959#
SLPAT0  046010  6922 6938 6948#
SLPAT1  046014  6923 6939 6950#
SLPAT2  046022  6924 6940 6952#
SLPAT3  046030  6925 6941 6954#
SLPAT4  046036  6926 6942 6956#
SLPAT5  046044  6927 6943 6958#
SOBK    002136  1149# 3357
SOBLEN= 000056  3355 3360 4896#
SOURCE  001760  1085# 3731* 3818* 3907* 3979* 4055* 4131* 4737* 5536
SPECIA  002010  1098# 2478 2599* 2705* 2721* 2737* 2753* 2781* 2797* 2813* 3130
ST    = 177776  9495# 10152 10162*
STACK = 001400  165# 166 167 168 969 980 1248 1255 1265 .367 2080 2096 6729
START   003614  977 979 1246# 6018 10307
START1  000300  973 976#
START2  000310  974 978#
STOPOK  002102  1132# 1745* 7490 7491*
STRIPE  002036  1112# 4509* 4513 4533 4563*
STRTDI  001740  1077# 4607* 4635
SUBAAA  004336  1317 1327#
SUBAAB  004456  1344 1351#
SUBAAC  004646  1365 1378#
SUBAAD  005040  1412 1414 1417#
SUBAAP  005052  1420#
SUBAAQ  006510  1587 1645#
SUBAAR  010506  1828 1864 1914#
SUBAAS  005606  1502 1521#
SUBAAT  010014  1798 1811 1826#
SUBAAU  011474  2039 2048#
SUCCES  002002  1095# 2472* 2492* 2500 7858* 7881 7888 7896* 7898* 7900* 7902* 7905* 7907*
                7909#
SUPD0A  001732  1072# 2997* 3035* 3045* 3053* 3062* 3071* 3079* 3089* 3099* 3108* 3113* 3122*
                3214# 3220* 3229* 3237* 3269* 3361* 3429* 6965* 6984#
SUPD01  021546  2032* 2963* 2972* 2981* 3150* 3198* 3253* 3257* 3281* 3300* 3326* 3370# 4942*
                6973 7011#
SUPD02  021562  3003* 3018* 3029* 3155* 3161* 3168* 3174* 3180* 3201* 3205* 3208* 3372#
SUPD03  021724  2998* 3036* 3046* 3054* 3063* 3072* 3081* 3090* 3100* 3109* 3114* 3215* 3221*
                3230* 3238* 3284* 3303* 3404# 6416* 6974 6987*
SUPD04  021740  3184* 3362* 3405#
SUPDR0  001644  1043# 3374* 3382 3407* 3415
SUPDR1  001646  1044# 3375 3408
SUPDR2  001650  1045#
SUPDR3  001652  1046#
SUPDR4  001654  1047#
SUPDR5  001656  1048#
SUPDR6  001660  1049# 3385 3418
SUPSTK= 000700  167# 1403 3394 3427 7954 7960
SWAPAT  002176  1167# 1280* 4704 4722*
SWR     002200  1170# 1341* 1343 1349* 1357* 1413 1546 1709 1962 2001 2073 2857 3151
                3156 3162 3169 3175 3790 3872 3935 4017 4092 4170 4960 5640 5672
                5915 5954 6288 6316 6381 6434 6439 6481 6506 6529 6576 6581 7490
                7497 7512 7567 7582 7584 7590 7594 7691 8044 8060* 8408 8416 8441*
SWREG = 000176  483# 1349 1413 8408
```

```
SW0    = 000001          231#   1546   1709   2857   3156   3162   3169   3175   6316   6381   6439   6481   6581
SW1    = 000002          230#
SW10   = 002000          221#   7567
SW11   = 004000          220#   2073   3151   3790   3872   3935   4017   4092   4170   5915   5954   6288   6434
                        6506    6576
SW12   = 010000          219#   5640
SW13   = 020000          218#   5672   7582
SW14   = 040000          217#   7497
SW15   = 100000          216#
SW2    = 000004          229#
SW3    = 000010          228#
SW4    = 000020          227#   4960
SW5    = 000040          226#   7584
SW6    = 000100          225#   1962
SW7    = 000200          224#   7691
SW8    = 000400          223#   7490
SW9    = 001000          222#   7512   7594
SYSSIZ   007670         1803*   1807*  1814#
SYSTID=  177764          281#
TAG2$    006172         1562    1584#
TAG3$    006216         1583    1588#
TAG4$    021642         3383    3385#
TAG70$   051666         7667    7703#
TAG71$   051676         7668    7709#
TAG72$   051706         7669    7715#
TAG73$   051752         7670    7728#
TAG74$   051760         7671    7734#
TAG75$   051772         7672    7740#
TAG76$   052004         7673    7746#
TAG77$   052026         7674    7756#
TAG78$   052030         7675    7761#
TAG79$   052036         7676    7767#
TAG80$   052116         7677    7781#
TAG81$   052130         7678    7787#
TAG82$   052152         7679    7794#
TAG83$   052406         7680    7835#
TAG9$    006036         1560#   1590   1622   1643
TASK     003564         1239#   1852*  1875   6681*  6682   6748*  6754   6760*  6771   6790*  6796   6827*  6833
                        6856*   6857   6885   6887   6902*  6918   6934
TASKDO   045346         1855    6733   6751*  6764*  6792*  6829*  6863   6868#
TASKRE   045330         6862#   6904
TASKO0   045600         6891    6906#
TASKO1   045602         6892    6907#
TASKO2   045644         6893    6916#
TASKO3   045726         6894    6932#
TASKO4   046172         6895    6978#
TASKO5   046302         6896    6996#
TASKO6   046516         6897    7034#
TBG4$    022020         3416    3418#
TCFIG1   031572         5049#   5079   7719*
TCONFI   031444         4966*   4975*  4986*  5010#
TEMP     002120         1139#   1966*  1983   1984#
TESTAD   002070         1125#   2480*  2495*  2539   3034   3052   3740   3801   3824   3863   3879   3889   3946
                        3989    3995   4028   4065   4071   4075   4103   4141   4150   4181   4740   7418
TIME     002012         1099#   6389*  6391
TIMEOU   032466         1256    1296   1348   1397   1500   1543   1585   1777   5217#  6133   6137   6148   6165
```

```
TKVEC = 000060   6203  6252
                  261# 4948  4950* 4951* 4991* 6256  6301* 6302* 6364* 6380  6394* 6395* 6405*
                 6425  6447* 6448* 6472* 6478  6518* 6519* 6539* 6567  6589* 6590* 6614*
TMFLAG  001522   1030# 2288* 2331* 2377* 2420* 5886
TOOMAN  002054   1119# 7412* 7584  7613*
TRAPVE= 000034    260# 1288* 1289*
TRT   = 000003   9501# 9829  10276
TSKOSA  046510   7001  7029# 7030
TSTBAN  006404   1571  1625#
TSTDAT  001716   1069# 3723* 3724* 3727* 3728* 3729  3730  3731  3737* 3739* 3742  3744  3750
                 3756  3816* 3817* 3818  3821* 3823* 3827  3829  3896* 3897* 3900* 3901* 3907
                 3910* 3912* 3914* 3916* 3918  3920  3968* 3969* 3972* 3973* 3979  3982* 3984*
                 3986* 3988* 3991  3993  3999  4005  4044* 4045* 4048* 4049* 4055  4058* 4060*
                 4062* 4064* 4067  4069  4077  4082  4121* 4122* 4125* 4126* 4131  4134* 4136*
                 4138* 4140* 4143  4145  4152  4158  4735* 4736* 4737  4752* 4754* 4756* 4757*
                 4759* 4762* 4763* 4765* 4767* 4768* 4770* 4794  4796  .268  7273  7419  7421
                 9325
TSTRD1  033262   5330  5351#
TSTREA= 104510    160# 2558  2577  3766  3780  3836  3850
TST1    005130   1434#
TST2    005610   1522#
TST3    005764   1552#
TST4    007256   1747#
TST5    011254   1999#
TST6    011322   2017#
TST7    011356   2024#
TYPDS = 104405    83# 1956  1958  1960  2077  6391  6546  6558  7709
TYPEIT= 104401    79# 1384  1857  1872  1874  1891  1902  1909  1955  1957  1959  1961  2072
                 2074  2503  2528  2530  2775  4956  4957  4958  4965  4967  4968  4969  4970
                 4971  4972  4982  4983  4984  4985  5010  5013  5015  5028  5038  5047  5076
                 5087  5100  5110  5124  5133  5141  5143  5654  5673  6082  6085  6090  6096
                 6101  6135  6145  6149  6152  6159  6163  6171  6172  6194  6198  6209  6210
                 6231  6235  6238  6240  6247  6259  6260  6268  6272  6277  6282  6289  6296
                 6300  6386  6390  6392  6393  6407  6419  6427  6428  6435  6444  6446  6466
                 6480  6486  6500  6507  6515  6517  6547  6549  6559  6569  6570  6577  6586
                 6588  6608  6622  6637  7569  7606  7627  7647  7648  7650  7657  7659  7684
                 7722  7748  7750  7761  7776  7787  7789  7829  7836  7845  7854  7891  7941
                 7943  7945  7953  7956  7958  7962  7968  7971  7973  7977  8118  8224  8311
                 8376  8414  8415  8417  8426  8434  8443  8464  8471  8473  8475  8480  8482
                 8530  8539  8545  8550  8554  8559  8560  8562  8565  8569  8628  8630  8631
                 8632  8687  8689  8690  8691  9001
TYPOC = 104402    80# 7632  7703  7944  7946  7957  7959  7972  7974  8416
TYPOS = 104403    81# 1873  1903  1910  6191  6239  6248  6555  7734  7740  7781  7788  8474
                 8479  8481
TYPSO = 000000  1249# 1319# 1325  1329# 1334  1353# 1355  1356# 1358  1359# 1366  1367# 1375
                 1412# 1446# 1447# 1456  1492# 1495  1546# 1548# 1550  1637# 1639  1640# 1642
                 1651# 1705  1709# 1711# 1713  1718# 1743  1749# 1759  1787# 1790  1794# 1815#
                 1816# 1817# 1818# 1819# 1820# 1821# 1822# 1828# 1829# 1832  1833# 1836  1837#
                 1840  1853# 1860# 1861# 1862# 1865  1866# 1877  1883# 1898  1905# 1911  1915#
                 1927  1932# 1939  1954# 1962# 1965# 1997  2028# 2037  2048# 2052  2054# 2073#
                 2076  2102# 2110  2143# 2176# 2216# 2256# 2298# 2346# 2387# 2435# 2466# 2510
                 2527# 2532# 2534  2559# 2578# 2604# 2621  2641# 2642# 2648# 2649# 2655# 2692
                 2711# 2713# 2715  2727# 2729# 2731  2743# 2745# 2747  2759# 2761# 2763  2773#
                 2776  2787# 2789# 2791  2803# 2805# 2807  2819# 2821# 2823  2851# 2852# 2857#
                 2859# 2861  3057# 3058# 3060  3066# 3067# 3069  3075# 3077  3084# 3085# 3087
                 3094# 3095# 3097  3103# 3104# 3106  3117# 3119  3124# 3138  3151# 3153  3156#
```

