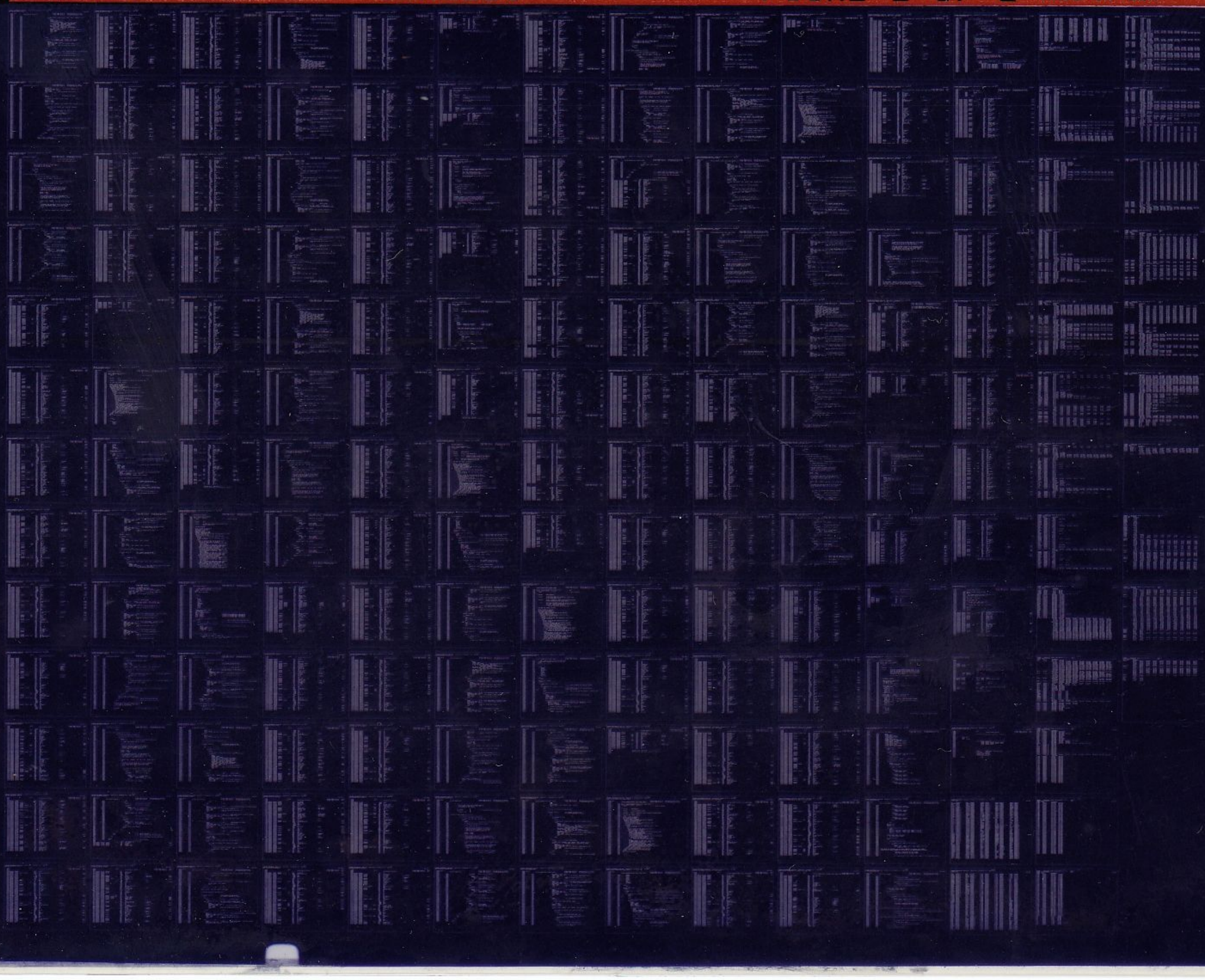
ML11

PERF EXERCISER
CZMLBBO

AH-S430B-MC
FICHE 1 OF 2

ML11 PERF EXERCISER
CZMLBBO

AH-S430B-MC
FICHE 2 OF 2



.REM &

IDENTIFICATION

PRODUCT CODE: AC-S428B-MC

PRODUCT NAME: CZMLBBO ML11 PERFORMANCE EXERCISER

PRODUCT DATE: 19-MAR-82

MAINTAINER: MEMORY DIAGNOSTICS ENGINEERING

AUTHOR: MICHELE D. ROSEN

TABLE OF CONTENTS

1.0 GENERAL INFORMATION

1.1 PROGRAM ABSTRACT

THE ML11 PERFORMANCE EXERCISER IS A BLISS PROGRAM WHICH RUNS UNDER THE PDP-11 DIAGNOSTIC SUPERVISOR AND WHICH EXERCISES UP TO 8 ML11 UNITS ON A SINGLE RH CONTROLLER. AN ML11 UNIT IS A FAST, RANDOM ACCESS, BLOCK MODE MOS MEMORY SYSTEM WITH ECC CAPABILITY. IT IS MADE UP OF 3 CONTROL MODULES AND UP TO 16 ARRAY MODULES. IT IS A MASSBUS DEVICE, AND AS SUCH, CONFORMS TO MASSBUS STANDARDS.

THIS DIAGNOSTIC HAS BEEN WRITTEN FOR USE WITH THE DIAGNOSTIC RUNTIME SERVICES SOFTWARE (SUPERVISOR). THESE SERVICES PROVIDE THE INTERFACE TO THE OPERATOR AND TC THE SOFTWARE ENVIRONMENT. THIS PROGRAM CAN BE USED WITH XXDP+, ACT, APT, SLIDE AND PAPER TAPE. FOR A COMPLETE DESCRIPTION OF THE RUNTIME SERVICES, REFER TO THE XXDP+ USER'S MANUAL. THERE IS A BRIEF DESCRIPTION OF THE RUNTIME SERVICES IN SECTION 2 OF THIS DOCUMENT.

1.2 SYSTEM REQUIREMENTS

1. PDP-11 WITH 28K WORDS OF MEMORY
2. CONSOLE TERMINAL
3. RH11 OR RH70 CONTROLLER
4. 1 TO 8 ML11A OR ML11B DRIVES
5. XXDP+ LOAD MEDIA

1.3 RELATED DOCUMENTS AND STANDARDS

THE HARDWARE DESIGN IS EXPECTED TO CONFORM TO THE STAI'DARDS SET FORTH IN THE MASSBUS SPECIFICATION (DEC STANDARD 159).

THE FOLLOWING DOCUMENTATION MAY PROVE USEFUL IN LEARNING MORE ABOUT THE ML11:

1. ML11 ENGINEERING SPECIFICATION
2. RWS04 FIXED-HEAD DISK SUBSYSTEM USER'S MANUAL
3. THE ML11 PERFORMANCE EXERCISER'S PROGRAM LISTING
4. THE ML11 LOGIC TEST'S SPECIFICATIONS AND LISTINGS
5. ML11 PROJECT PLAN
6. XXDP+ USER'S MANUAL (CHQUS)
7. PDP-11 DIAGNOSTIC SUPERVISOR DOCUMENTATION (SUPINT, SUPFUN,

1.4 DIAGNOSTIC HIERARCY PREREQUISITES

ML11 SUBSYSTEM FAULTS WILL BE DETECTED BUT NOT ISOLATED, SINCE OTHER DIAGNOSTICS WILL BE AVAILABLE FOR TROUBLESHOOTING THE EXACT CAUSE OF

FAILURE. A LIST OF AVAILABLE DIAGNOSTIC TOOLS IS INCLUDED IN APPENDIX A OF THE ML11 FUNCTIONAL SPECIFICATION.

1.5 ASSUMPTIONS

2.0 OPERATING INSTRUCTIONS

THIS SECTION CONTAINS A VERY BRIEF DESCRIPTION OF THOSE PARTS OF THE DIAGNOSTIC RUNTIME SERVICES OF THE PDP-11 DIAGNOSTIC SUPERVISOR WHICH ARE APPLICABLE FOR THIS EXERCISER. CONSULT THE XXDP+ USER'S MANUAL (CHQUS) FOR MORE DETAILS.

2.1 COMMANDS

THE FOLLOWING IS A LIST OF THE 11 COMMANDS AVAILABLE TO THE DRS. ANY COMMAND IS RECOGNIZED BY ITS FIRST 3 CHARACTERS, AND MANY COMMANDS MAY BE MODIFIED BY OPTIONAL SWITCHES WHICH ARE DESCRIBED IN THE NEXT SECTION.

DRS COMMAND	EFFECT
START	START THE DIAGNOSTIC FROM AN INITIAL STATE
RESTART	START THE DIAGNOSTIC WITHOUT INITIALIZING
CONTINUE	CONTINUE AT TEST THAT WAS INTERRUPTED (AFTER ^C)
PROCEED	CONTINUE FROM AN ERROR HALT
EXIT	RETURN TO THE XXDP+ MONITOR
ADD	ACTIVATE A UNIT FOR TESTING (ALL UNITS ARE CONSIDERED TO BE ACTIVE AT START TIME)
DROP	DEACTIVATE A UNIT
PRINT	PRINT STATISTICAL INFORMATION
DISPLAY	TYPE A LIST OF ALL DEVICE INFORMATION
FLAGS	TYPE THE STATE OF ALL FLAGS (SEE SECTION 2.3)
ZFLAGS	CLEAR ALL FLAGS (SEE SECTION 2.3)

2.2 SWITCHES

SWITCHES ARE USED TO MODIFY PROGRAM OPERATION, AND ARE APPENDED TO COMMANDS. THE SWITCHES WHICH HAVE MEANING FOR THIS EXERCISER AND INFORMATION ABOUT THEIR USE ARE GIVEN IN THE FOLLOWING TWO TABLES.

SWITCH	EFFECT
/PASS:DDDDD	EXECUTE DDDDD PASSES (DDDDD = 1 TO 64000)
/FLAGS:FLGS	SET SPECIFIED FLAGS (SEE SECTION 2.3)
/EOP:DDDDD	REPORT END OF PASS MESSAGE AFTER EVERY

DDDDD PASSES ONLY (DDDDD = 1 TO 64000).

/UNITS:LIST TEST/ADD/DROP ONLY THOSE UNITS SPECIFIED IN THE LIST. LIST EXAMPLE: /UNITS:0:3:5-7 USE UNITS 0,3,5,6,7 (UNIT NUMBERS = 0-7)

EXAMPLE OF SWITCH USAGE:

START/PASS:1000/EOP:100

THE EFFECT OF THIS COMMAND WILL BE:

1. ALL UNITS WILL BE TESTED 1000 TIM

2. THE END OF PASS MESSAGES WILL BE PRINTED AFTER EACH 100 PASSES ONLY.

BELOW IS A TABLE THAT SPECIFIES WHICH SWITCHES CAN BE USED BY EACH COMMAND.

	PASS	FLAGS	EOP	UNITS
START	X	X	X	X
RESTART	X	X	X	X
CONTINUE	X	X	X	
PROCEED		X		
DROP				X
ADD				X
PRINT				
DISPLAY				X
FLAGS				
ZFLAGS				
EXIT				

2.3 FLAGS

FLAGS ARE USED TO SET UP CERTAIN OPERATION PARAMETERS SUCH AS LOOPING ON ERROR. ALL FLAGS ARE CLEARED AT STARTUP AND REMAIN CLEARED UNTIL EXPLICITLY SET USING THE FLAGS SWITCH. FLAGS ARE ALSO CLEARED AFTER A START COMMAND UNLESS SET USING THE FLAG SWITCH. THE ZFLAGS COMMAND MAY ALSO BE USED TO CLEAR ALL FLAGS. WITH THE EXCEPTION OF THE START AND ZFLAGS COMMANDS, NO COMMANDS AFFECT THE STATE OF THE FLAGS. THEY REMAIN SET OF CLEARED AS SPECIFED BY THE LAST FLAG SWITCH.