```
                    3159    3162#   3165    3169#   3172    3175#   3178    3233#   3235    3271#   3288    3289#   3307
                    3320#   3329    3330#   3345    3790#   3791#   3872#   3873#   3935#   3936#   4017#   4018#   4092#
                    4093#   4170#   4171#   4501#   4564    4568#   4570    4572#   4575    4960#   5077#   5091#   5093#
                    5099    5267#   5281    5290#   5304    5344#   5346#   5348    5362#   5368#   5384#   5390#   5397#
                    5404#   5414#   5426    5429#   5431#   5433    5447#   5459    5462#   5464#   5466    5497#   5504#
                    5512#   5520    5640#   5641#   5655    5656#   5676    5687#   5689    5848#   5853    5915#   5917
                    5927#   5929    5930#   5932    5954#   5956    5958#   5960#   5962    5970#   5981    5995#   6014#
                    6016#   6019    6031#   6032#   6034    6084#   6088    6089#   6093    6103#   6117    6181#   6183#
                    6184#   6219#   6221#   6222#   6263#   6267#   6270    6275#   6276#   6280    6285#   6287    6288#
                    6291    6295#   6298    6316#   6318#   6320    6381#   6383#   6385    6408#   6418    6431#   6433
                    6434#   6437    6439#   6441#   6443    6460#   6465    6481#   6483#   6485    6495#   6497#   6498#
                    6503#   6505    6506#   6509    6548#   6562    6573#   6575    6576#   6579    6581#   6583#   6585
                    6602#   6607    6660#   6669    6670#   6690    6693#   6696    6701#   6705    6736#   6778    6782#
                    6813    6817#   6849    6853#   6859    6862#   6864#   6866    6871#   6880    6881#   6883    6890#
                    6898    6921#   6928    6937#   6944    6973#   6974#   7042#   7056    7307#   7311    7312#   7314#
                    7316    7321#   7323#   7325    7326#   7328#   7330    7336#   7338#   7340    7341#   7343#   7345
                    7393#   7397    7418#   7420#   7422    7428#   7430    7431#   7433    7434#   7436    7437#   7439
                    7485#   7488    7490#   7493    7494#   7496    7497#   7560#   7572    7576#   7587    7589#   7612
                    7691#   7693    7694#   7697    7796#   7828    7838#   7846    7861#   7887    7888#   7890    7898#
                    7900#   7902#   7905#   7907#   7909#   7942#   7947    7954#   7961#   7963    7969#   7976#   7978
                    8004#   8024#   8033    8076#   8085    8470#   8472
TYPS1 = 000002       1321#   1323    1370#   1372#   1374    1546#   1655#   1704    1709#   1720#   1731    1735#   1742
                    1750#   1758    1787#   1790    1886#   1897    1915#   1919#   1924    1933#   1938    1968#   1993
                    1994#   1996    2030#   2036    2049#   2051    2073#   2467#   2506    2857#   3058#   3059#   3067#
                    3068#   3075#   3076#   3085#   3086#   3095#   3096#   3104#   3105#   3117#   3118#   3127#   3137
                    3151#   3156#   3158#   3162#   3164#   3169#   3171#   3175#   3177#   3233#   3234#   3272#   3287
                    3290#   3306    3322#   3328    3334#   3338    4502#   4505#   4508    4509#   4563    4572#   4575
                    5094#   5096#   5098    5275#   5277#   5279    5298#   5300#   5302    5415#   5425    5448#   5458
                    5513#   5519    5642#   5653    5657#   5671    5672#   5674    5849#   5851    5954#   5971#   5980
                    6014#   6285#   6410#   6417    6431#   6462#   6464    6503#   6550#   6561    6573#   6604#   6606
                    6675#   6689    6739#   6777    6786#   6812    6821#   6848    6872#   6879    7042#   7056    7577#
                    7590    7584#   7586    7594#   7596    7600#   7611    7800#   7815    7817#   7827    7840#   7842#
                    7844    7862#   7886    7955#   7960    7970#   7975    8025#   8032    8077#   8084
TYPS10= 000000       2482#   2494
TYPS11= 000002       2485#   2490
TYPS2 = 000000       1656#   1703    1721#   1728    1752#   1757    1887#   1896    1921#   1923    1934#   1935#   1937
                    1969#   1992    1994#   2033#   2035    2463#   2505    3130#   3132    3274#   3286    3292#   3305
                    3323#   3327    4515#   4527    4536#   4560    5417#   5423    5424#   5450#   5456    5457#   5515#
                    5517    5518#   5644#   5652    5658#   5670    5972#   5979    6554#   6560    6676#   6688    6743#
                    6776    6787#   6791    6793#   6798    6799#   6811    6823#   6828    6830#   6835    6836#   6847
                    6877#   7047#   7054    7601#   7604    7605#   7610    7801#   7814    7818#   7822#   7825    7863#
                    7883    8027#   8030    8031#   8079#   8082    8083#
TYPS3 = 000000       1667#   1693    1694#   1698    1723#   1727    1753#   1755    1888#   1895    1972#   1974#   1976
                    1985#   1987    1988#   1990    2471#   2499    2500#   2504    3280#   3282    3283#   3285    3299#
                    3301    3302#   3304    4515#   4527    4536#   4560    5419#   5422    5452#   5455    5659#   5669
                    5972#   6678#   6687    6744#   6775    6800#   6810    6837#   6846    7049#   7051#   7053    7601#
                    7803#   7812    7813#   7866#   7882
TYPS4 = 000000       1675#   1692    1724#   1726    1889#   1894    2473#   2498    4516#   4517#   4518#   4521#   4524
                    4537#   4538#   4539#   4548#   4557    5660#   5668    6749#   6753    6757#   6761    6762#   6766
                    6767#   6774    6806#   6809    6842#   6845    7805#   7811    7871#   7880    7881#
TYPS5 = 000000       1683#   1691    1724#   2475#   2497    4541#   4543    4545#   4547    4550#   4552    4554#   4556
                    5661#   5662#   5667    6760#   6770#   6773    6806#   7809#
TYPS6 = 000000       2478#   2479#   2496    5663#   5666
TYPS7 = 000002       2480#   2495
T12A    025404       4115#   4179
T12B    025426       4119#   4175
```

```
UDPAR7= 177676  330#
UDPDR7= 177636  309#  2995
UIPAR0= 177640  312  313#  5788
UIPAR2= 177644  315#  1572*  1607*  3268*  3542  4866  4941*  6964*
UIPAR3= 177646  316#  1593*  3269  6965
UIPAR4= 177650  317#  5752  5769
UIPAR6= 177654  319#  1558*  3564
UIPAR7= 177656  320#  1559*
UIPDR0= 177600  292#  5790  8007  8088
UNITOP  002100  1131#  1781*  1794  1948*  1949*  1950*  1951*  1952*  1954
UNRELO  035340  2146*  2179*  2219*  2259*  2301*  2349*  2390*  2438*  3343*  5699#  5995  6031
UPPFLG  001735  1074#  3994*  4009  4011*  4146*  4162  4164*
USERMA  035736  5680*  5713*  5747*  5787#
USESTK= 000600  168#  1408  7035  7969  7975
WARN1   006124  1572#
WARN2   022226  3498#  4928*
WARN3   022242  3503#  4929*
WARN4   022266  3515#  4930*
WARN5   022302  3520#  4931*
WARN6   031144  4941#
WARN7   020666  3268#
WASDBE= 104500  152#  1665  1674  1682  1690
WASSBE= 104476  150#
WAS1DB= 104501  153#  3924
WAS1SB= 104477  151#  3157  3163  3170  3176
WHICHC  043616  6127*  6141*  6630#
WHOBOX  052472  7728*  7850#
WHODUN= 000115  7861#  7881
WOOPEN  055032  8134  8137  8139  8152  8173  8181#  8182
WOOPS   054500  8004  8127#
WOOPSA  055062  8132*  8133*  8134  8168  8169  8172  8182#
WOOPUP  054656  8134  8135  8137  8151#  8173  8174  8182
WORST   002140  1150#  1272*  2207*  2247*  2337*  2426*  5875
WRITE0  002064  1123#  6920*  6929*  6973
XXDPCH  002024  1104#  1373*  7605
ZERO    002004  1096#
ZEROS   002006  1097#  2845  2846  4503  4507
$APTHD  060264  971  8872#
$AUTO   001450  1008#  1362*  1369*  1412  8412
$BANK   001403  988#  7525*
$BASE   060240  8844#
$BELL   002215  1177#  6466  6608  7569
$CACHF  032614  5254#  8936
$CACHN  032576  5248#  8935
$CBCSR  033526  5395#  8979
$CB1CS  033570  5402#  8980
$CDW1   060244  1915  1917  8849#
$CDW2   060246  1915  1918  8850#
$CHARC  055462  8226*  8236*  8243*  8248#
$CHKDI  034214  5496#  8987
$CHK1D  034250  5503#  8988
$CKSWR  056140  8393#  8919
$CLRCS  034162  5485#  8985
$CLR1C  034200  5490#  8986
$CMTAG  001400  984#  1258
$CMTGE  002132  1144#  1260
```