FLAG	EFFECT
HOE	HALT ON ERROR - CONTROL IS RETURNED TO DRS COMMAND MODE
LOE	LOOP ON ERROR
IER*	INHIBIT ALL ERROR REPORTS
IBE*	INHIBIT ALL ERROR REPORTS EXCEPT FIRST LEVEL (FIRST LEVEL CONTAINS ERROR TYPE, NUMBER, PC, TEST AND UNIT)

IXE*	INHIBIT EXTENDED ERROR REPORTS (THOSE CALLED BY PRINTX MACRO'S)
PRI	DIRECT MESSAGES TO LINE PRINTER
BOE	'BELL'' ON ERROR
ISR	INHIBIT STATISTICAL REPORTS
IDU	INHIBIT PROGRAM DROPPING OF UNITS
ADR	EXECUTE AUTODROP CODE

* ERROR MESSAGES ARE DESCRIBED IN SECTION 4.7

SEE THE XXDP+ USER'S MANUAL FOR MORE DETAILS ON FLAGS. MORE THAN ONE FLAG MAY BE SPECIFIED WITH THE /FLAGS SWITCH. FOR EXAMPLE, TO CAUSE THE PROGRAM TO LOOP ON ERROR, INHIBIT ERROR REPORTS AND TYPE A 'BELL'' ON ERROR, YOU MAY USE THE FOLLOWING STRING:

/FLAGS:LOE:IER:BOE

2.4 HARDWARE QUESTIONS

WHEN A DIAGNOSTIC IS STARTED, THE DIAGNOSTIC RUNTIME SERVICES PROMPTS THE USER FOR HARDWARE INFORMATION BY TYPING "CHANGE HW (L) ?". YOU MUST ANSWER 'Y' AFTER A START COMMAND UNLESS THE HARDWARE INFORMATION HAS BEEN PRELOADED USING THE SETUP UTILITY (SEE CHAP. 6 OF THE XXDP+ USER'S MANUAL). WHEN YOU ANSWER THIS QUESTION WITH A 'Y' THE RUNTIME SERVICES WILL ASK FOR THE NUMBER OF UNITS (IN DECIMAL). YOU WILL THEN BE ASKED THE FOLLOWING QUESTIONS FOR EACH UNIT:

2.4.1 QUESTIONS AND ANSWERS

1. CSR ADDRESS (172000) 0?

   ANSWER: THE CSR ADDRESS OF THE RH CONTROLLER (OCTAL, DEFAULT 172000).

2. INTERRUPT VECTOR (204) 0?

   ANSWER: THE INTERRUPT VECTOR ADDRESS (OCTAL, DEFAULT 204).

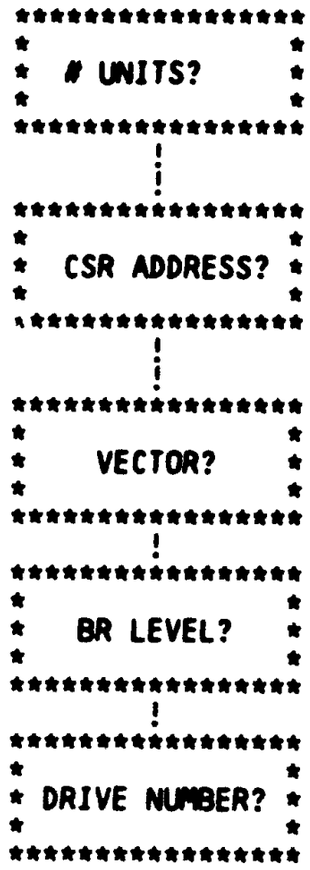
3. BR LEVEL FOR INTERRUPT (5) 0?

   ANSWER: THE BUS REQUEST LEVEL FOR INTERRUPT (OCTAL, DEFAULT 5).

4. DRIVE NUMBER (0) 0?

   ANSWER: THE PHYSICAL DRIVE NUMBER(S) OF THE DRIVE(S) TO BE TESTED (OCTAL, {0-7}, DEFAULT 0).

2.4.2 SUMMARY OF HARDWARE QUESTION SEQUENCE



2.5 SOFTWARE QUESTIONS

AFTER YOU HAVE ANSWERED THE HARDWARE QUESTIONS, OR AFTER A RESTART OR OR CONTINUE COMMAND, THE RUNTIME SERVICES WILL ASK FOR SOFTWARE PARAMETERS. THESE PARAMETERS WILL GOVERN SOME DIAGNOSTIC SPECIFIC OPERATION MODES. YOU WILL BE PROMPTED BY "CHANGE SW (L) ?" IF YOU WISH TO CHANGE ANY PARAMETERS, ANSWER BY TYPING "Y". THE SOFTWARE QUESTIONS AND THE DEFAULT VALUES ARE DESCRIBED IN THE NEXT PARAGRAPH(S).

INCLUDED IN THIS SECTION ARE QUESTIONS ABOUT PARAMETERS WHICH AFFECT PROGRAM OPERATION. REQUESTS FOR SOFTWARE OPTIONS SHOULD BE DISCUSSED AND INCORPORATED INTO THIS SPECIFICATION AS SOON AS POSSIBLE, AND CERTAINLY BEFORE SIGNOFF TIME.

IF A ^Z (CONTROL-Z) IS TYPED AS THE ANSWER TO ANY QUESTION, THE PROGRAM WILL START TO EXECUTE.

IF A ^C (CONTROL-C) IS TYPED AT ANY TIME, CONTROL WILL RETURN TO THE DIAGNOSTIC SUPERVISOR.

2.5.1 QUESTIONS AND ANSWERS

1. LIMIT RANGE OF SECTORS TO BE TESTED (N) L?

   ANSWER: TO TEST ONLY A CERTAIN RANGE OF SECTORS INSTEAD OF AN ENTIRE UNIT. ALTHOUGH THIS QUESTION IS NOT OPTIONAL, IT SHOULD BE ANSWERED 'NO' UNLESS THE ANSWER TO THE HARDWARE QUESTION '# UNITS ?' WAS 1.
   (LOGICAL, {Y TO TEST PARTIAL UNIT, N FOR ENTIRE UNIT}, DEFAULT N).

THE QUESTIONS WHICH FOLLOW ARE OPTIONAL, AND DEPEND ON AN AFFIRMATIVE ANSWER TO THE PREVIOUS QUESTION:

2. DO YOU KNOW THE EXACT RANGE OF SECTORS TO BE TESTED (N) L?

   ANSWER: THIS DECIDES WHETHER TO ASK FOR SECTOR NUMBERS OR BOARD NUMBER.

THE NEXT TWO QUESTIONS WILL BE ASKED ONLY IF QUESTION 2 IS ANSWERED YES:

3. FIRST SECTOR TO TEST (0) O?

   ANSWER: THE SECTOR NUMBER WHERE TESTING BEGINS
   (OCTAL, {0-7777 FOR ML11A, 0-37777 FOR ML11B},
   DEFAULT 0).

4. LAST SECTOR TO TEST (LAST) O?

   ANSWER: THE LAST SECTOR NUMBER TO BE TESTED
   (OCTAL, {0-7777 FOR ML11A, 0-37777 FOR ML11B},
   DEFAULT 37777).

THE NEXT QUESTION WILL BE ASKED ONLY IF QUESTION 2 IS ANSWERED NO:

5. WHICH BOARD {1-16} SHOULD BE TESTED (1) D?

   ANSWER: THE NUMBER OF THE ONLY BOARD TO BE TESTED
   (DECIMAL, {1-16}, DEFAULT 1).

6. THE PROGRAM OPTIONS INCLUDE:

   1. ADDRESS CHECK
   2. PATTERN TEST

3. UNIQUE DATA CHECK

4. MARCH TEST

5. RANDOMNESS TESTS

DO YOU WANT TO DROP ANY OF THESE FROM THE EXERCISER (N) L?

ANSWER: THIS DECIDES WHETHER TO ASK FOR OPTION NUMBERS.
(LOGICAL, {Y FOR PARTIAL EXERCISER, N FOR COMPLETE EXERCISER}, DEFAULT N).

THE FOLLOWING SET OF QUESTIONS WILL BE ASKED ONLY IF QUESTION 6 IS ANSWERED YES:

7. DROP OPTION 1 (N) L?

8. DROP OPTION 2 (N) L?

9. DROP OPTION 3 (N) L?

10. DROP OPTION 4 (N) L?

11. DROP OPTION 5 (N) L?

12. ENABLE REFRESH MARGINING (N) L?

ANSWER: NORMAL OPERATION IS WITHOUT REFRESH MARGINING.
(LOGICAL, {Y TO ENABLE, N TO DISABLE}, DEFAULT N).

13. DISABLE ERROR CORRECTION (N) L?

ANSWER: NORMAL OPERATION IS WITH ECC ENABLED.
(LOGICAL, {Y TO DISABLE, N TO ENABLE}, DEFAULT N).

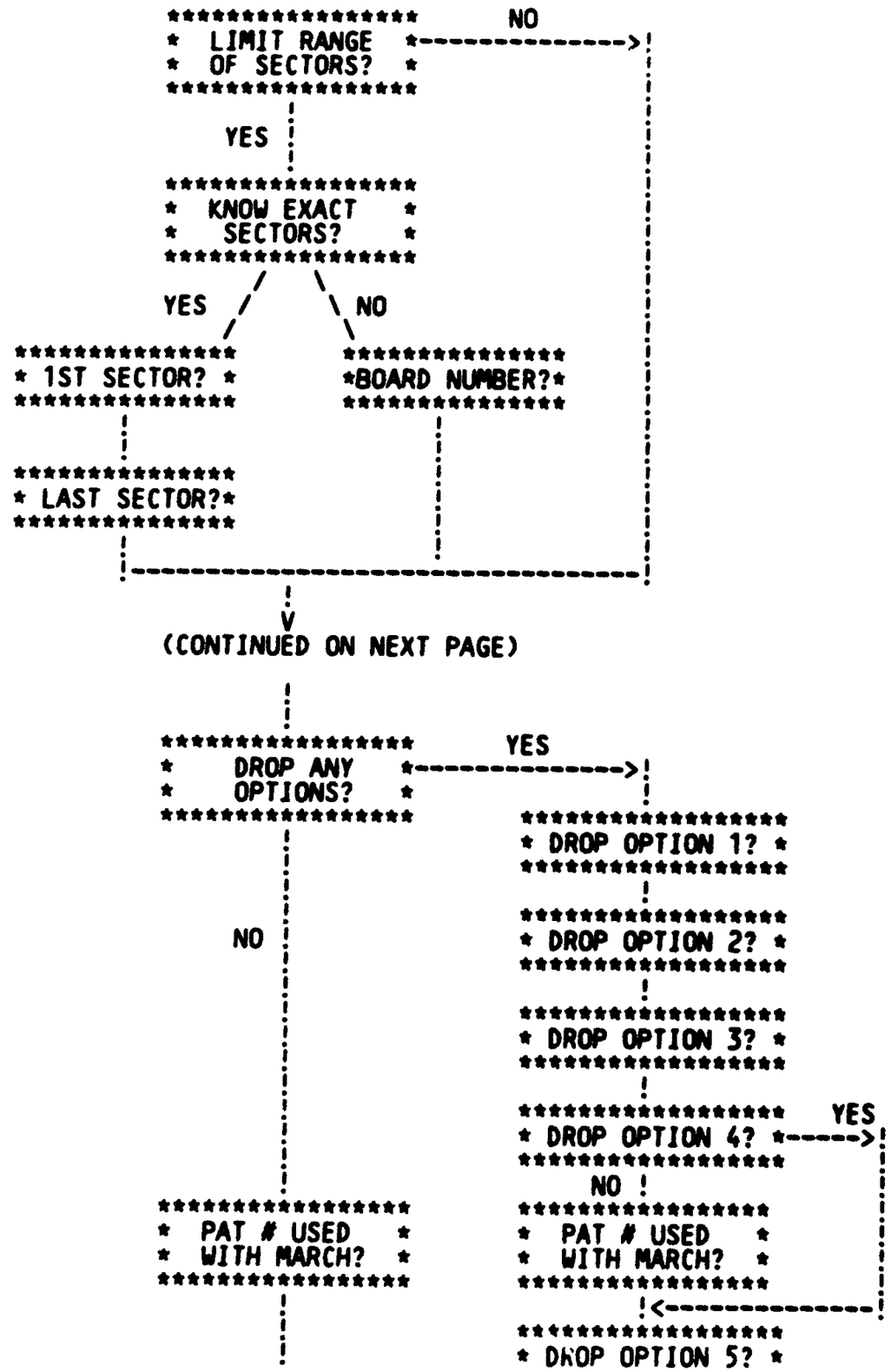
14. ENABLE END OF PASS SUMMARY PRINTOUT (N) L?

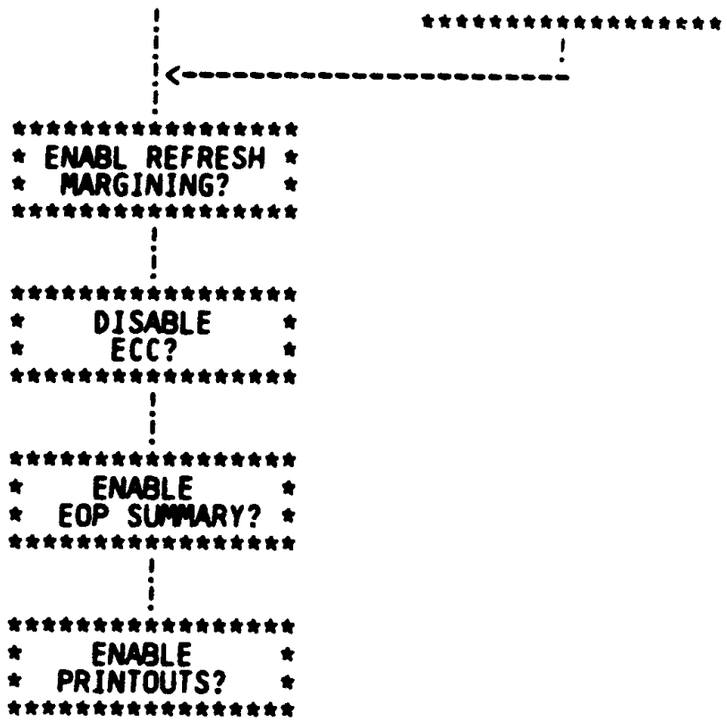
ANSWER: NORMAL OPERATION IS WITHOUT END OF PASS SUMMARY PRINTOUT.
(LOGICAL, {Y TO ENABLE, N TO DISABLE PRINTOUT}, DEFAULT N).

15. ENABLE ERROR PRINTOUTS (N) L?

ANSWER: NORMAL OPERATION IS WITHOUT ERROR PRINTOUTS.
(LOGICAL, {Y TO ENABLE, N TO DISABLE PRINTOUT}, DEFAULT N).

2.5.2 SUMMARY OF SOFTWARE QUESTION SEQUENCE


LIMIT RANGE OF SECTORS?
NO
YES
KNOW EXACT SECTORS?
YES
NO
1ST SECTOR?
LAST SECTOR?
BOARD NUMBER?
(CONTINUED ON NEXT PAGE)
DROP ANY OPTIONS?
YES
NO
DROP OPTION 1?
DROP OPTION 2?
DROP OPTION 3?
DROP OPTION 4?
YES
NO
DROP OPTION 5?
PAT # USED WITH MARCH?
PAT # USED WITH MARCH?



2.6 EXTENDED P-TABLE DIALOGUE

WHEN YOU ANSWER THE HARDWARE QUESTIONS, YOU ARE BUILDING ENTRIES IN A TABLE THAT DESCRIBES THE DEVICES UNDER TEST. THE SIMPLEST WAY TO BUILD THIS TABLE IS TO ANSWER ALL QUESTIONS FOR EACH UNIT TO BE TESTED. IF YOU HAVE A MULTIPLEXED DEVICE, SUCH AS A MASS STORAGE CONTROLLER WITH SEVERAL DRIVES OR A COMMUNICATION DEVICE WITH SEVERAL LINES, THE DIALOGUE BECOMES TEDIOUS BECAUSE MOST OF THE ANSWERS ARE REPETITIOUS.

SUPPOSE YOU ARE TESTING A HYPOTHETICAL DEVICE, THE XY11 WHICH IS MADE UP OF ONE CONTROL MODULE WITH 8 UNITS (SUB-DEVICES). THESE 8 UNITS, NUMBERED 0 THROUGH 7, HAVE JUST ONE HARDWARE PARAMETER THAT CAN VARY AMONG UNITS, CALLED THE Q-FACTOR.

THE 3 EXAMPLES WHICH FOLLOW SHOW DIFFERENT WAYS OF ANSWERING THE HARDWARE QUESTIONS TO ACHIEVE IDENTICAL RESULTS. IN EACH EXAMPLE, THERE ARE TO BE 8 XY11'S TESTED WHICH HAVE THE FOLLOWING Q-FACTORS:

UNIT #	Q-FACTOR
0	0
1	1
2	0
3	0
4	0
5	0
6	1
7	1

EXAMPLE 1: ANSWER SEPARATELY FOR EACH UNIT

```
# UNITS (D) ?  8<CR>

UNIT 0
CSR ADDRESS (O) ?  160000<CR>
SUB-DEVICE # (O) ?  0<CR>
Q-FACTOR (O) 0 ?  0<CR>

UNIT 1
CSR ADDRESS (O) ?  160000<CR>
SUB-DEVICE # (O) ?  1<CR>
Q-FACTOR (O) 0 ?  1<CR>

UNIT 2
CSR ADDRESS (O) ?  160000<CR>
SUB-DEVICE # (O) ?  2<CR>
Q-FACTOR (O) 1 ? 0<CR>

UNIT 3
CSR ADDRESS (O) ?  160000<CR>
SUB-DEVICE # (O) ?  3<CR>
Q-FACTOR (O) 0 ?  <CR>

UNIT 4
CSR ADDRESS (O) ?  160000<CR>
SUB-DEVICE # (O) ?  4<CR>
Q-FACTOR (O) 0 ?  <CR>

UNIT 5
CSR ADDRESS (O) ?  160000<CR>
SUB-DEVICE # (O) ?  5<CR>
Q-FACTOR (O) 0 ? 1<CR>

UNIT 6
CSR ADDRESS (O) ?  160000<CR>
SUB-DEVICE # (O) ?  6<CR>
Q-FACTOR (O) 1 ? 1<CR>

UNIT 7
CSR ADDRESS (O) ?  160000<CR>
SUB-DEVICE # (O) ?  7<CR>
Q-FACTOR (O) 1 ? <CR>
```

NOTICE THAT THE DEFAULT VALUE FOR THE Q-FACTOR CHANGES WHEN A NON-DEFAULT RESPONSE IS GIVEN. BE CAREFUL WHEN SPECIFYING MULTIPLE UNITS!

AS YOU CAN SEE, THE HARDWARE PARAMETERS DO NOT DIFFER SIGNIFICANTLY FROM UNIT TO UNIT. THE PROCEDURE SHOWN IS NOT VERY EFFICIENT.

EXAMPLE 2: USE THE MULTIPLE SPECIFICATION FEATURE OF THE RUNTIME SERVICES TO MAKE FEWER PASSES THROUGH THE QUESTIONS.

```
# UNITS (D) ?  8<CR>

UNIT 0
CSR ADDRESS (O) ?  160000<CR>
SUB-DEVICE # (O) ? 0,1<CR>
Q-FACTOR (O) 0 ?  0,1<CR>

UNIT 2
CSR ADDRESS (O) ?  160000<CR>
SUB-DEVICE # (O) ?  2-5<CR>
Q-FACTOR (O) 1 ?  0<CR>

UNIT 6
CSR ADDRESS (O) ?  160000<CR>
SUB-DEVICE # (O) ?  6,7<CR>
Q-FACTOR (O) 0 ?  1,1<CR>
```

AS YOU CAN SEE IN THE ABOVE DIALOGUE, RUNTIME SERVICES WILL BUILD AS MANY ENTRIES AS IT CAN WITH THE INFORMATION GIVEN IN ANY ONE PASS THROUGH THE QUESTIONS. THE FIRST PASS BUILDS 2 ENTRIES, BECAUSE 2 SUB-DEVICES AND Q-FACTORS WERE SPECIFIED. THE DRS ASSUMES THAT THE CSR ADDRESS IS 160000 FOR BOTH UNITS. IN THE SECOND PASS, 4 ENTRIES WERE BUILT. THE "-" CONSTRUCT TELLS THE DRS TO INCREMENT THE DATA FROM THE FIRST VALUE TO THE SECOND. IN THIS CASE, "2-5" SPECIFIES SUB-DEVICES 2,3,4, AND 5. THE CSR ADDRESS AND Q-FACTOR FOR THE 4 ENTRIES ARE ASSUMED TO BE 160000 AND 0 RESPECTIVELY SINCE THEY WERE ONLY SPECIFIED ONCE. THE LAST 2 UNITS ARE SPECIFIED IN THE THIRD PASS.

EXAMPLE 3: ACCOMPLISH THE WHOLE PROCESS IN JUST ONE PASS THROUGH THE QUESTIONS.

```
# UNITS  (D) ?  8<CR>

UNIT 1
CSR ADDRESS (O) ?  160000<CR>
SUB-DEVICE # (O) ? 0-7<CR>
Q-FACTOR (O) 0 ? 0,1,0,,,,1,1<CR>
```

AS YOU CAN SEE FROM THIS EXAMPLE, NULL REPLIES (COMMAS ENCLOSING A BLANK FIELD) DIRECT THE DRS TO REPEAT THE LAST REPLY.

2.7 QUICK START-UP PROCEDURE

2.7.1 TO START-UP THIS PROGRAM WHEN RUNNING UNDER XXDP+

1. BOOT XXDP+
2. GIVE THE DATE AND ANSWER THE LSI AND 50HZ QUESTIONS
3. TYPE "R NAME", WHERE "NAME" IS THE FILENAME OF THE .BIN OR .BIC FILE FOR THIS PROGRAM
4. TYPE "START"
5. ANSWER THE "CHANGE HW" QUESTION WITH "Y"
6. ANSWER ALL THE HARDWARE QUESTIONS
7. ANSWER THE "CHANGE SW" QUESTION WITH "N"

WHEN YOU FOLLOW THIS PROCEDURE YOU WILL BE USING ONLY THE DEFAULTS FOR FLAGS AND SOFTWARE PARAMETERS. THE DEFAULTS ARE DESCRIBED IN SECTIONS 2.3 AND 2.5.

2.7.2 TO START-UP THIS PROGRAM WHEN RUNNING UNDER ACT

THE FILE WHICH WILL BE STARTED MUST, AT SOME TIME, HAVE BEEN CREATED WITH THE 'SETUP' UTILITY PROGRAM. ASSUMING THAT THIS HAS BEEN DONE, THE START-UP PROCEDURE IS THE SAME AS THE ONE IN SECTION 2.7.1.