```
$CNTLC  057254          8426    8530    8579#
$CNTLG  057266          8414    8581#
$CNTLU  057261          8434    8554    8580#
$CPUOP  060212          8817#
$CRLF   002222          1179#   1955    4967    4970    6559    7627    7650    7659    7943    7956    7971    8225    8443
                        8482    8559
$DBLK   056130          8346    8376    8384#
$DB20   060044          7774*   8769#
$DDW0   060250          1301    8861#
$DDW1   060252          8862#
$DDW2   060254          8863#
$DDW3   060256          8864#
$DDW4   060260          8865#
$DDW5   060262          8866#
$DEENE  032566          5244#   8931
$DEVCT  060174          2105*   7484*   7486*   8807#
$DEVM   060242          8845#
$DIDDO= 000000            73#   1249#   1321    1329    1353    1356    1359    1367    1370    1412    1446#   1492    1546
                        1637    1640    1655    1656    1667    1675    1683    1694    1709    1718    1723    1724    1735
                        1752    1753    1787#   1794#   1815#   1816#   1817#   1818#   1819#   1820#   1821#   1822#   1828#
                        1829    1833    1837    1853    1860    1861    1862    1888    1889    1915    1921    1933    1934#
                        1935    1954#   1962#   1965    1969    1972    1985    1988    1994    2030    2033    2048    2049
                        2054#   2073    2102    2143#   2176#   2216#   2256#   2298#   2346#   2387#   2435#   2467    2475
                        2478#   2479    2482    2500    2527    2559    2578    2641#   2642#   2648#   2649#   2711    2727
                        2743    2759    2773    2787    2803    2819    2851#   2852#   2857    3057#   3058    3059#   3066#
                        3067    3068#   3075    3076#   3084#   3085    3086#   3094#   3095    3096#   3103#   3104    3105#
                        3117    3118#   3127    3130    3151    3156    3158    3162    3164    3169    3171    3175    3177
                        3233    3234#   3274    3280    3283    3292    3299    3302    3322    3323    3330    3334    3790
                        3791    3872    3873    3935    3936    4017    4018    4092    4093    4170    4171    4502    4515#
                        4516#   4517#   4518    4536#   4537#   4538#   4539    4541    4545    4550    4554    4572#   4960#
                        5077#   5091    5094    5267    5275    5290    5298    5344    5362#   5368#   5384#   5390#   5397#
                        5404#   5417    5419    5424    5429    5450    5452    5457    5462    5497#   5504#   5515    5518
                        5640#   5644    5658    5659    5660    5661    5662    5663    5672    5687    5849    5915    5927
                        5930    5954    5958    5970    5971    5972    5995#   6014    6031#   6032    6084    6089    6181
                        6183    6184    6219    6221    6222    6263    6267    6275    6276    6285    6288    6295    6316
                        6381    6410    6431    6434    6439    6462    6481    6495    6497    6498    6503    6506    6548
                        6554    6573    6576    6581    6604    6678    6743    6749    6760#   6762    6770    6786    6806
                        6821    6842    6862    6872    6877#   6881    6973#   6974#   7042#   7049    7307    7312    7321
                        7326    7336    7341    7393    7418    7428    7431    7434    7437    7485    7490    7494    7497
                        7560    7576    7577    7584    7589    7594    7600    7601    7605    7691    7694    7796    7803
                        7805    7809#   7813    7817    7818    7840    7863    7866    7881    7888    7898#   7900#   7902#
                        7905#   7907#   7909#   7954    7969    8004#   8027    8031    8079    8083    8470
$DOAGA  011710          2079    2081    2102#
$DOAGN  011660          2091#
$DOWN   054102          8049#
$DTBL   056120          8349    8380#
$ECCDI  033314          5361#   8975
$ECCIN  033402          5373#   8977
$ECC1D  033350          5367#   8976
$ECC1I  033422          5378#   8978
$ENASB  033440          5383#   8989
$ENAIS  033474          5389#   8990
$ENDAD  011650           962    1370    1382    2087#
$ENERG  032556          5240#   8930
$ENV    060204          1359    8812#
$ENVM   060205          1353    1356    8813#
```

```
$EOP    011532  2067#
$ERFLG  001404   989#  7510  7516*  7562*
$ERROR  050770  1286  7560#
$ERRTB  060606  7638  9019#
$ERRTY  051354  7588*  7626#
$ERTTL  001406   991#  1810  2049  2069  6546  6548  7570*  7571*  7605
$ESCAP  002032  1110#  7519*  7597  7599
$ETABL  060204  8811#
$ETEND  060264  8867#  8878
$EXHAL  037150  6017#
$ES     = 000001  1249#  1319#  1321#  1325#  1329#  1353#  1356#  1359#  1367#  1370#  1372#  1403#  1408#
                  1412#  1431#  1436#  1446#  1447#  1456#  1479#  1482#  1492#  1493#  1501#  1546#  1548#
                  1569#  1625#  1636#  1637#  1640#  1651#  1655#  1656#  1657#  1667#  1675#  1683#  1694#
                  1705#  1706#  1709#  1711#  1718#  1721#  1723#  1724#  1725#  1728#  1735#  1750#  1752#
                  1753#  1756#  1758#  1787#  1790#  1794#  1815#  1816#  1817#  1818#  1819#  1820#  1821#
                  1822#  1828#  1829#  1833#  1837#  1853#  1859#  1860#  1861#  1862#  1864#  1866#  1877#
                  1886#  1887#  1888#  1889#  1892#  1896#  1897#  1905#  1911#  1915#  1919#  1921#  1924#
                  1932#  1933#  1934#  1935#  1936#  1939#  1953#  1954#  1962#  1965#  1968#  1969#  1972#
                  1974#  1985#  1988#  1993#  1994#  2028#  2030#  2033#  2037#  2048#  2049#  2054#  2073#
                  2083#  2085#  2102#  2143#  2176#  2216#  2256#  2298#  2346#  2387#  2435#  2463#  2466#
                  2467#  2469#  2473#  2475#  2476#  2478#  2479#  2480#  2482#  2485#  2488#  2490#  2493#
                  2495#  2498#  2500#  2505#  2510#  2512#  2527#  2532#  2559#  2578#  2583#  2591#  2595#
                  2604#  2621#  2622#  2627#  2641#  2642#  2648#  2649#  2650#  2655#  2692#  2704#  2706#
                  2710#  2711#  2712#  2713#  2714#  2716#  2720#  2722#  2726#  2727#  2728#  2729#  2730#
                  2732#  2736#  2738#  2742#  2743#  2744#  2745#  2746#  2748#  2752#  2754#  2758#  2759#
                  2760#  2761#  2762#  2764#  2773#  2777#  2780#  2782#  2786#  2787#  2788#  2789#  2790#
                  2792#  2796#  2798#  2802#  2803#  2804#  2805#  2806#  2808#  2812#  2814#  2818#  2819#
                  2820#  2821#  2822#  2824#  2851#  2852#  2853#  2857#  2859#  3057#  3058#  3059#  3066#
                  3067#  3068#  3075#  3076#  3084#  3085#  3086#  3094#  3095#  3096#  3103#  3104#  3105#
                  3117#  3118#  3124#  3127#  3130#  3134#  3136#  3138#  3151#  3152#  3156#  3158#  3162#
                  3164#  3169#  3171#  3175#  3177#  3233#  3234#  3271#  3272#  3274#  3275#  3280#  3283#
                  3287#  3288#  3289#  3290#  3292#  3293#  3299#  3302#  3306#  3307#  3320#  3322#  3323#
                  3329#  3330#  3334#  3353#  3371#  3373#  3400#  3404#  3406#  3433#  3790#  3791#  3872#
                  3873#  3935#  3936#  4017#  4018#  4092#  4093#  4170#  4171#  4501#  4502#  4503#  4504#
                  4505#  4506#  4507#  4509#  4513#  4514#  4515#  4516#  4517#  4518#  4519#  4520#  4521#
                  4522#  4523#  4525#  4526#  4527#  4533#  4534#  4536#  4537#  4538#  4539#  4540#  4541#
                  4544#  4545#  4548#  4549#  4550#  4553#  4554#  4558#  4559#  4560#  4563#  4564#  4568#
                  4570#  4571#  4572#  4574#  4575#  4908#  4932#  4948#  4960#  4991#  5077#  5091#  5093#
                  5094#  5096#  5260#  5267#  5268#  5275#  5277#  5280#  5283#  5290#  5291#  5298#  5300#
                  5303#  5307#  5311#  5325#  5329#  5343#  5344#  5346#  5362#  5368#  5384#  5390#  5397#
                  5404#  5411#  5415#  5417#  5419#  5421#  5424#  5425#  5428#  5429#  5431#  5444#  5448#
                  5450#  5452#  5454#  5457#  5458#  5461#  5462#  5464#  5497#  5504#  5510#  5513#  5515#
                  5518#  5519#  5521#  5533#  5549#  5599#  5620#  5623#  5635#  5640#  5642#  5644#  5651#
                  5653#  5657#  5658#  5659#  5660#  5661#  5662#  5663#  5665#  5671#  5672#  5675#  5679#
                  5687#  5697#  5701#  5705#  5710#  5712#  5726#  5730#  5738#  5832#  5848#  5849#  5853#
                  5878#  5915#  5916#  5927#  5928#  5930#  5931#  5954#  5955#  5958#  5960#  5970#  5971#
                  5972#  5977#  5995#  6014#  6016#  6031#  6032#  6043#  6058#  6063#  6067#  6074#  6084#
                  6089#  6094#  6098#  6103#  6117#  6122#  6151#  6154#  6169#  6175#  6181#  6183#  6184#
                  6187#  6202#  6207#  6213#  6219#  6221#  6222#  6225#  6242#  6251#  6256#  6262#  6263#
                  6267#  6269#  6274#  6275#  6276#  6279#  6284#  6285#  6286#  6288#  6290#  6293#  6295#
                  6297#  6316#  6318#  6364#  6366#  6368#  6369#  6380#  6381#  6383#  6405#  6408#  6410#
                  6411#  6418#  6420#  6425#  6430#  6431#  6432#  6434#  6436#  6439#  6441#  6460#  6462#
                  6465#  6467#  6472#  6473#  6478#  6481#  6483#  6489#  6495#  6497#  6498#  6502#  6503#
                  6504#  6506#  6508#  6514#  6539#  6540#  6545#  6548#  6550#  6554#  6561#  6563#  6567#
                  6572#  6573#  6574#  6576#  6578#  6581#  6583#  6602#  6604#  6607#  6609#  6614#  6615#
                  6621#  6624#  6627#  6639#  6660#  6669#  6670#  6676#  6678#  6686#  6688#  6690#  6693#
```