TO CREATE A FILE USING THE SETUP UTILITY:

1. TYPE "R SETUP"
2. THE TARGET ENVIRONMENT IS ACT, SO TYPE "AC"
3. TYPE THE SETUP COMMAND:
   *SETUP OUTFILE.EXT=INFILE.EXT
   WHERE OUTFILE.EXT = THE NEWLY CREATED FILE
   AND INFILE.EXT = THE RELEASED .BIN FILE
4. YOU WILL HAVE AN OPPORTUNITY TO SET UP A PERMANENT DEFAULT TEST CONFIGURATION BY ANSWERING THE HARDWARE AND SOFTWARE QUESTIONS. ONCE THIS IS DONE, YOU WILL NO LONGER BE FORCED TO ANSWER "Y" TO "CHANGE HW ?" EVERY TIME YOU START THE PROGRAM. YOU WILL ALSO BE ASKED THE LSI AND 50HZ QUESTIONS HERE.
5. TYPE "EXIT"

3.0 ERROR INFORMATION

3.1 TYPES OF ERROR MESSAGES

THERE ARE THREE LEVELS OF ERROR MESSAGES THAT MAY BE ISSUED BY A DIAGNOSTIC: GENERAL, BASIC AND EXTENDED. GENERAL ERROR MESSAGES ARE ALWAYS PRINTED UNLESS THE "IER" FLAG IS SET (SECTION 2.3). THE GENERAL ERROR MESSAGE IS OF THE FORM:

```
NAME TYPE NUMBER ON UNIT NUMBER  TST NUMBER PC:XXXXXX
ERROR MESSAGE
```

WHERE:

```
NAME = DIAGNOSTIC NAME
TYPE = ERROR TYPE (SYS FATAL, DEV FATAL, HARD OR SOFT)
NUMBER = ERROR NUMBER
UNIT NUMBER = 0 - N (N IS LAST UNIT IN PTABLE)
TST NUMBER = TEST AND SUBTEST WHERE ERROR OCCURRED
PC:XXXXXX = ADDRESS OF ERROR MESSAGE CALL
```

BASIC ERROR MESSAGES ARE MESSAGES THAT CONTAIN SOME ADDITIONAL INFORMATION ABOUT THE ERROR. THESE ARE ALWAYS PRINTED UNLESS THE "IER" OR "IBR" FLAGS ARE SET (SECTION 2.3). THESE MESSAGES ARE PRINTED AFTER THE ASSOCIATED GENERAL MESSAGE.

EXTENDED ERROR MESSAGES CONTAIN SUPPLEMENTARY ERROR INFORMATION SUCH AS REGISTER CONTENTS OR GOOD/BAD DATA. THESE ARE ALWAYS PRINTED UNLESS THE "IER", "IBR" OR "IXR" FLAGS ARE SET (SECTION 2.3). THESE MESSAGES ARE PRINTED AFTER THE ASSOCIATED GENERAL ERROR MESSAGE AND ANY ASSOCIATED BASIC ERROR MESSAGES.

3.2 SPECIFIC ERROR MESSAGES

3.2.1 MESSAGES DURING INITIALIZATION

DURING INITIALIZATION, THERE ARE 3 ERROR CONDITIONS WHICH CAN BE DETECTED AND WHICH CAUSE THE DRIVE TO BE DROPPED:

```
CAUSE1  =  '  (NOT POWERED UP)'
CAUSE2  =  '  (NOT AN ML11 UNIT)'
CAUSE3  =  '  (OPERATOR SELECTED TEST LIMITS INCORRECTLY)'
```

```
THE 'CAUSE3' MESSAGE IS THEN FOLLOWED BY
EITHER: !'TOP SECTOR OF XXXXXX EXCEEDS SYSTEM LIMIT OF YYYYYY'
  OR:   !'LOW SECTOR OF XXXXXX EXCEEDS TOP SECTOR OF YYYYYY'
```

3.2.2 MESSAGES DURING TESTING

AFTER INITIALIZATION, THERE ARE 4 ERROR DIAGNOSES WHICH ARE POSSIBLE:

```
MSG1    =  '--> RUN ML11 LOGIC TEST'

           THIS MESSAGE APPEARS WHEN AN ML11
```

SYSTEM ERROR OCCURS. THE USER IS INSTRUCTED TO RUN THE LOGIC TEST TO SEE IF THAT PROGRAM CAN ISOLATE THE REASON FOR THE FAILURE. NOTE THAT THIS MESSAGE IS NOT USED FOR DATA ERRORS.

MSG2 = '--> RUN ML11 PROM MAINTENANCE PROGRAM'

THIS MESSAGE APPEARS WHENEVER ANY OF THE FOLLOWING CONDITIONS IS TRUE:

A) AN ECC HARD ERROR (ECH OR UNC) IS DETECTED

B) AN ARRAY REACHES ITS HARD OR SOFT ERROR THRESHOLD

C) DURING A PERFORMANCE SUMMARY, IF AN ARRAY HAS REACHED OR EXCEEDED ITS HARD OR SOFT ERROR THRESHOLD.

MSG3 = 'SOFT ERROR'

A SOFT ERROR IS A CORRECTABLE DATA ERROR WHICH CAN BE ELIMINATED BY REWRITING AND REREADING.

MSG4 = 'HARD ERROR'

A HARD ERROR IS A CORRECTABLE DATA ERROR WHICH PERSISTS AFTER A REWRITE AND A REREAD.

ANY WRITE COMMAND HAS 3 POSSIBLE ERROR CALLS ASSOCIATED WITH IT:

ERROR POSITION	ERROR TYPE	DIAGNOSTIC MESSAGE	CAUSE OF ERROR/ACTION
1ST	ERRDF	MSG1	ALL 6 RETRIES FAILED FOR AN ML11 SYSTEM ERROR WHICH WAS ORIGINALLY CONSIDERED TO BE NON-FATAL. DROP THE DRIVE.
2ND	ERRDF	MSG1	CONTROLLER FATAL ERROR. DROP THE DRIVE.
3RD	ERRDF	MSG1	DRIVE FATAL ERROR. DROP THE DRIVE.

ANY READ OR WRITE CHECK COMMAND HAS 8 ERROR CALLS ASSOCIATED WITH IT, 5 FOR SYSTEM ERRORS (WHICH RESULT IN DROPPING THE DRIVE) AND 3 FOR HARD AND SOFT DATA ERRORS (WHICH ARE COUNTED ON A PER ARRAY BASIS):

THE SYSTEM ERRORS:

ERROR	ERROR	DIAGNOSTIC

POSITION	TYPE	MESSAGE	CAUSE OF ERROR/ACTION
1ST	ERRDF	MSG1	AFTER A SUCCESSFUL READ COMMAND, THE WRITE AND READ BUFFERS ARE COMPARED. IF THEY DO NOT MATCH PERFECTLY, THEN THE ECC LOGIC FAILED. DROP THE DRIVE.
2ND	ERRDF	MSG1	ALL 6 RETRIES FAILED FOR AN ML11 SYSTEM ERROR WHICH WAS ORIGINALLY CONSIDERED TO BE NON-FATAL. DROP THE DRIVE.
3RD	ERRDF	MSG1	CONTROLLER FATAL ERROR. DROP THE DRIVE.
4TH	ERRDF	MSG1	DRIVE FATAL ERROR. DROP THE DRIVE.
5TH	ERRDF	MSG2	ECC HARD ERROR DETECTED. DROP THE DRIVE.

THE DATA ERRORS:

ERROR POSITION	ERROR TYPE	DIAGNOSTIC MESSAGE	CAUSE OF ERROR/ACTION
6TH	ERRHRD	MSG4	HARD ERROR. DURING ERROR CLASSIFICATION, RETRY FAILED AGAIN (AS IF A DATA ERROR IN THE SAME BIT POSITION). UPDATE THE HARD COUNT FOR THE ASSOCIATED ARRAY AND CHECK THE NEW COUNT AGAINST THE THRESHOLD.
7TH	ERRSOFT	MSG3	SOFT ERROR. DURING ERROR CLASSIFICATION, RETRY FAILED AGAIN (AS IF A DATA ERROR IN A DIFFERENT BIT POSITION). UPDATE THE SOFT COUNT FOR THE ASSOCIATED ARRAY AND CHECK THE NEW COUNT AGAINST THE THRESHOLD.
8TH	ERRSOFT	MSG3	SOFT ERROR. DURING ERROR CLASSIFICATION, RETRY DID NOT PRODUCE ANOTHER DATA ERROR (THE RETRY MAY PASS, OR IT MAY FAIL FOR SOME NEW REASON). UPDATE THE SOFT COUNT FOR THE ASSOCIATED ARRAY AND CHECK THE NEW COUNT AGAINST THE THRESHOLD.

THERE IS A CORRESPONDENCE BETWEEN EVERY NUMBER AND THE PLACE IN THE PROGRAM WHERE THE ERROR WAS DETECTED:

A) THE MOST SIGNIFICANT DIGIT IDENTIFIES THE OPTION NUMBER.

EXCEPTION: THE COMMAND INTEGRITY ROUTINE IS THOUGHT OF AS OPTION 0, AND SO ALL ITS ERROR NUMBERS WILL APPEAR TO ONLY BE 1 OR 2 SIGNIFICANT DIGITS. TAKE THE 0 TO BE A SIGNIFICANT DIGIT (SEE EXAMPLES).

B) THE 2 LEAST SIGNIFICANT DIGITS ARE THE POSITIONAL NUMBERS TO IDENTIFY WHERE THE ERROR CALL IS LOCATED WITHIN THE OPTION.

C) FOR OPTION 5, THE SECOND MOST SIGNIFICANT DIGIT IS USED TO IDENTIFY WHICH RANDOM TEST IS RUNNING.

EXAMPLES:

1) ERROR NUMBER: 6 => COMMAND INTEGRITY ROUTINE (OPTION 0), ERROR 06

2) ERROR NUMBER: 208 => OPTION 2, ERROR 08

3) ERROR NUMBER: 5311 => OPTION 5, RAND3, ERROR 11

BELOW IS A SUMMARY OF THE ACTUAL ERROR CALLS IN EACH TEST OPTION:

```
!
! THE 'INTEGRITY' ROUTINE:
!
                    !WRITE COMMAND:

!ERRDF(1,MSG1,0);    !**** INTEGRITY ROUTINE ERROR 01 ****
!ERRDF(2,MSG1,0);    !**** INTEGRITY ROUTINE ERROR 02 ****
!ERRDF(3,MSG1,0);    !**** INTEGRITY ROUTINE ERROR 03 ****

                    !READ COMMAND:

!ERRDF(4,MSG1,0);    !**** INTEGRITY ROUTINE ERROR 04 ****
!ERRDF(5,MSG1,0);    !**** INTEGRITY ROUTINE ERROR 05 ****
!ERRDF(6,MSG1,0);    !**** INTEGRITY ROUTINE ERROR 06 ****
!ERRDF(7,MSG1,0);    !**** INTEGRITY ROUTINE ERROR 07 ****
!ERRDF(8,MSG2,0);    !**** INTEGRITY ROUTINE ERROR 08 ****
!ERRHRD(9,MSG4,0);   !**** INTEGRITY ROUTINE ERROR 09 ****
!ERRSOFT(10,MSG3,0);!**** INTEGRITY ROUTINE ERROR 10 ****
!ERRSOFT(11,MSG3,1);!**** INTEGRITY ROUTINE ERROR 10 ****

                    !WRITE CHECK COMMAND:

!ERRDF(12,MSG1,0);   !**** INTEGRITY ROUTINE ERROR 12 ****
!ERRDF(13,MSG1,0);   !**** INTEGRITY ROUTINE ERROR 13 ****
!ERRDF(14,MSG1,0);   !**** INTEGRITY ROUTINE ERROR 14 ****
!ERRDF(15,MSG2,0);   !**** INTEGRITY ROUTINE ERROR 15 ****
!ERRHRD(16,MSG4,0);  !**** INTEGRITY ROUTINE ERROR 16 ****
!ERRSOFT(17,MSG3,8);!**** INTEGRITY ROUTINE ERROR 17 ****
!ERRSOFT(18,MSG3,1);!**** INTEGRITY ROUTINE ERROR 18 ****

!
! OPTION 1:
!
                    !WRITE COMMAND:

!ERRDF(101,MSG1,0);   !**** OPTION 1 ERROR 01 ****
!ERRDF(102,MSG1,0);   !**** OPTION 1 ERROR 02 ****
!ERRDF(103,MSG1,0);   !**** OPTION 1 ERROR 03 ****

                    !CHECK OR READ:

!ERRDF(104,MSG1,0);   !**** OPTION 1 ERROR 04 ****
```

```
!ERRDF(105,MSG1,0);    !**** OPTION 1 ERROR 05 ****
!ERRDF(106,MSG1,0);    !**** OPTION 1 ERROR 06 ****
!ERRDF(107,MSG1,0);    !**** OPTION 1 ERROR 07 ****
!ERRDF(108,MSG2,0);    !**** OPTION 1 ERROR 08 ****
!ERRHRD(109,MSG4,0);   !**** OPTION 1 ERROR 09 ****
!ERRSOFT(110,MSG3,0);  !**** OPTION 1 ERROR 10 ****
!ERRSOFT(111,MSG3,0);  !**** OPTION 1 ERROR 11 ****

!
! OPTION 2:
!
                    !WRITE COMMAND

!ERRDF(201,MSG1,0);    !**** OPTION 2 ERROR 01 ****
!ERRDF(202,MSG1,0);    !**** OPTION 2 ERROR 02 ****
!ERRDF(203,MSG1,0);    !**** OPTION 2 ERROR 03 ****

                    !CHECK OR READ:

!ERRDF(204,MSG1,0);    !**** OPTION 2 ERROR 04 ****
!ERRDF(205,MSG1,0);    !**** OPTION 2 ERROR 05 ****
!ERRDF(206,MSG1,0);    !**** OPTION 2 ERROR 06 ****
!ERRDF(207,MSG1,0);    !**** OPTION 2 ERROR 07 ****
!ERRDF(208,MSG2,0);    !**** OPTION 2 ERROR 08 ****
!ERRHRD(209,MSG4,0);   !**** OPTION 2 ERROR 09 ****
!ERRSOFT(210,MSG3,0);  !**** OPTION 2 ERROR 10 ****
!ERRSOFT(211,MSG3,1);  !**** OPTION 2 ERROR 10 ****

                    !LOOP CHECK OR READ:

!ERRDF(212,MSG1,0);    !**** OPTION 2 ERROR 12 ****
!ERRDF(213,MSG1,0);    !**** OPTION 2 ERROR 13 ****
!ERRDF(214,MSG1,0);    !**** OPTION 2 ERROR 14 ****
!ERRDF(215,MSG1,0);    !**** OPTION 2 ERROR 15 ****
!ERRDF(216,MSG2,0);    !**** OPTION 2 ERROR 16 ****
!ERRHRD(217,MSG4,0);   !**** OPTION 2 ERROR 17 ****
!ERRSOFT(218,MSG3,0);  !**** OPTION 2 ERROR 18 ****
!ERRSOFT(219,MSG3,0);  !**** OPTION 2 ERROR 19 ****

!
! OPTION 3:
!

                    !WRITE COMMAND:

!ERRDF(301,MSG1,0);    !**** OPTION 3 ERROR 01 ****
!ERRDF(302,MSG1,0);    !**** OPTION 3 ERROR 02 ****
!ERRDF(303,MSG1,0);    !**** OPTION 3 ERROR 03 ****

                    !CHECK OR READ:

!ERRDF(304,MSG1,0);    !**** OPTION 3 ERROR 04 ****
!ERRDF(305,MSG1,0);    !**** OPTION 3 ERROR 05 ****
!ERRDF(306,MSG1,0);    !**** OPTION 3 ERROR 06 ****
!ERRDF(307,MSG1,0);    !**** OPTION 3 ERROR 07 ****
!ERRDF(308,MSG2,0);    !**** OPTION 3 ERROR 08 ****
!ERRHRD(309,MSG4,0);   !**** OPTION 3 ERROR 09 ****
```

```
!ERRSOFT(310,MSG3,0); !**** OPTION 3 ERROR 10 ****
!ERRSOFT(311,MSG3,0); !**** OPTION 3 ERROR 11 ****

!
! OPTION 4:
!

        !MARCHING UP:

                !WRITE DATA:

!ERRDF(401,MSG1,0);    !**** OPTION 4 ERROR 01 ****
!ERRDF(402,MSG1,0);    !**** OPTION 4 ERROR 02 ****
!ERRDF(403,MSG1,0);    !**** OPTION 4 ERROR 03 ****

        !MARCHING UP:

                !CHECK OR READ DATA:

!ERRDF(404,MSG1,0);    !**** OPTION 4 ERROR 04 ****
!ERRDF(405,MSG1,0);    !**** OPTION 4 ERROR 05 ****
!ERRDF(406,MSG1,0);    !**** OPTION 4 ERROR 06 ****
!ERRDF(407,MSG1,0);    !**** OPTION 4 ERROR 07 ****
!ERRDF(408,MSG2,0);    !**** OPTION 4 ERROR 08 ****
!ERRHRD(409,MSG4,0);   !**** OPTION 4 ERROR 09 ****
!ERRSOFT(410,MSG3,0); !**** OPTION 4 ERROR 10 ****
!ERRSOFT(411,MSG3,0); !**** OPTION 4 ERROR 11 ****

                !WRITE COMP:

!ERRDF(412,MSG1,0);    !**** OPTION 4 ERROR 12 ****
!ERRDF(413,MSG1,0);    !**** OPTION 4 ERROR 13 ****
!ERRDF(414,MSG1,0);    !**** OPTION 4 ERROR 14 ****

        !MARCHING DOWN:

                !CHECK OR READ COMP:

!ERRDF(415,MSG1,0);    !**** OPTION 4 ERROR 15 ****
!ERRDF(416,MSG1,0);    !**** OPTION 4 ERROR 16 ****
!ERRDF(417,MSG1,0);    !**** OPTION 4 ERROR 17 ****
!ERRDF(418,MSG1,0);    !**** OPTION 4 ERROR 18 ****
!ERRDF(419,MSG2,0);    !**** OPTION 4 ERROR 19 ****
!ERRHRD(420,MSG4,0);   !**** OPTION 4 ERROR 20 ****
!ERRSOFT(421,MSG3,0); !**** OPTION 4 ERROR 21 ****
!ERRSOFT(422,MSG3,0); !**** OPTION 4 ERROR 22 ****

                !WRITE DATA:

!ERRDF(423,MSG1,0);    !**** OPTION 4 ERROR 23 ****
!ERRDF(424,MSG1,0);    !**** OPTION 4 ERROR 24 ****
!ERRDF(425,MSG1,0);    !**** OPTION 4 ERROR 25 ****

        !MARCHING UP:

                !CHECK OR READ DATA:
```

```
                                !ERRDF(426,MSG1,0);    !**** OPTION 4 ERROR 26 ****
                                !ERRDF(427,MSG1,0);    !**** OPTION 4 ERROR 27 ****
                                !ERRDF(428,MSG1,0);    !**** OPTION 4 ERROR 28 ****
                                !ERRDF(429,MSG1,0);    !**** OPTION 4 ERROR 29 ****
                                !ERRDF(430,MSG2,0);    !**** OPTION 4 ERROR 30 ****
                                !ERRHRD(431,MSG4,0);   !**** OPTION 4 ERROR 31 ****
                                !ERRSOFT(432,MSG3,0);  !**** OPTION 4 ERROR 32 ****
                                !ERRSOFT(433,MSG3,0);  !**** OPTION 4 ERROR 33 ****


                                !
                                ! OPTION 5:
                                !

                                        !'RAND1' ROUTINE:

                                                !WRITE COMMAND:

                                !ERRDF(5101,MSG1,0);   !**** OPTION 5, RAND1 ERROR 01 ****
                                !ERRDF(5102,MSG1,0);   !**** OPTION 5, RAND1 ERROR 02 ****
                                !ERRDF(5103,MSG1,0);   !**** OPTION 5, RAND1 ERROR 03 ****

                                                !CHECK OR READ:

                                !ERRDF(5104,MSG1,0);   !**** OPTION 5, RAND1 ERROR 04 ****
                                !ERRDF(5105,MSG1,0);   !**** OPTION 5, RAND1 ERROR 05 ****
                                !ERRDF(5106,MSG1,0);   !**** OPTION 5, RAND1 ERROR 06 ****
                                !ERRDF(5107,MSG1,0);   !**** OPTION 5, RAND1 ERROR 07 ****
                                !ERRDF(5108,MSG2,0);   !**** OPTION 5, RAND1 ERROR 08 ****
                                !ERRHRD(5109,MSG4,0);  !**** OPTION 5, RAND1 ERROR 09 ****
                                !ERRSOFT(5110,MSG3,0); !**** OPTION 5, RAND1 ERROR 10 ****
                                !ERRSOFT(5111,MSG3,0); !**** OPTION 5, RAND1 ERROR 11 ****

                                        !'RAND2' ROUTINE:

                                                !WRITE COMMAND:

                                !ERRDF(5201,MSG1,0);   !**** OPTION 5, RAND2 ERROR 01 ****
                                !ERRDF(5202,MSG1,0);   !**** OPTION 5, RAND2 ERROR 02 ****
                                !ERRDF(5203,MSG1,0);   !**** OPTION 5, RAND2 ERROR 03 ****

                                                !CHECK OR READ:

                                !ERRDF(5204,MSG1,0);   !**** OPTION 5, RAND2 ERROR 04 ****
                                !ERRDF(5205,MSG1,0);   !**** OPTION 5, RAND2 ERROR 05 ****
                                !ERRDF(5206,MSG1,0);   !**** OPTION 5, RAND2 ERROR 06 ****
                                !ERRDF(5207,MSG1,0);   !**** OPTION 5, RAND2 ERROR 07 ****
                                !ERRDF(5208,MSG2,0);   !**** OPTION 5, RAND2 ERROR 08 ****
                                !ERRHRD(5209,MSG4,0);  !**** OPTION 5, RAND2 ERROR 09 ****
                                !ERRSOFT(5210,MSG3,0); !**** OPTION 5, RAND2 ERROR 10 ****
                                !ERRSOFT(5211,MSG3,0); !**** OPTION 5, RAND2 ERROR 11 ****

                                        !'RAND3' ROUTINE:

                                                !WRITE COMMAND:

                                !ERRDF(5301,MSG1,0);   !**** OPTION 5, RAND3 ERROR 01 ****
```

```
!ERRDF(5302,MSG1,0);    !**** OPTION 5, RAND3 ERROR 02 ****
!ERRDF(5303,MSG1,0);    !**** OPTION 5, RAND3 ERROR 03 ****

                        !CHECK OR READ:

!ERRDF(5304,MSG1,0);    !**** OPTION 5, RAND3 ERROR 04 ****
!ERRDF(5305,MSG1,0);    !**** OPTION 5, RAND3 ERROR 05 ****
!ERRDF(5306,MSG1,0);    !**** OPTION 5, RAND3 ERROR 06 ****
!ERRDF(5307,MSG1,0);    !**** OPTION 5, RAND3 ERROR 07 ****
!ERRDF(5308,MSG2,0);    !**** OPTION 5, RAND3 ERROR 08 ****
!ERRHRD(5309,MSG4,0);   !**** OPTION 5, RAND3 ERROR 09 ****
!ERRSOFT(5310,MSG3,0);  !**** OPTION 5, RAND3 ERROR 10 ****
!ERRSOFT(5311,MSG3,0);  !**** OPTION 5, RAND3 ERROR 11 ****

            !'RAND4' ROUTINE:

                        !WRITE COMMAND:

!ERRDF(5401,MSG1,0);    !**** OPTION 5, RAND4 ERROR 01 ****
!ERRDF(5402,MSG1,0);    !**** OPTION 5, RAND4 ERROR 02 ****
!ERRDF(5403,MSG1,0);    !**** OPTION 5, RAND4 ERROR 03 ****

                        !CHECK OR READ:

!ERRDF(5404,MSG1,0);    !**** OPTION 5, RAND4 ERROR 04 ****
!ERRDF(5405,MSG1,0);    !**** OPTION 5, RAND4 ERROR 05 ****
!ERRDF(5406,MSG1,0);    !**** OPTION 5, RAND4 ERROR 06 ****
!ERRDF(5407,MSG1,0);    !**** OPTION 5, RAND4 ERROR 07 ****
!ERRDF(5408,MSG2,0);    !**** OPTION 5, RAND4 ERROR 08 ****
!ERRHRD(5409,MSG4,0);   !**** OPTION 5, RAND4 ERROR 09 ****
!ERRSOFT(5410,MSG3,0);  !**** OPTION 5, RAND4 ERROR 10 ****
!ERRSOFT(5411,MSG3,0);  !**** OPTION 5, RAND4 ERROR 11 ****
```

4.0 PERFORMANCE AND PROGRESS REPORTS

AT THE END OF EACH PASS, THE PASS COUNT IS GIVEN ALONG WITH A SUMMARY WHICH SHOWS THE DIAGNOSTIC'S PERFORMANCE SINCE IT WAS STARTED. THE SAME PROGRESS REPORT CAN BE OBTAINED BY STOPPING THE PROGRAM'S EXECUTION (VIA A ^C) AND BY ISSUING THE 'PRINT' COMMAND. A TYPICAL REPORT FOR 2 DRIVES IS SHOWN BELOW:

```
PERFORMANCE SUMMARY

NUMBER OF MBYTES TRANSFERRED:
 1028 MBYTES WRITTEN
  250 MBYTES READ
 1145 MBYTES WRITE CHECKED

LOGICAL UNIT: 0    DRIVE: 1    SERIAL #: 1234
  SOFT ERROR COUNT:     9
    ARRAY  3:     9
  HARD ERROR COUNT:    11
    ARRAY  0:     6
    ARRAY 10:     2
    ARRAY 15:     3
  TRANSFER RETRIES:     0
```

```
LOGICAL UNIT: 1    DRIVE: 1    SERIAL #: 9876
DRIVE DROPPED   (CONTROLLER FATAL ERROR)
  SOFT ERROR COUNT:    100
    ARRAY  1:        9
    ARRAY 13:       10  --> RUN ML11 PROM MAINTENANCE PROGRAM
    ARRAY 14:        1
    ARRAY 15:       80  --> RUN ML11 PROM MAINTENANCE PROGRAM
  HARD ERROR COUNT:      2
    ARRAY 14:        1
    ARRAY 15:        1
  TRANSFER RETRIES:      0
```

5.0 DEVICE INFORMATION TABLES

AT THE START OF THE PROGRAM, AN AUTOMATIC CHECK OF THE SYSTEM CONFIGURATION IS MADE AND DEVICE INFORMATION SIMILAR TO THE FOLLOWING IS PRINTED FOR EACH UNIT:

```
    LOGICAL UNIT: 0    DRIVE: 0    SERIAL #: 1234
    ML11-A     SECTORS UNDER TEST: 000000 TO 017777
    TRANSFER RATE: 1 MBYTES/SECOND    CSR ADDRESS: 176400
```

6.0 TEST SUMMARIES

THERE IS JUST ONE HARDWARE TEST IN THE EXERCISER, AND ITS PURPOSE IS TO ACT AS A SCHEDULER TO CALL THE SUBROUTINES WHICH ACTUALLY PERFORM ALL OF THE TEST CODE. THE SUBROUTINES, CALLED OPTIONS, ARE DESCRIBED BELOW:

6.1 OPTION 1 (OPT1)

PURPOSE: TO CHECK ADDRESSES USING DATA = SECTOR NUMBER. TRANSFERS ARE 4K WORDS IN LENGTH, AND ALL SECTORS ARE TESTED.

THE CODE FOR 'OPT1' IN BRIEF:

```
BEGIN 1 (START OF ROUTINE)
SAY ROUTINE IS RUNNING
INCR COMPLEMENT FLAG FROM 0 TO 1
: BEGIN 2 (START OF COMPLEMENT FLAG SELECTION LOOP)
: INCR LOGICAL UNIT FROM 0 TO LAST
: : BEGIN 3 (START OF LOGICAL UNIT SELECTION LOOP)
: : TESTLOOP:
: : : BEGIN 4 (START OF LOOP THAT COMPLETELY TESTS ONE UNIT)
: : : IF UNIT IS ACTIVE
: : : THEN
: : : : BEGIN 5 (START OF TEST FOR AN ACTIVE UNIT)
: : : : INITIALIZE WRITE AND READ BUFFER POINTERS
: : : : SECTOR = LOWEST
: : : : WHILE SECTOR LEQ HIGHEST DO
: : : : : BEGIN 6 (START OF SECTOR SELECTION LOOP)
: : : : : GET WRDCNT
: : : : : GENERATE THE PATTERN
```

```
: : : : : SET UP BUFFER POINTERS BEFORE TRANSFER
: : : : : WRITE
: : : : : LOOK FOR ERRORS (IF DROP UNIT, LEAVE TESTLOOP)
: : : : : CHOOSE WHETHER TO WRITE CHECK OR READ
: : : : : DO THE WRITE CHECK OR READ
: : : : : LOOK FOR ERRORS (IF DROP UNIT, LEAVE TESTLOOP)
: : : : : CHANGE BUFFER POINTERS AFTER TRANSFER
: : : : : UPDATE SECTOR NUMBER BY NUMBER OF SECTORS IN PREVIOUS TRANSFER
: : : : : END 6 (END OF SECTOR SELECTION LOOP)
: : : : END 5 (END OF TEST FOR AN ACTIVE UNIT)
: : : END 4 (END OF TESTLOOP)
: : END 3 (END OF LOGICAL UNIT SELECTION LOOP)
: END 2 (END OF COMPLEMENT FLAG SELECTION LOOP)
RETURN
END 1 (END OF ROUTINE)
```

6.2 OPTION 2 (OPT2)

PURPOSE: TO CHECK ON DATA RELIABILITY USING THE PATTERNS FROM THE PATTERN TABLE.

THE CODE FOR 'OPT2' IN BRIEF:

```
BEGIN 1 (START OF ROUTINE)
SAY THE ROUTINE IS RUNNING
CHOOSE A MAXIMUM PATTERN NUMBER
INCR COUNT FROM 1 TO (2*MAX)
: BEGIN 2 (START OF PATTERN SELECTION LOOP)
: GENERATE THE PATTERN
: INCR LUN FROM 0 TO LAST
: : BEGIN 3 (START OF LOGICAL UNIT SELECTION LOOP)
: : TESTLOOP:
: : : BEGIN 4 (START OF LOOP THAT COMPLETELY TESTS 1 UNIT)
: : : IF UNIT IS ACTIVE
: : : THEN
: : : : BEGIN 5 (START OF TEST FOR AN ACTIVE UNIT)
: : : : INITIALIZE THE WRITE AND READ BUFFER POINTERS
: : : : SECTOR = LOWEST
: : : : WHILE SECTOR LEQ HIGHEST DO
: : : : : BEGIN 6 (START OF SECTOR SELECTION LOOP)
: : : : : GET_WRDCNT
: : : : : SET_UP BUFFER POINTERS BEFORE TRANSFER
: : : : : WRITE
: : : : : LOOK FOR ERRORS (IF DROP UNIT, LEAVE TESTLOOP)
: : : : : CHOOSE WHETHER TO WRITE CHECK OR READ
: : : : : DO THE WRITE CHECK OR READ
: : : : : LOOK FOR ERRORS (IF DROP UNIT, LEAVE TESTLOOP)
: : : : : CHANGE BUFFER POINTERS AFTER TRANSFER
: : : : : UPDATE SECTOR NUMBER BY NUMBER OF SECTORS IN PREVIOUS TRANSFER
: : : : : END 6 (END OF SECTOR SELECTION LOOP)
: : : : END 5 (END OF TEST FOR AN ACTIVE UNIT)
: : : END 4 (END OF TESTLOOP)
: : END 3 (END OF LOGICAL UNIT SELECTION LOOP)
: IF NOT THE QUICK VERIFY PASS THEN 'LOOP READ' (DESCRIBED BELOW)
: END 2 (END OF PATTERN SELECTION LOOP)
RETURN
END 1 (END OF ROUTINE)
```

```
THE 'LOOP READ' CODE IN BRIEF:

BEGIN 11 (START OF LOOP READING SECTION)
INCR LUN FROM 0 TO LAST
: BEGIN 12 (START OF LOGICAL UNIT SELECTION LOOP)
: TESTLOOP2:
: : BEGIN 13 (START OF LOOP THAT COMPLETELY TESTS 1 UNIT)
: : IF UNIT IS ACTIVE
: : THEN
: : : BEGIN 14 (START OF TEST FOR AN ACTIVE UNIT)
: : : INITIALIZE THE WRITE AND READ BUFFER POINTERS
: : : SECTOR = LOWEST
: : : WHILE SECTOR LEQ HIGHEST DO
: : : : BEGIN 15 (START OF SECTOR SELECTION LOOP)
: : : : GET_WRDCNT
: : : : SET_UP BUFFER POINTERS BEFORE TRANSFER
: : : : CHOOSE WHETHER TO WRITE CHECK OR READ
: : : : INCR KOUNT FROM 1 TO TIMES
: : : : : BEGIN 16 (START OF COUNTING LOOP FOR LOOP READING)
: : : : : DO THE WRITE CHECK OR READ
: : : : : LOOK FOR ERRORS (IF DROP UNIT, LEAVE TESTLOOP2)
: : : : : END 16 (END OF COUNTING LOOP FOR LOOP READING)
: : : : CHANGE BUFFER POINTERS AFTER TRANSFER
: : : : UPDATE SECTOR NUMBER BY # SECTORS IN PREVIOUS TRANSFER
: : : : END 15 (END OF SECTOR SELECTION LOOP)
: : : END 14 (END OF TEST FOR AN ACTIVE UNIT)
: : END 13 (END OF TESTLOOP2)
: END 12 (END OF LOGICAL UNIT SELECTION LOOP)
END 11 (END OF LOOP READING SECTION)
```

6.3 OPTION 3 (OPT3)

PURPOSE: TO DO A UNIQUE DATA CHECK ON ALL AVAILABLE UNITS.

THE CODE FOR 'OPT3' IN BRIEF:

```
BEGIN 1 (START OF ROUTINE)
SAY ROUTINE IS RUNNING
INCR COMPLEMENT FLAG FROM 0 TO 1
: BEGIN 2 (START OF COMPLEMENT FLAG SELECTION LOOP)
: GENERATE THE PATTERN
: INCR LUN FROM 0 TO LAST
: : BEGIN 3 (START OF LOGICAL UNIT SELECTION LOOP)
: : TESTLOOP:
: : : BEGIN 4 (START OF LOOP THAT COMPLETELY TESTS 1 UNIT)
: : : IF UNIT IS ACTIVE
: : : THEN
: : : : BEGIN 5 (START OF TEST FOR AN ACTIVE UNIT)
: : : : INITIALIZE THE WRITE AND READ BUFFER POINTERS
: : : : SECTOR = LOWEST
: : : : WHILE SECTOR LEQ HIGHEST DO
: : : : : BEGIN 6 (START OF SECTOR SELECTION LOOP)
: : : : : GET_WRDCNT
: : : : : SET_UP BUFFER POINTERS BEFORE TRANSFER
: : : : : WRITE
: : : : : LOOK FOR ERRORS (IF DROP UNIT, LEAVE TESTLOOP)
```

```
: : : : : CHOOSE WHETHER TO WRITE CHECK OR READ
: : : : : DO THE WRITE CHECK OR READ
: : : : : LOOK FOR ERRORS (IF DROP UNIT, LEAVE TESTLOOP)
: : : : : CHANGE BUFFER POINTERS AFTER TRANSFER
: : : : : UPDATE SECTOR NUMBER BY NUMBER OF SECTORS IN PREVIOUS TRANSFER
: : : : : END 6 (END OF SECTOR SELECTION LOOP)
: : : : END 5 (END OF TEST FOR AN ACTIVE UNIT)
: : : END 4 (END OF TESTLOOP)
: : END 3 (END OF LOGICAL UNIT SELECTION LOOP)
: END 2 (END OF COMPLEMENT FLAG SELECTION LOOP)
RETURN
END 1 (END OF ROUTINE)
```

6.4 OPTION 4 (OPT4)

PURPOSE: TO LOOK FOR INTEREACTIONS BETWEEN SECTORS
USING A MARCH TEST.