```
                    6696#  6701#  6705#  6716#  6726#  6736#  6739#  6743#  6744#  6749#  6750#  6752#  6757#
                    6760#  6761#  6762#  6763#  6765#  6767#  6770#  6774#  6775#  6777#  6778#  6782#  6786#
                    6787#  6791#  6793#  6798#  6800#  6806#  6808#  6810#  6813#  6817#  6821#  6823#  6828#
                    6830#  6835#  6837#  6842#  6844#  6846#  6849#  6853#  6859#  6862#  6864#  6869#  6871#
                    6872#  6877#  6880#  6881#  6889#  6890#  6898#  6899#  6900#  6912#  6921#  6928#  6937#
                    6944#  6973#  6974#  6985#  6989#  6990#  7004#  7015#  7020#  7023#  7026#  7040#  7042#
                    7049#  7051#  7054#  7056#  7206#  7212#  7218#  7241#  7242#  7243#  7246#  7247#  7248#
                    7251#  7252#  7253#  7256#  7257#  7258#  7261#  7262#  7263#  7266#  7267#  7268#  7271#
                    7272#  7273#  7276#  7277#  7278#  7281#  7282#  7283#  7286#  7287#  7288#  7291#  7292#
                    7293#  7296#  7297#  7298#  7301#  7302#  7303#  7306#  7307#  7309#  7312#  7313#  7314#
                    7315#  7321#  7322#  7323#  7326#  7328#  7336#  7337#  7338#  7341#  7343#  7354#  7355#
                    7356#  7359#  7360#  7363#  7370#  7380#  7384#  7388#  7391#  7392#  7393#  7395#  7398#
                    7402#  7413#  7418#  7420#  7428#  7431#  7434#  7437#  7459#  7471#  7485#  7490#  7494#
                    7497#  7527#  7541#  7560#  7576#  7577#  7578#  7579#  7584#  7585#  7589#  7594#  7600#
                    7601#  7605#  7691#  7694#  7715#  7721#  7794#  7796#  7797#  7801#  7803#  7805#  7809#
                    7810#  7813#  7814#  7816#  7817#  7818#  7822#  7830#  7835#  7838#  7840#  7842#  7846#
                    7847#  7857#  7862#  7863#  7866#  7871#  7880#  7881#  7886#  7888#  7893#  7898#  7900#
                    7902#  7905#  7907#  7909#  7918#  7932#  7936#  7940#  7942#  7947#  7952#  7954#  7955
                    7960#  7961#  7967#  7969#  7970    7975#  7976#  7980#  7989#  7993#  7997#  8002#  8004#
                    8017#  8021#  8025#  8027#  8029#  8031#  8032#  8035#  8039#  8042#  8044#  8060#  8062#
                    8068#  8072#  8077#  8079#  8080#  8083#  8084#  8095#  8099#  8106#  8111#  8116#  8120#
                    8128#  8130#  8148#  8164#  8166#  8179#  8339#  8375#  8469#  8470#  8483#  8600#  8624#
                    8651#  8682#  8712#  8727#  8741#  8756#
$FATAL  060166      7581*  8804#
$FILLC  002214      1176#  8229    10140
$FILLS  002027      1107#  6626*   10142
$FS     = 000000    1249#  1321#  1329#  1353#  1356#  1359#  1367#  1370#  1412#  1446#  1492#  1546#  1637#
                    1640#  1655#  1656#  1667#  1675#  1683#  1694#  1709#  1718#  1723#  1724#  1735#  1752#
                    1753#  1787#  1794#  1815#  1816#  1817#  1818#  1819#  1820#  1821#  1822#  1828#  1829#
                    1833#  1837#  1853#  1860#  1861#  1862#  1888#  1889#  1915#  1921#  1933#  1934#  1935#
                    1954#  1962#  1965#  1969#  1972#  1985#  1988#  1994#  2030#  2033#  2048#  2049#  2054#
                    2073#  2102#  2143#  2176#  2216#  2256#  2298#  2346#  2387#  2435#  2467#  2475#  2478#
                    2479#  2482#  2500#  2527#  2559#  2578#  2641#  2642#  2648#  2649#  2711#  2727#  2743#
                    2759#  2773#  2787#  2803#  2819#  2851#  2852#  2857#  3057#  3058#  3059#  3066#  3067#
                    3068#  3075#  3076#  3084#  3085#  3086#  3094#  3095#  3096#  3103#  3104#  3105#  3117#
                    3118#  3127#  3130#  3151#  3156#  3158#  3162#  3164#  3169#  3171#  3175#  3177#  3233#
                    3234#  3274#  3280#  3283#  3292#  3299#  3302#  3322#  3323#  3330#  3334#  3790#  3791#
                    3872#  3873#  3935#  3936#  4017#  4018#  4092#  4093#  4170#  4171#  4502#  4515#  4516#
                    4517#  4518#  4536#  4537#  4538#  4539#  4541#  4545#  4550#  4554#  4572#  4960#  5077#
                    5091#  5094#  5267#  5275#  5290#  5298#  5344#  5362#  5368#  5384#  5390#  5397#  5404#
                    5417#  5419#  5424#  5429#  5450#  5452#  5457#  5462#  5497#  5504#  5515#  5518#  5640#
                    5644#  5658#  5659#  5660#  5661#  5662#  5663#  5672#  5687#  5849#  5915#  5927#  5930#
                    5954#  5958#  5970#  5971#  5972#  5995#  6014#  6031#  6032#  6084#  6089#  6181#  6183#
                    6184#  6219#  6221#  6222#  6263#  6267#  6275#  6276#  6285#  6288#  6295#  6316#  6381#
                    6410#  6431#  6434#  6439#  6462#  6481#  6495#  6497#  6498#  6503#  6506#  6548#  6554#
                    6573#  6576#  6581#  6604#  6678#  6743#  6749#  6760#  6762#  6770#  6786#  6806#  6821#
                    6842#  6862#  6872#  6877#  6881#  6973#  6974#  7042#  7049#  7307#  7312#  7321#  7326#
                    7336#  7341#  7393#  7418#  7428#  7431#  7434#  7437#  7485#  7490#  7494#  7497#  7560#
                    7576#  7577#  7584#  7589#  7594#  7600#  7601#  7605#  7691#  7694#  7796#  7803#  7805#
                    7809#  7813#  7817#  7818#  7840#  7863#  7866#  7881#  7888#  7898#  7900#  7902#  7905#
                    7907#  7909#  7954#  7969#  8004#  8027#  8031#  8079#  8083#  8470#
$GTSWR  056252      8415#  8918
$HALT   051202      7592#
$HALT2  060604      9002#
$HIBTS  060264      8873#
$HIOCT  057464      6174    6212    6488    8623*   8634#
```

```
$ILLUP  054472          7991    8053    8122#
$INTAG  001452          1009#   8428    8444
$ITEMB  001414           994#   7575#   7629
$I$    = 000001         1249#   1321#   1329#   1353#   1356#   1359#   1367#   1370#   1412#   1446#   1492#   1546#   1637#
                        1640#   1655#   1656#   1667#   1675#   1683#   1694#   1709#   1718#   1723#   1724#   1735#   1752#
                        1753#   1787#   1794#   1815#   1816#   1817#   1818#   1819#   1820#   1821#   1822#   1828#   1829#
                        1833#   1837#   1853#   1860#   1861#   1862#   1888#   1889#   1915#   1921#   1933#   1934#   1935#
                        1954#   1962#   1965#   1969#   1972#   1985#   1988#   1994#   2030#   2033#   2048#   2049#   2054#
                        2073#   2102#   2143#   2176#   2216#   2256#   2298#   2346#   2387#   2435#   2467#   2475#   2478#
                        2479#   2482#   2500#   2527#   2559#   2578#   2641#   2642#   2648#   2649#   2711#   2727#   2743#
                        2759#   2773#   2787#   2803#   2819#   2851#   2852#   2857#   3057#   3058#   3059#   3066#   3067#
                        3068#   3075#   3076#   3084#   3085#   3086#   3094#   3095#   3096#   3103#   3104#   3105#   3117#
                        3118#   3127#   3130#   3151#   3156#   3158#   3162#   3164#   3169#   3171#   3175#   3177#   3233#
                        3234#   3274#   3280#   3283#   3292#   3299#   3302#   3322#   3323#   3330#   3334#   3790#   3791#
                        3872#   3873#   3935#   3936#   4017#   4018#   4092#   4093#   4170#   4171#   4502#   4515#   4516#
                        4517#   4518#   4536#   4537#   4538#   4539#   4541#   4545#   4550#   4554#   4572#   4960#   5077#
                        5091#   5094#   5267#   5275#   5290#   5298#   5344#   5362#   5368#   5384#   5390#   5397#   5404#
                        5417#   5419#   5424#   5429#   5450#   5452#   5457#   5462#   5497#   5504#   5515#   5518#   5640#
                        5644#   5658#   5659#   5660#   5661#   5662#   5663#   5672#   5687#   5849#   5915#   5927#   5930#
                        5954#   5958#   5970#   5971#   5972#   5995#   6014#   6031#   6032#   6084#   6089#   6181#   6183#
                        6184#   6219#   6221#   6222#   6263#   6267#   6275#   6276#   6285#   6288#   6295#   6316#   6381#
                        6410#   6431#   6434#   6439#   6462#   6481#   6495#   6497#   6498#   6503#   6506#   6548#   6554#
                        6573#   6576#   6581#   6604#   6678#   6743#   6749#   6760#   6762#   6770#   6786#   6806#   6821#
                        6842#   6862#   6872#   6877#   6881#   6973#   6974#   7042#   7049#   7307#   7312#   7321#   7326#
                        7336#   7341#   7393#   7418#   7428#   7431#   7434#   7437#   7485#   7490#   7494#   7497#   7560#
                        7576#   7577#   7584#   7589#   7594#   7600#   7601#   7605#   7691#   7694#   7796#   7803#   7805#
                        7809#   7813#   7817#   7818#   7840#   7863#   7866#   7881#   7888#   7898#   7900#   7902#   7905#
                        7907#   7909#   7954#   7969#   8004#   8027#   8031#   8079#   8083#   8470#
$KERNE  032546          5236#   8929
$KMAP   034626          5623#   8933
$K$    = 000124         5424#   5457#   5518#   5661#   7813#   7881#   8031#   8083#
$L     = 000126           73#   1319#   1651#   1720#   1721#   1749#   1750#   1787#   1866#   1883#   1886#   1887#   1905#
                        1919#   1932#   1968#   2028#   2466#   2469#   2471#   2473#   2480#   2485#   3124#   3271#   3272#
                        3289#   3290#   3320#   4501#   4509#   4515#   4536#   4568#   4572#   5414#   5415#   5447#   5448#
                        5512#   5513#   5641#   5642#   5656#   5657#   5848#   6408#   6460#   6550#   6602#   6660#   6670#
                        6675#   6676#   6693#   6701#   6736#   6739#   6744#   6757#   6767#   6782#   6787#   6793#   6799#
                        6800#   6817#   6823#   6830#   6836#   6837#   6853#   6871#   7042#   7047#   7800#   7801#   7838#
                        7861#   7862#   7942#   7955#   7970#   8024#   8025#   8076#   8077#
$LF     002223          1180#   8569
$LL    = 000123           73#   1319#   1325#   1456#   1651#   1705#   1720#   1721#   1728#   1731#   1749#   1750#   1758#
                        1759#   1787#   1790#   1866#   1877#   1883#   1886#   1887#   1896#   1897#   1898#   1905#   1911#
                        1919#   1924#   1932#   1939#   1968#   1993#   2028#   2037#   2466#   2469#   2471#   2473#   2480#
                        2485#   2490#   2495#   2498#   2499#   2505#   2510#   2621#   2692#   3124#   3138#   3271#   3272#
                        3287#   3288#   3289#   3290#   3306#   3307#   3320#   3329#   4501#   4509#   4515#   4527#   4536#
                        4560#   4563#   4564#   4568#   4570#   4572#   4575#   5414#   5415#   5425#   5426#   5447#   5448#
                        5458#   5459#   5512#   5513#   5519#   5520#   5641#   5642#   5653#   5655#   5656#   5657#   5671#
                        5676#   5848#   5853#   6117#   6408#   6418#   6460#   6465#   6550#   6561#   6602#   6607#   6660#
                        6669#   6670#   6675#   6676#   6688#   6689#   6690#   6693#   6696#   6701#   6705#   6736#   6739#
                        6744#   6757#   6761#   6767#   6774#   6775#   6777#   6778#   6782#   6787#   6791#   6793#   6798#
                        6799#   6800#   6810#   6811#   6813#   6817#   6823#   6828#   6830#   6835#   6836#   6837#   6846#
                        6847#   6849#   6853#   6859#   6871#   6880#   6898#   6928#   6944#   7042#   7047#   7054#   7056#
                        7800#   7801#   7814#   7815#   7838#   7846#   7861#   7862#   7880#   7886#   7887#   7942#   7947#
                        7955#   7960#   7970#   7975#   8024#   8025#   8032#   8033#   8076#   8077#   8084#   8085#
$LOADC  032624          5260#   8938
$LOAD1  032740          5264    5285#
$LPADR  002170          1162#   1281#   3373    3383#   3384    3400#   3406    3416#   3417    3433#   7514#   7517#   7521
```

```
$LPERR  002172          1163#  1282*  3373   3384*  3400*  3406   3417*  3433*  7514   7518#  7595
$L$   = 000000            73#  1249   1321   1329   1353   1356   1359   1367   1370   1412#  1446   1492   1546#
                        1637   1640   1655   1656   1667   1675   1683   1694   1709#  1718   1723   1724#  1735
                        1752   1753   1787#  1794   1815   1816   1817   1818   1819   1820   1821   1822   1828
                        1829   1833   1837   1853#  1860#  1861#  1862   1888   1889   1915#  1921   1933   1934
                        1935   1954   1962   1965   1969   1972   1985   1988   1994#  2030   2033   2048   2049
                        2054   2073#  2102   2143   2176   2216   2256   2298   2346   2387   2435   2467   2475
                        2478   2479   2482   2500   2527   2559#  2578#  2641   2642   2648   2649   2711   2727
                        2743   2759   2773   2787   2803   2819   2851   2852   2857#  3057   3058#  3059   3066
                        3067#  3068   3075#  3076   3084   3085#  3086   3094   3095#  3096   3103   3104#  3105
                        3117#  3118   3127   3130   3151#  3156#  3158#  3162#  3164#  3169#  3171#  3175#  3177#
                        3233#  3234   3274   3280   3283   3292   3299   3302   3322   3323   3330   3334   3790#
                        3791#  3872#  3873#  3935#  3936#  4017#  4018#  4092#  4093#  4170#  4171#  4502   4515
                        4516   4517   4518   4536   4537   4538   4539   4541   4545   4550   4554   4572   4960
                        5077   5091   5094   5267   5275   5290   5298   5344   5362   5368   5384   5390   5397
                        5404   5417   5419   5424#  5429   5450   5452   5457#  5462   5497   5504   5515   5518#
                        5640   5644   5658   5659   5660   5661#  5662   5663   5672   5687   5849   5915   5927
                        5930   5954#  5958   5970   5971   5972#  5995   6014#  6031   6032   6084   6089   6181#
                        6183#  6184#  6219#  6221#  6222#  6263#  6267   6275#  6276   6285#  6288   6295   6316
                        6381   6410   6431#  6434   6439   6462   6481   6495#  6497#  6498#  6503#  6506   6548
                        6554   6573#  6576   6581   6604   6678   6743   6749   6760   6762   6770   6786   6806#
                        6821   6842   6862   6872   6877   6881   6973   6974   7042   7049   7307   7312   7321
                        7326   7336   7341   7393   7418   7428   7431   7434   7437   7485   7490   7494   7497#
                        7560   7576   7577   7584   7589   7594   7600   7601#  7605   7691   7694   7796   7803
                        7805   7809   7813#  7817   7818   7840   7863   7866   7881#  7888   7898   7900   7902
                        7905   7907   7909   7954   7969   8004   8027   8031#  8079   8083#  8470

$MADR1  060216          8831#
$MADR2  060222          8835#
$MADR3  060226          8838#
$MADR4  060232          8841#
$MAIL   060164          8802#  8874   8878
$MAMS1  060214          1967   8824#
$MAMS2  060220          8833#
$MAMS3  060224          8836#
$MAMS4  060230          8839#
$MBADR  060266          8874#
$MNEW   057304          8417   8583#
$MSGAD  060200          8809#
$MSGLG  060202          8810#
$MSGTY  060164          7602*  8803#
$MSWR   057273          8415   8582#
$MTYP1  060215          8825#
$MTYP2  060221          8834#
$MTYP3  060225          8837#
$MTYP4  060231          8840#
$NOTRA  060600          8914   8916   8966   8967   8968   8969   8970   8971   8972   8973   8992   8993   8994
                        8995   8996   8997   8998   8999   9000   9001#

$NULL   002026          1106#  8231
$NWTST= 000001          1434#  1522#  1552#  1747#  1999#  2017#  2024#
$OCNT   055710          8283*  8312*  8325#
$OCTVL  060146          8771   8796#  8797
$OCT8 = 060152          7776   8797#
$OMODE  055712          8278*  8282*  8287   8290*  8301*  8327#
$OVER   050636          7497   7515   7520#
$O$   = 000000          1249#  1321#  1329#  1353#  1356#  1359#  1367#  1370#  1412#  1446#  1492#  1546#  1637#
                        1640#  1655#  1656#  1667#  1675#  1683#  1694#  1709#  1718#  1723#  1724#  1735#  1752#
```