THE CODE FOR 'OPT4' IN BRIEF:

```
BEGIN 1 (START OF ROUTINE)
SAY ROUTINE IS RUNNING
WORD COUNT = 256
GENERATE A BUFFER OF DATA
GENERATE A BUFFER OF COMP
INITIALIZE POINTERS TO 4 BUFFERS FOR WRITE/READ DATA/COMP
INCR LUN FROM 0 TO LAST
: BEGIN 2 (START OF LOGICAL UNIT SELECTION LOOP)
: TESTLOOP:
: : BEGIN 3 (START OF LOOP THAT COMPLETELY TESTS 1 UNIT)
: : IF UNIT IS ACTIVE
: : THEN
: : : BEGIN 4 (START OF TEST FOR AN ACTIVE UNIT)
: : : INCR SECTOR FROM LOWEST TO HIGHEST
: : : : BEGIN 4A (START OF 1ST SECTOR SELECTION LOOP)
: : : : WRITE 'DATA'
: : : : LOOK FOR ERRORS (IF DROP UNIT, LEAVE TESTLOOP)
: : : : END 4A (END OF 1ST SECTOR SELECTION LOOP)
: : : CHOOSE WHETHER TO WRITE CHECK OR READ 'DATA'
: : : INCR SECTOR FROM LOWEST TO HIGHEST
: : : : BEGIN 4B (START OF 2ND SECTOR SELECTION LOOP)
: : : : DO THE WRITE CHECK OR READ OF 'DATA'
: : : : LOOK FOR ERRORS (IF DROP UNIT, LEAVE TESTLOOP)
: : : : WRITE 'COMP'
: : : : LOOK FOR ERRORS (IF DROP UNIT, LEAVE TESTLOOP)
: : : : END 4B (END OF 2ND SECTOR SELECTION LOOP)
: : : CHOOSE WHETHER TO WRITE CHECK OR READ 'COMP'
: : : DECR SECTOR FROM HIGHEST TO LOWEST
: : : : BEGIN 4C (START OF 3RD SECTOR SELECTION LOOP)
: : : : DO THE WRITE CHECK OR READ OF 'COMP'
: : : : LOOK FOR ERRORS (IF DROP UNIT, LEAVE TESTLOOP)
: : : : WRITE DATA
: : : : LOOK FOR ERRORS (IF DROP UNIT, LEAVE TESTLOOP)
: : : : END 4C (END OF 3RD SECTOR SELECTION LOOP)
: : : CHOOSE WHETHER TO WRITE CHECK OR READ 'DATA'
: : : INCR SECTOR FROM LOWEST TO HIGHEST
: : : : BEGIN 4D (START OF 4TH SECTOR SELECTION LOOP)
```

```
: : : : DO THE WRITE CHECK OR READ OF 'DATA'
: : : : LOOK FOR ERRORS (IF DROP UNIT, LEAVE TESTLOOP)
: : : : END 4D (END OF 4TH SECTOR SELECTION LOOP)
: : : END 4 (END OF TEST FOR AN ACTIVE UNIT)
: : END 3 (END OF TESTLOOP)
: END 2 (END OF LOGICAL UNIT SELECTION LOOP)
RETURN
END 1 (END OF ROUTINE)
```

6.5 OPTION 5 (OPT5)

PURPOSE: TO EXERCISE THE ML11 SYSTEMS UNDER TEST IN A RANDOM MANNER. SO AS TO SIMULATE THE FLEXIBILITY OF TESTING THAT WOULD BE DONE BY AN OPERATING SYSTEM.

THERE ARE 4 RANDOM TESTS WHICH ARE CALLED BY 'OPT5' TO ACCOMPLISH ALL TESTING. IT IS THE RESPONSIBILITY OF THIS ROUTINE TO DECIDE HOW MANY TIMES THOSE 4 RANDOM TESTS WILL BE EXECUTED. REFER TO 'RAND1' TO 'RAND4' BELOW FOR MORE INFORMATION.

6.5.1 RAND1 ROUTINE

PURPOSE: TO TEST USING RANDOM DATA

THE CODE FOR 'RAND1' IN BRIEF:

```
BEGIN 1  (START OF ROUTINE)
SAY ROUTINE IS RUNNING
INCR COUNT FROM 1 TO REPEAT
: BEGIN 2 (START OF REPEAT LOOP FOR THE ROUTINE)
: GENERATE THE RANDOM PATTERN
: INCR LUN FROM 0 TO LAST
: : BEGIN 3 (START OF LOGICAL UNIT SELECTION LOOP)
: : TESTLOOP:
: : : BEGIN 4 (START OF LOOP THAT COMPLETELY TESTS 1 UNIT)
: : : IF UNIT IS ACTIVE
: : : THEN
: : : : BEGIN 5 (START OF TEST FOR AN ACTIVE UNIT)
: : : : INITIALIZE THE WRITE AND READ BUFFER POINTERS
: : : : SECTOR = LOWEST
: : : : WHILE SECTOR LEQ HIGHEST DO
: : : : : BEGIN 6 (START OF SECTOR SELECTION LOOP)
: : : : : GET_WRDCNT
: : : : : SET_UP BUFFER POINTERS BEFORE TRANSFER
: : : : : WRITE
: : : : : LOOK FOR ERRORS (IF DROP UNIT, LEAVE TESTLOOP)
: : : : : CHOOSE WHETHER TO WRITE CHECK OR READ
: : : : : DO THE WRITE CHECK OR READ
: : : : : LOOK FOR ERRORS (IF DROP UNIT, LEAVE TESTLOOP)
: : : : : CHANGE BUFFER POINTERS AFTER TRANSFER
: : : : : UPDATE SECTOR NUMBER BY NUMBER OF SECTORS IN PREVIOUS TRANSFER
: : : : : END 6 (END OF SECTOR SELECTION LOOP)
: : : : END 5 (END OF TEST FOR AN ACTIVE UNIT)
: : : END 4 (END OF TESTLOOP)
: : END 3 (END OF LOGICAL UNIT SELECTION LOOP)
```

```
: END 2 (END OF REPEAT LOOP FOR THIS ROUTINE)
RETURN
END 1 (END OF ROUTINE)
```

6.5.2 RAND2 ROUTINE

PURPOSE: TO TEST USING RANDOM DATA AND WORD COUNTS

THE CODE FOR 'RAND2' IN BRIEF:

```
BEGIN 1 (START OF ROUTINE)
SAY ROUTINE IS RUNNING
INCR COUNT FROM 1 TO REPEAT
: BEGIN 2 (START OF REPEAT LOOP FOR THE ROUTINE)
: INCR LUN FROM 0 TO LAST
: : BEGIN 3 (START OF LOGICAL UNIT SELECTION LOOP)
: : GENERATE THE RANDOM PATTERN
: : TESTLOOP:
: : : BEGIN 4 (START OF LOOP THAT COMPLETELY TESTS 1 UNIT)
: : : IF UNIT IS ACTIVE
: : : THEN
: : : : BEGIN 5 (START OF TEST FOR AN ACTIVE UNIT)
: : : : SECTOR = LOWEST
: : : : INITIALIZE THE WRITE AND READ BUFFER POINTERS
: : : : WHILE SECTOR LEQ HIGHEST DO
: : : : : BEGIN 6 (START OF A PASS THROUGH ALL SECTORS)
: : : : : CHOOSE A RANDOM WORD COUNT
: : : : : SET UP BUFFER POINTERS BEFORE TRANSFER
: : : : : CALCULATE NEXT STARTING SECTOR (BASED ON WORD COUNT)
: : : : : IF NEXT STARTING SECTOR GTR HIGHEST
: : : : : : THEN ADJUST THE WORD COUNT AND NEXT SECTOR SO THEY FIT
: : : : : : :    WITHIN THE TESTABLE SECTOR LIMITS
: : : : : WRITE
: : : : : LOOK FOR ERRORS (IF DROP UNIT, LEAVE TESTLOOP)
: : : : : CHOOSE WHETHER TO WRITE CHECK OR READ
: : : : : DO THE WRITE CHECK OR READ
: : : : : LOOK FOR ERRORS (IF DROP UNIT, LEAVE TESTLOOP)
: : : : : SECTOR = THE CALCULATED NEXT STARTING SECTOR
: : : : : CHANGE BUFFER POINTERS AFTER TRANSFER
: : : : : END 6 (END OF A PASS THROUGH ALL SECTORS)
: : : : END 5 (END OF TEST FOR AN ACTIVE UNIT)
: : : END 4 (END TESTLOOP)
: : END 3 (END OF LOGICAL UNIT SELECTION LOOP)
: END 2 (END OF REPEAT LOOP FOR THIS ROUTINE)
RETURN
END 1 (END OF ROUTINE)
```

6.5.3 RAND3 ROUTINE

PURPOSE: TO TEST USING RANDOM DATA, WORD COUNTS AND SECTORS

THE CODE FOR 'RAND3' IN BRIEF:

```
BEGIN 1 (START OF ROUTINE)
SAY ROUTINE IS RUNNING
INCR COUNT FROM 1 TO REPEAT
: BEGIN 2 (START OF REPEAT LOOP FOR THIS ROUTINE)
: GENERATE THE RANDOM PATTERN
: INCR LUN FROM 0 TO LAST
```

```
: : BEGIN 3 (START OF LOGICAL UNIT SELECTION LOOP)
: : INITIALIZE BUFFER POINTERS
: : TESTLOOP:
: : : BEGIN 4 (START OF LOOP THAT COMPLETELY TESTS 1 UNIT)
: . : IF UNIT IS ACTIVE
: : : THEN
: : : : BEGIN 5 (START OF TEST FOR AN ACTIVE UNIT)
: : : : TIMES = (HIGHEST - LOWEST)/2 + 1
: : : : INCR KOUNT FROM 1 TO TIMES
: : : : : BEGIN 6 (START OF COUNTING LOOP FOR SECTOR SELECTION)
: : : : : CHOOSE A RANDOM WORD COUNT
: : : : : SET UP BUFFER POINTERS BEFORE TRANSFER
: : : : : CHOOSE A RANDOM SECTOR
: : : : : CALCULATE WHERE TRANSFER WILL END (BASED ON WORD COUNT)
: : : : : IF CALCULATED VALUE GTR HIGHEST
: : : : : : THEN ADJUST THE WORD COUNT SO IT FITS
: : : : : : :    WITHIN THE TESTABLE SECTOR LIMITS
: : : : : WRITE
: : : : : LOOK FOR ERRORS (IF DROP UNIT, LEAVE LABEL)
: : : : : CHOOSE WHETHER TO WRITE CHECK OR READ
: : : : : DO THE WRITE CHECK OR READ
: : : : : LOOK FOR ERRORS (IF DROP UNIT, LEAVE LABEL)
: : : : : CHANGE BUFFER POINTERS AFTER TRANSFER
: : : : : END 6 (END OF COUNTING LOOP FOR SECTOR SELECTION)
: : : : END 5 (END OF TEST FOR AN ACTIVE UNIT)
: : : END 4 (END OF TESTLOOP)
: : END 3 (END OF LOGICAL UNIT SELECTION LOOP)
: END 2 (END OF REPEAT LOOP FOR THIS ROUTINE)
RETURN
END 1 (END OF ROUTINE)
```