```
                    1753#  1787#  1794#  1815#  1816#  1817#  1818#  1819#  1820#  1821#  1822#  1828#  1829#
                    1833#  1837#  1853#  1860#  1861#  1862#  1888#  1889#  1915#  1921#  1933#  1934#  1935#
                    1954#  1962#  1965#  1969#  1972#  1985#  1988#  1994#  2030#  2033#  2048#  2049#  2054#
                    2073#  2102#  2143#  2176#  2216#  2256#  2298#  2346#  2387#  2435#  2467#  2475#  2478#
                    2479#  2482#  2500#  2527#  2559#  2578#  2641#  2642#  2648#  2649#  2711#  2727#  2743#
                    2759#  2773#  2787#  2803#  2819#  2851#  2852#  2857#  3057#  3058#  3059#  3066#  3067#
                    3068#  3075#  3076#  3084#  3085#  3086#  3094#  3095#  3096#  3103#  3104#  3105#  3117#
                    3118#  3127#  3130#  3151#  3156#  3158#  3162#  3164#  3169#  3171#  3175#  3177#  3233#
                    3234#  3274#  3280#  3283#  3292#  3299#  3302#  3322#  3323#  3330#  3334#  3790#  3791#
                    3872#  3873#  3935#  3936#  4017#  4018#  4092#  4093#  4170#  4171#  4502#  4515#  4516#
                    4517#  4518#  4536#  4537#  4538#  4539#  4541#  4545#  4550#  4554#  4572#  4960#  5077#
                    5091#  5094#  5267#  5275#  5290#  5298#  5344#  5362#  5368#  5384#  5390#  5397#  5404#
                    5417#  5419#  5424#  5429#  5450#  5452#  5457#  5462#  5497#  5504#  5515#  5518#  5640#
                    5644#  5658#  5659#  5660#  5661#  5662#  5663#  5672#  5687#  5849#  5915#  5927#  5930#
                    5954#  5958#  5970#  5971#  5972#  5995#  6014#  6031#  6032#  6084#  6089#  6181#  6183#
                    6184#  6219#  6221#  6222#  6263#  6267#  6275#  6276#  6285#  6288#  6295#  6316#  6381#
                    6410#  6431#  6434#  6439#  6462#  6481#  6495#  6497#  6498#  6503#  6506#  6548#  6554#
                    6573#  6576#  6581#  6604#  6678#  6743#  6749#  6760#  6762#  6770#  6786#  6806#  6821#
                    6842#  6862#  6872#  6877#  6881#  6973#  6974#  7042#  7049#  7307#  7312#  7321#  7326#
                    7336#  7341#  7393#  7418#  7428#  7431#  7434#  7437#  7485#  7490#  7494#  7497#  7560#
                    7576#  7577#  7584#  7589#  7594#  7600#  7601#  7605#  7691#  7694#  7796#  7803#  7805#
                    7809#  7813#  7817#  7818#  7840#  7863#  7866#  7881#  7888#  7898#  7900#  7902#  7905#
                    7907#  7909#  7954#  7969#  8004#  8027#  8031#  8079#  8083#  8470#
$PASS   060172      1352*  1925*  2070*  2071*  2077    2102    2108*  3059    3068    3076    3086    3096    3105
                    3118   3234   8806#
$PASTM  060272      8876#
$PATMA  001402       987#  7526*  7535*  7538*  7539    7540    7564    7565    7566    8114    8115
$PER01  046646      7202#  8941
$PER02  046674      7208#  8942
$PER03  046722      7214#  8943
$PER04  046752      7220#  8944
$PER07  047034      7236#  8945
$PER10  047054      7241#  8946
$PER11  047102      7246#  8947
$PER12  047120      7251#  8948
$PER13  047140      7256#  8949
$PER14  047156      7261#  8950
$PER15  047176      7266#  8951
$PER16  047216      7271#  8952
$PER17  047236      7276#  8953
$PER20  047254      7281#  8954
$PER21  047272      7286#  8955
$PER22  047312      7291#  8956
$PER23  047330      7296#  8957
$PER24  047350      7301#  8958
$PER25  047370      7306#  8959
$PER26  047600      7354#  8960
$PER27  047616      7358#  8961
$PER30  047754      7391#  8962
$PER31  050154      7427#  8963
$PER32  050240      7443#  8964
$PER33  050256      7448#  8965
$PWRDN  053552      1290   7988#  8117
$PWRUP  054106      8048   9053#  8180
$QUES   002221      1178#  8464   8562
$R    = 177777        73#  1249   1321    1329    1353    1356    1359    1367    1370    1412#  1446    1492    1546#
```

```
                    1637    1640    1655    1656    1667    1675    1683    1694    1709#   1718    1723    1724#   1735
                    1752    1753    1787#   1794    1815    1816    1817    1818    1819    1820    1821    1822    1828
                    1829    1833    1837    1853#   1860#   1861#   1862    1888    1889    1915#   1921    1933    1934
                    1935    1954    1962    1965    1969    1972    1985    1988    1994#   2030    2033    2048    2049
                    2054    2073#   2102    2143    2176    2216    2256    2298    2346    2387    2435    2467    2475
                    2478    2479    2482    2500    2527    2559#   2578#   2641    2642    2648    2649    2711    2727
                    2743    2759    2773    2787    2803    2819    2851    2852    2857#   3057    3058#   3059    3066
                    3067#   3068    3075#   3076    3084    3085#   3086    3094    3095#   3096    3103    3104#   3105
                    3117#   3118    3127    3130    3151#   3156#   3158#   3162#   3164#   3169#   3171#   3175#   3177#
                    3233#   3234    3274    3280    3283    3292    3299    3302    3322    3323    3330    3334    3790#
                    3791#   3872#   3873#   3935#   3936#   4017#   4018#   4092#   4093#   4170#   4171#   4502    4515
                    4516    4517    4518    4536    4537    4538    4539    4541    4545    4550    4554    4572    4960
                    5077    5091    5094    5267    5275    5290    5298    5344    5362    5368    5384    5390    5397
                    5404    5417    5419    5424#   5429    5450    5452    5457#   5462    5497    5504    5515    5518#
                    5640    5644    5658    5659    5660    5661#   5662    5663    5672    5687    5849    5915    5927
                    5930    5954#   5958    5970    5971    5972#   5995    6014#   6031    6032    6084    6089    6181#
                    6183#   6184#   6219#   6221#   6222#   6263#   6267    6275#   6276    6285#   6288    6295    6316
                    6381    6410    6431#   6434    6439    6462    6481    6495#   6497#   6498#   6503#   6506    6548
                    6554    6573#   6576    6581    6604    6678    6743    6749    6760    6762    6770    6786    6806#
                    6821    6842    6862    6872    6877    6881    6973    6974    7042    7049    7307    7312    7321
                    7326    7336    7341    7393    7418    7428    7431    7434    7437    7485    7490    7494    7497#
                    7560    7576    7577    7584    7589    7594    7600    7601#   7605    7691    7694    7796    7803
                    7805    7809    7813#   7817    7818    7840    7863    7866    7881#   7888    7898    7900    7902
                    7905    7907    7909    7954    7969    8004    8027    8031#   8079    8083#   8470
$RAND   057750      7043*   8741#
$RDCHR  056632      8494#   8921
$RDDEC  057466      8649#   8924
$RDLIN  056752      8521#   8922
$RDOCT  057316      8598#   8923
$READC  033056      5311#   8939
$RESRE  057712      8722#   8927
$SAVRE  057654      8711#   8926
$SAVR6  054476      8046*   8055    8056*   8057*   8124#   8163
$SCOPE  050444      1284    7484#
$STN    = 000001    1434#   1522#   1552#   1747#   1999#   2017#   2024#
$SVLAD  050622      7505    7511    7517#
$SV$    = 000000    1546#   1709#   1724#   1787#   1915#   1994#   2073#   2857#   3058#   3067#   3075#   3085#   3095#
                    3104#   3117#   3151#   3156#   3162#   3169#   3175#   3233#   5954#   5972#   6014#   6285#   6431#
                    6503#   6573#   6806#   7601#
$SWR    = 163000      71#   1434    1522    1552    1747    1999    2017    2024
$SWREG  060206      1357    8815#
$T      = 000550      73#   1249#   1321#   1329#   1353#   1356#   1359#   1367#   1370#   1372#   1412    1446#   1447#
                    1492#   1546#   1548#   1637#   1640#   1655#   1656#   1667#   1675#   1683#   1694#   1709#   1711#
                    1718#   1723#   1724#   1735#   1752#   1753#   1787#   1794#   1815#   1816#   1817#   1818#   1819#
                    1820#   1821#   1822#   1828#   1829#   1833#   1837#   1853    1860    1861    1862#   1888#   1889#
                    1915#   1921#   1933#   1934#   1935#   1954#   1962#   1965#   1969#   1972#   1974#   1985#   1988#
                    1994#   2030#   2033#   2048#   2049#   2054#   2073#   2102#   2143#   2176#   2216#   2256#   2298#
                    2346#   2387#   2435#   2467#   2475#   2478#   2479#   2482#   2500#   2527#   2532#   2559    2578
                    2604#   2641#   2642#   2648#   2649#   2655#   2711#   2713#   2727#   2729#   2743#   2745#   2759#
                    2761#   2773#   2787#   2789#   2803#   2805#   2819#   2821#   2851#   2852#   2857#   2859#   3057#
                    3058#   3059#   3066#   3067#   3068#   3075#   3076#   3084#   3085#   3086#   3094#   3095#   3096#
                    3103#   3104#   3105#   3117#   3118#   3127#   3130#   3151#   3156#   3158    3162#   3164    3169#
                    3171    3175#   3177    3233#   3234#   3274#   3280#   3283#   3292#   3299#   3302#   3322#   3323#
                    3330#   3334#   3790    3791    3872    3873    3935    3936    4017    4018    4092    4093    4170
                    4171    4502#   4505#   4515#   4516#   4517#   4518#   4521#   4536#   4537#   4538#   4539#   4541#
                    4545#   4548#   4550#   4554#   4572#   4960#   5077#   5091#   5093#   5094#   5096#   5267#   5275#
```

```
                    5277#  5290#  5298#  5300#  5344#  5346#  5362#  5368#  5384#  5390#  5397#  5404#  5417#
                    5419#  5424   5429#  5431#  5450#  5452#  5457   5462#  5464#  5497#  5504#  5515#  5518
                    5640#  5644#  5658#  5659#  5660#  5661   5662#  5663#  5672#  5687#  5849#  5915#  5927#
                    5930#  5954#  5958#  5960#  5970#  5971#  5972#  5995#  6014#  6016#  6031#  6032#  6084#
                    6089#  6103#  6181   6183   6184   6219   6221   6222   6263   6267#  6275   6276#  6285#
                    6288#  6295#  6316#  6318#  6381#  6383#  6410#  6431#  6434#  6439#  6441#  6462#  6481#
                    6483#  6495   6497   6498   6503#  6506#  6548#  6554#  6573#  6576#  6581#  6583#  6604#
                    6678#  6743#  6749#  6760#  6762#  6770#  6786#  6806#  6821#  6842#  6862#  6864#  6872#
                    6877#  6881#  6890#  6921#  6937#  6973#  6974#  7042#  7049#  7051#  7307#  7312#  7314#
                    7321#  7323#  7326#  7328#  7336#  7338#  7341#  7343#  7393#  7418#  7420#  7428#  7431#
                    7434#  7437#  7485#  7490#  7494#  7497   7560#  7576#  7577#  7584#  7589#  7594#  7600#
                    7601#  7605#  7691#  7694#  7796#  7803#  7805#  7809#  7813   7817#  7818#  7822#  7840#
                    7842#  7863#  7866#  7871#  7881   7888#  7898#  7900#  7902#  7905#  7907#  7909#  7954#
                    7961#  7969#  7976#  8004#  8027#  8031   8079#  8083   8470#
$TESTN  060170      7565*  8805#
$TKB    002206      1173#  4952   4990   6303   6365   6396   6406   6449   6471   6520   6538   6591   6613
                    8395   8422   8498   8504
$TKS    002204      1172#  4954*  4989*  6305*  6363*  6398*  6404*  6451*  6470*  6522*  6537*  6593*  6612*
                    8393   8420   8430*  8446*  8496   8502
$TN   = 000010        72#  1434#  1522#  1552#  1747#  1999#  2017#  2024#
$TPB    002212      1175#  8240#
$TPFLG  002030      1108#  1354*  8208
$TPS    002210      1174#  8238
$TRAP   060300      1288   8888#
$TRAP2  060322      8899#  8910
$TRPAD  060334      8893   8910#
$TSTM   060270      8875#
$TSTRD  033146      5329#  8991
$TTYIN  057230      8523   8524   8542   8560   8574   8578#
$TYPDS  055714      8339#  8915
$TYPE   055272      8208#  8911
$TYPEC  055416      8228*  8235*  8238#  8239   8450*
$TYPEX  055464      8244   8246   8249#
$TYPOC  055512      8281#  8912
$TYPON  055526      8280   8283#
$TYPOS  055466      8276#  8913
$T1   = 000000      1323#  1325#  1334#  1355#  1358#  1366#  1372#  1374#  1375#  1456#  1495#  1548#  1550#
                    1639#  1642#  1691#  1692#  1693#  1698#  1703#  1704#  1705#  1711#  1713#  1726#  1727#
                    1728#  1731#  1742#  1743#  1755#  1757#  1758#  1759#  1790#  1832#  1836#  1840#  1865#
                    1877#  1894#  1895#  1896#  1897#  1898#  1911#  1923#  1924#  1927#  1937#  1938#  1939#
                    1974#  1976#  1987#  1990#  1992#  1993#  1996#  1997#  2035#  2036#  2037#  2051#  2052#
                    2076#  2110#  2490#  2494#  2495#  2496#  2497#  2498#  2499#  2504#  2505#  2506#  2510#
                    2532#  2534#  2621#  2692#  2713#  2715#  2729#  2731#  2745#  2747#  2761#  2763#  2776#
                    2789#  2791#  2805#  2807#  2821#  2823#  2859#  2861#  3060#  3069#  3077#  3087#  3097#
                    3106#  3119#  3132#  3137#  3138#  3153#  3159#  3165#  3172#  3178#  3235#  3282#  3285#
                    3286#  3287#  3288#  3301#  3304#  3305#  3306#  3307#  3327#  3328#  3329#  3338#  3345#
                    4505#  4508#  4521#  4524#  4527#  4543#  4547#  4548#  4552#  4556#  4557#  4560#  4563#
                    4564#  4570#  4575#  5093#  5096#  5098#  5099#  5277#  5279#  5281#  5300#  5302#  5304#
                    5346#  5348#  5422#  5423#  5425#  5426#  5431#  5433#  5455#  5456#  5458#  5459#  5464#
                    5466#  5517#  5519#  5520#  5652#  5653#  5655#  5666#  5667#  5668#  5669#  5670#  5671#
                    5674#  5676#  5689#  5851#  5853#  5917#  5929#  5932#  5956#  5960#  5962#  5972#  5979#
                    5980#  5981#  6016#  6019#  6034#  6088#  6093#  6117#  6270#  6280#  6287#  6291#  6298#
                    6318#  6320#  6383#  6385#  6417#  6418#  6433#  6437#  6441#  6443#  6464#  6465#  6483#
                    6485#  6505#  6509#  6560#  6561#  6562#  6575#  6579#  6583#  6585#  6606#  6607#  6669#
                    6687#  6688#  6689#  6690#  6696#  6705#  6753#  6761#  6766#  6773#  6774#  6775#  6776#
                    6777#  6778#  6791#  6798#  6809#  6810#  6811#  6812#  6813#  6828#  6835#  6845#  6846#
```

```
                    6847#   6848#   6849#   6859#   6864#   6866#   6879#   6880#   6883#   6898#   6928#   6944#   7051#
                    7053#   7054#   7056#   7311#   7314#   7316#   7323#   7325#   7328#   7330#   7338#   7340#   7343#
                    7345#   7397#   7420#   7422#   7430#   7433#   7436#   7439#   7488#   7493#   7496#   7572#   7580#
                    7586#   7587#   7596#   7601#   7604#   7610#   7611#   7612#   7693#   7697#   7811#   7812#   7814#
                    7815#   7822#   7825#   7827#   7828#   7842#   7844#   7846#   7880#   7882#   7883#   7886#   7887#
                    7890#   7947#   7960#   7961#   7963#   7975#   7976#   7978#   8030#   8032#   8033#   8082#   8084#
                    8085#   8472#
$T2     = 000547    1323#   1325#   1334#   1355#   1358#   1366#   1372#   1374#   1375#   1456#   1495#   1548#   1550#
                    1639#   1642#   1691#   1692#   1693#   1698#   1703#   1704#   1705#   1711#   1713#   1726#   1727#
                    1728#   1731#   1742#   1743#   1755#   1757#   1758#   1759#   1790#   1832#   1836#   1840#   1865#
                    1877#   1894#   1895#   1896#   1897#   1898#   1911#   1923#   1924#   1927#   1937#   1938#   1939#
                    1974#   1976#   1987#   1990#   1992#   1993#   1996#   1997#   2035#   2036#   2037#   2051#   2052#
                    2076#   2110#   2490#   2494#   2495#   2496#   2497#   2498#   2499#   2504#   2505#   2506#   2510#
                    2532#   2534#   2621#   2692#   2713#   2715#   2729#   2731#   2745#   2747#   2761#   2763#   2776#
                    2789#   2791#   2805#   2807#   2821#   2823#   2859#   2861#   3060#   3069#   3077#   3087#   3097#
                    3106#   3119#   3132#   3137#   3138#   3153#   3159#   3165#   3172#   3178#   3235#   3282#   3285#
                    3286#   3287#   3288#   3301#   3304#   3305#   3306#   3307#   3327#   3328#   3329#   3338#   3345#
                    4505#   4508#   4521#   4524#   4527#   4543#   4547#   4548#   4552#   4556#   4557#   4560#   4563#
                    4564#   4570#   4575#   5093#   5096#   5098#   5099#   5277#   5279#   5281#   5300#   5302#   5304#
                    5346#   5348#   5422#   5423#   5425#   5426#   5431#   5433#   5455#   5456#   5458#   5459#   5464#
                    5466#   5517#   5519#   5520#   5652#   5653#   5655#   5666#   5667#   5668#   5669#   5670#   5671#
                    5674#   5676#   5689#   5851#   5853#   5917#   5929#   5932#   5956#   5960#   5962#   5979#   5980#
                    5981#   6016#   6019#   6034#   6088#   6093#   6117#   6270#   6280#   6287#   6291#   6298#   6318#
                    6320#   6383#   6385#   6417#   6418#   6433#   6437#   6441#   6443#   6464#   6465#   6483#   6485#
                    6505#   6509#   6560#   6561#   6562#   6575#   6579#   6583#   6585#   6606#   6607#   6669#   6687#
                    6688#   6689#   6690#   6696#   6705#   6753#   6761#   6766#   6773#   6774#   6775#   6776#   6777#
                    6778#   6791#   6798#   6809#   6810#   6811#   6812#   6813#   6828#   6835#   6845#   6846#   6847#
                    6848#   6849#   6859#   6864#   6866#   6879#   6880#   6883#   6898#   6928#   6944#   7051#   7053#
                    7054#   7056#   7311#   7314#   7316#   7323#   7325#   7328#   7330#   7338#   7340#   7343#   7345#
                    7397#   7420#   7422#   7430#   7433#   7436#   7439#   7488#   7493#   7496#   7572#   7580#   7586#
                    7587#   7596#   7604#   7610#   7611#   7612#   7693#   7697#   7811#   7812#   7814#   7815#   7822#
                    7825#   7827#   7828#   7842#   7844#   7846#   7880#   7882#   7883#   7886#   7887#   7890#   7947#
                    7960#   7961#   7963#   7975#   7976#   7978#   8030#   8032#   8033#   8082#   8084#   8085#   8472#
$UNIT   060176      2106*   7487*   8808#
$UNITM  060274      8877#
$USWR   060210      2102    8816#
$VECT1  060234      8842#
$VECT2  060236      8843#
$WASDB  033774      5444#   8983
$WASSB  033630      5411#   8981
$WAS1D  034110      5470#   8984
$WAS1S  033744      5437#   8982
$XTSTR  050532      7499#
$Y$     = 000000    1319#   1446#   1493#   1651#   1721#   1750#   1794#   1815#   1816#   1817#   1818#   1819#   1820#
                    1821#   1822#   1866#   1886#   1887#   1905#   1919#   1932#   1934#   1936#   1953#   1954#   1968#
                    2028#   2466#   2469#   2473#   2480#   2485#   2488#   2641#   2642#   2648#   2649#   2704#   2706#
                    2710#   2712#   2714#   2716#   2720#   2722#   2726#   2728#   2730#   2732#   2736#   2738#   2742#
                    2744#   2746#   2748#   2752#   2754#   2758#   2760#   2762#   2764#   2780#   2782#   2786#   2788#
                    2790#   2792#   2796#   2798#   2802#   2804#   2806#   2808#   2812#   2814#   2818#   2820#   2822#
                    2824#   3124#   3271#   3272#   3275#   3289#   3290#   3293#   3320#   4501#   4503#   4504#   4506#
                    4507#   4509#   4513#   4514#   4516#   4517#   4519#   4520#   4522#   4523#   4525#   4526#   4533#
                    4534#   4537#   4538#   4540#   4544#   4549#   4553#   4558#   4559#   4568#   4571#   4574#   5362#
                    5368#   5384#   5390#   5397#   5404#   5415#   5448#   5497#   5504#   5513#   5642#   5657#   5679#
                    5712#   5848#   5977#   6408#   6411#   6460#   6550#   6602#   6660#   6670#   6676#   6693#   6701#
                    6736#   6739#   6744#   6757#   6760#   6767#   6782#   6787#   6793#   6800#   6817#   6823#   6830#
                    6837#   6853#   6871#   6877#   7241#   7242#   7243#   7246#   7247#   7248#   7251#   7252#   7253#
```

```
                    7256#  7257#  7258#  7261#  7262#  7263#  7266#  7267#  7268#  7271#  7272#  7273#  7276#
                    7277#  7278#  7281#  7282#  7283#  7286#  7287#  7288#  7291#  7292#  7293#  7296#  7297#
                    7298#  7301#  7302#  7303#  7306#  7309#  7313#  7315#  7322#  7337#  7354#  7355#  7359#
                    7370#  7391#  7392#  7395#  7578#  7579#  7801#  7809#  7838#  7862#  7898#  7900#  7902#
                    7905#  7907#  7909#  7942#  7955#  7970#  8025#  8077#
$ZAP42  011630      2080#  7609
$Z$     = 000000    1319#  1446#  1493#  1651#  1721#  1750#  1794#  1815#  1816#  1817#  1818#  1819#  1820#
                    1821#  1822#  1866#  1886#  1887#  1905#  1919#  1932#  1934#  1936#  1953#  1954#  1968#
                    2028#  2466#  2469#  2473#  2480#  2485#  2488#  2641#  2642#  2648#  2649#  2704#  2706#
                    2710#  2712#  2714#  2716#  2720#  2722#  2726#  2728#  2730#  2732#  2736#  2738#  2742#
                    2744#  2746#  2748#  2752#  2754#  2758#  2760#  2762#  2764#  2780#  2782#  2786#  2788#
                    2790#  2792#  2796#  2798#  2802#  2804#  2806#  2808#  2812#  2814#  2818#  2820#  2822#
                    2824#  3124#  3271#  3272#  3275#  3289#  3290#  3293#  3320#  4501#  4503#  4504#  4506#
                    4507#  4509#  4513#  4514#  4516#  4517#  4519#  4520#  4522#  4523#  4525#  4526#  4533#
                    4534#  4537#  4538#  4540#  4544#  4549#  4553#  4558#  4559#  4568#  4571#  4574#  5362#
                    5368#  5384#  5390#  5397#  5404#  5415#  5448#  5497#  5504#  5513#  5642#  5657#  5679#
                    5712#  5848#  5977#  6408#  6411#  6460#  6550#  6602#  6660#  6670#  6676#  6693#  6701#
                    6736#  6739#  6744#  6757#  6760#  6767#  6782#  6787#  6793#  6800#  6817#  6823#  6830#
                    6837#  6853#  6871#  6877#  7241#  7242#  7243#  7246#  7247#  7248#  7251#  7252#  7253#
                    7256#  7257#  7258#  7261#  7262#  7263#  7266#  7267#  7268#  7271#  7272#  7273#  7276#
                    7277#  7278#  7281#  7282#  7283#  7286#  7287#  7288#  7291#  7292#  7293#  7296#  7297#
                    7298#  7301#  7302#  7303#  7306#  7309#  7313#  7315#  7322#  7337#  7354#  7355#  7359#
                    7370#  7391#  7392#  7395#  7578#  7579#  7801#  7809#  7838#  7862#  7898#  7900#  7902#
                    7905#  7907#  7909#  7942#  7955#  7970#  8025#  8077#
$$S     = 000000    1367#  1656#  1723#  1752#  1935#  2475#  2851#  2852#  3274#  3292#  3323#  5644#  5662#
                    5663#  5849#  5915#  5927#  5930#  5970#  6288#  6410#  6434#  6462#  6506#  6576#  6604#
                    6678#  6743#  6786#  6821#  7490#  7584#  7594#  7605#  7691#  7866#  7900#  7902#  7907#
                    7909#
$$T     = 000544    1249#  1446#  1546#  1709#  1724#  1787#  1790#  1794#  1815#  1816#  1817#  1818#  1819#
                    1820#  1821#  1822#  1828#  1915#  1934#  1954#  1962#  1994#  2054#  2073#  2143#  2176#
                    2216#  2256#  2298#  2346#  2387#  2435#  2478#  2641#  2642#  2648#  2649#  2851#  2852#
                    2857#  3057#  3058#  3059#  3066#  3067#  3068#  3075#  3076#  3084#  3085#  3086#  3094#
                    3095#  3096#  3103#  3104#  3105#  3117#  3118#  3151#  3156#  3162#  3169#  3175#  3233#
                    3234#  4516#  4517#  4527#  4537#  4538#  4560#  4575#  4960#  5077#  5362#  5368#  5384#
                    5390#  5397#  5404#  5497#  5504#  5640#  5954#  5972#  5995#  6014#  6031#  6285#  6431#
                    6503#  6573#  6760#  6806#  6877#  6973#  6974#  7056#  7601#  7809#  7898#  7900#  7902#
                    7905#  7907#  7909#  8004#
$$TT    = 000547    1412#  1546#  1709#  1724#  1787#  1853#  1860#  1861#  1915#  1994#  2073#  2559#  2578#
                    2857#  3058#  3067#  3075#  3085#  3095#  3104#  3117#  3151#  3156#  3158#  3162#  3164#
                    3169#  3171#  3175#  3177#  3233#  3790#  3791#  3872#  3873#  3935#  3936#  4017#  4018#
                    4092#  4093#  4170#  4171#  5424#  5457#  5518#  5661#  5954#  5972#  6014#  6181#  6183#
                    6184#  6219#  6221#  6222#  6263#  6275#  6285#  6431#  6495#  6497#  6498#  6503#  6573#
                    6806#  7497#  7601#  7813#  7881#  8031#  8083#
$OFILL  055711      8277*  8281*  8291    8326#
.       = 076350     939#   944    961#   963#   966#   968#   970#   972#   975#   980#   990#   995#  1010#
                    1033#  1034#  1035#  1036#  1109#  1181#  1226#  1232#  1238#  1239#  1240#  1241#  1261
                    1321#  1667#  1675#  1683#  1694#  1790   1921#  2143#  2176#  2216#  2256#  2298#  2346#
                    2387#  2435#  2479#  2482#  3127#  3158#  3164#  3171#  3177#  3334#  3541    4527    4560
                    4575    4859    4896   5275#  5298#  5417#  5429#  5450#  5462#  5515#  5687#  6015    7049#
                    7056    7485#  7803#  7840#  8027#  8050   8079#  8123   8181#  8182#  8384#  8578#  8584#
                    8796#  9308#  9347#  9481#  9485#  10032  10036  10083  10102  10227  10254  10272  10273#
                   10279#  10295#  10297#  10299#  10300#

. ABS.  076350      000
```

```
ERRORS DETECTED:  0

DSKW:[400,3457]CEMKAA.BIC,DSKW:[400,3457]CEMKAA/EX/CRF:SYM=CEMKAA.MA1,CEMKAA.MA2
RUN-TIME:  145  161  11 SECONDS
RUN-TIME RATIO: 3721/318=11.6
CORE USED:  28K  (55 PAGES)


USER SYMBOLS:  1727
MACRO  NAMES:  153
UNDF SYMBOLS:  2
DISK BLOCKS  READ:  1380
DISK BLKS WRITTEN:  1778
KILO CORE SECONDS:  6660
```

EOF1CEMKAASEQ 00010000 780330 PDP10 411 2