6.5.4 RAND4 ROUTINE

PURPOSE: TO TEST USING RANDOM DATA, WORD COUNTS, SECTORS AND UNITS

```
THE CODE FOR 'RAND4' IN BRIEF:
BEGIN 1 (START OF ROUTINE)
SAY ROUTINE IS RUNNING
INCR COUNT FROM 1 TO REPEAT
: BEGIN 2 (START OF REPEAT LOOP FOR THIS ROUTINE)
: GENERATE THE RANDOM PATTERN
: TIMES = NUMBER OF UNITS * 4
: INCR KOUNT FROM 1 TO TIMES
: : BEGIN 3 (START OF COUNTING LOOP FOR UNIT SELECTION)
: : CHOOSE A RANDOM LOGICAL UNIT WHICH IS ACTIVE
: : INITIALIZE BUFFER POINTERS
: : TESTLOOP:
: : : BEGIN 4 (START OF LOOP THAT COMPLETELY TESTS 1 UNIT)
: : : INCR KOUNT2 FROM 1 TO 10
: : : : BEGIN 5 (START OF COUNTING LOOP FOR SECTOR SELECTION)
: : : : CHOOSE A RANDOM WORD COUNT
: : : : SET UP BUFFER POINTERS BEFORE TRANSFER
: : : : CHOOSE A RANDOM SECTOR
: : : : CALCULATE WHERE TRANSFER WILL END (BASED ON WORD COUNT)
: : : : IF CALCULATED VALUE GTR HIGHEST
: : : : : THEN ADJUST THE WORD COUNT SO IT FITS
: : : : : :    WITHIN THE TESTABLE SECTOR LIMITS
: : : : WRITE
```

```
: : : : LOOK FOR ERRORS (IF DROP UNIT, LEAVE TESTLOOP)
: : : : CHOOSE WHETHER TO WRITE CHECK OR READ
: : : : DO THE WRITE CHECK OR READ
: : : : LOOK FOR ERRORS (IF DROP UNIT, LEAVE TESTLOOP)
: : : : CHANGE BUFFER POINTERS AFTER TRANSFER
: : : : END 5 (END OF COUNTING LOOP FOR SECTOR SELECTION)
: : : END 4 (END OF TESTLOOP)
: : END 3 (END OF COUNTING LOOP FOR UNIT SELECTION)
: END 2 (END OF REPEAT LOOP FOR THIS ROUTINE
RETURN
END 1 (END OF ROUTINE)
```

7.0 MAINTENANCE HISTORY

MODIFIED BY: D.W. NEALE DATE: 18-FEB-82 VERSION: B

ALL FUNCTIONAL CHANGES TO THIS DIAGNOSTIC ARE LABLED WITH A COMMENT LINE OF 'VER CZMLBB' PRECEEDING ANY MODIFIED OR ADDED LINE/BLOCK OF CODE.

FUNCTIONAL CHANGES TO THIS DIAGNOSTIC INCLUDE:

1. MODIFING CODE TO ENSURE QUALITY TESTING OF ML-11B AND ML-11A BLOCK MODE MEMORY SYSTEMS.

2. CORRECT EXERCISER FUNCTIONALITY TO CALL OUT ML11 PROM MAINTENANCE PROGRAM AFTER 10 UNIQUE FAILING SINGLE BIT ERRORS (SBE'S) HAVE BEEN DISCOVERED PER ARRAY MODULE.

   VERSION 'A' PROM MAINTENANCE CALL OUT OCCURES AFTER ANY TEN SBE'S ARE DISCOVERED PER ARRAY MODULE.

3. PER REQUEST OF F/S AND MEMORY ENGINEERING, THE REPORTING OF 'MDPE' ERRORS DURING SBE'S CORRECTION WILL BE IGNORED.

4. ADDING TO THE REPORT SUMMARY CODE SECTION A TABLE OF SBE LOCATIONS AND PRINTING OF THIS TABLE DURING REPORT CODE PRINTING.

```
                        .SBTTL  PROGRAM HEADER AND TABLES

000000                  .ENABL  ABS,AMA
002000          .       =       2000

002000                  BGNMOD

                ;++
                ; THE PROGRAM HEADER IS THE INTERFACE BETWEEN
                ; THE DIAGNOSTIC PROGRAM AND THE SUPERVISOR.
                ;--

002000                  POINTER ALL

002000                  HEADER  CZMLB,B,0,1800.,1
002000
002000  103             LSNAME::
002001  132                     .ASCII  /C/
002002  115                     .ASCII  /Z/
002003  114                     .ASCII  /M/
002004  102                     .ASCII  /L/
002005  000                     .ASCII  /B/
002006  000                     .BYTE   0
002007  000                     .BYTE   0
002010                          .BYTE   0
002010  102             LSREV::
002011                          .ASCII  /B/
002011  060             LSDEPO::
002012                          .ASCII  /0/
002012  000000          LSUNIT::
002014                          .WORD   T$PTHV
002014  003410          LSTIML::
002016                          .WORD   1800.
002016  002266          LSHPCP::
002020                          .WORD   LSHARD
002020  002450          LSSPCP::
002022                          .WORD   LSSOFT
002022  002210          LSHPTP::
002024                          .WORD   LSHW
002024  002222          LSSPTP::
002026                          .WORD   LSSW
002026  105050          LSLADP::
002030                          .WORD   LSLAST
002030  000000          LSSTA::
002032                          .WORD   0
002032  000000          LSCO::
002034                          .WORD   0
002034  000001          LSDTYP::
002036                          .WORD   1
002036  000000          LSAPT::
002040                          .WORD   0
002040  002204          LSDTP::
002042                          .WORD   LSDISPATCH
002042  000000          LSPRIO::
002044                          .WORD   0
002044  000000          LSENVI::
                                .WORD   0
```

```
002046
002046  000000
002050
002050     003
002051     003
002052
002052  000000
002054  000000
002056
002056  000000
002060
002060  002122
002062
002062  005474
002064
002064  000000
002066
002066  000000
002070
002070  005652
002072
002072  005640
002074
002074  000000
002076
002076  002130
002100
002100  104035
002102
002102  002172
002104
002104  040722
002106
002106  105034
002110
002110  005506
002112
002112  004126
002114
002114  000000
002116
002116  000000
002120
002120  000000

66
77
78                              ;       NAMES OF DEVICES SUPPORTED BY THIS PROGRAM
79                              ;
80 002122                       ;       DEVTYP <ML11>
   002122
   002122     115     114     061

LSEXP1::
        .WORD   0
LSMREV::
        .BYTE   C$REVISION
        .BYTE   C$EDIT
LSEF::
        .WORD   0
        .WORD   0
LSSPC::
        .WORD   0
LSDEVP::
        .WORD   LSDVTYP
LSREPP::
        .WORD   LSRPT
LSEXP4::
        .WORD   0
LSEXP5::
        .WORD   0
LSAUT::
        .WORD   L3AU
LSDUT::
        .WORD   LSDU
LSLUN::
        .WORD   0
LSDESP::
        .WORD   LSDESC
LSLOAD::
        EMT     ESLOAD
LSETP::
        .WORD   LSERRTBL
LSICP::
        .WORD   LSINIT
LSCCP::
        .WORD   LSCLEAN
LSACP::
        .WORD   LSAUTO
LSPRT::
        .WORD   LSPROT
LSTEST::
        .WORD   0
LSDLY::
        .WORD   0
LSHIME::
        .WORD   0

LSDVTYP::
        .ASCIZ  /ML11/
        .EVEN
```

```
                                ;
                                ;       TEST DESCRIPTION
                                ;
002130                                  DESCRIPT <CZMLBB ML11 PERFORMANCE EXERCISER>
002130                                                                          LSDESC::
002130  103     132     115                                                             .ASCIZ  /CZMLBB ML11 PERFORM
                                                                                        .EVEN


                                ;
                                ;       THE GLOBAL ERROR TABLE (INFORMATION
                                ;       USED IN A CALL TO THE MACRO 'ERROR')
                                ;
002172                          ERRTBL
002172                                                                          LSERRTBL::
002172  000000                  ERRTYP::        .WORD   0
002174  000000                  ERRNBR::        .WORD   0
002176  000000                  ERRMSG::        .WORD   0
002200  000000                  ERRBLK::        .WORD   0

                                ;++
                                ; THE DISPATCH TABLE CONTAINS THE STARTING ADDRES OF EACH TEST.
                                ; IT IS USED BY THE SUPERVISOR TO DISPATCH TO EACH TEST.
                                ;--

002202                                  DISPATCH 1
002202  000001                                                                          .WORD   1
002204                                                                          LSDISPATCH::
002204  102712                                                                        .WORD   T1


                                ;++
                                ; THE DEFAULT HARDWARE P-TABLE CONTAINS DEFAULT VALUES OF
                                ; THE TEST-DEVICE PARAMETERS.  THE STRUCTURE OF THIS TABLE
                                ; IS IDENTICAL TO THE STRUCTURE OF THE HARDWARE P-TABLES.
                                ; AND IS USED AS A "TEMPLATE" FOR BUILDING THE P-TABLES.
                                ;--

002206                                  BGNHW   DFPTBL
002206  000004                                                                         .WORD   L10000-L$HW/2
002210                                                                          L$HW::
002210                                                                          DFPTBL::


002210  176400                          .WORD   176400          ;CSR ADDRESS
002212  000204                          .WORD   204             ;RH VECTOR ADDRESS
002214  000005                          .WORD   5               ;BR LEVEL FOR INTERRUPT
002216  000000                          .WORD   0               ;ML11 DRIVE NUMBER

002220                                  ENDHW
002220                                                                          L10000:
```

```
                        ;++
                        ; THE DEFAULT SOFTWARE P-TABLE CONTAINS VARIOUS DATA USED BY THE
                        ; PROGRAM AS OPERATIONAL PARAMETERS.  THESE PARAMETERS ARE SET
                        ; UP AT ASSEMBLY TIME AND MAY BE VARIED BY THE OPERATOR AT RUN
                        ; TIME.
                        ;--

002220                          BGNSW   SFPTBL
002220  000021                                                          .WORD   L10001-L$SW/2
002222                                                                  L$SW::
002222                                                                  SFPTBL::

002222  000000  LIMIT:: .WORD   0           ;LIMIT RANGE OF SECTORS TO BE TESTED
                                            ;       0 = NO  1 = YES
002224  000000  RANGE:: .WORD   0           ;DOES OPERATOR KNOW EXACT SECTOR NUMBERS?
                                            ;       0 = NO  1 = YES
002226  000000  LSECT:: .WORD   0           ;LOW SECTOR NUMBER
002230  000000  TSECT:: .WORD   0           ;TOP SECTOR NUMBER
002232  000000  ONLY::  .WORD   0           ;ONLY BOARD TO TEST (BOARD #'S ARE 0 TO 15)
002234  000000  DROPNE:: .WORD  0           ;DROP ANY OPTIONS?      0 = NO  1 = YES
002236  000000  DROP1:: .WORD   0           ;DROP OPTION #1?        0 = NO  1 = YES
002240  000000  DROP2:: .WORD   0           ;DROP OPTION #2?        0 = NO  1 = YES
002242  000000  DROP3:: .WORD   0           ;DROP OPTION #3?        0 = NO  1 = YES
002244  000000  DROP4:: .WORD   0           ;DROP OPTION #4?        0 = NO  1 = YES
002246  000001  MARPAT:: .WORD  1           ;PATTERN NUMBER USED FOR MARCH TEST
002250  000000  DROP5:: .WORD   0           ;DROP OPTION #5?        0 = NO  1 = YES
002252  000000  REFRESH::.WORD  0           ;ENABLE MARGINING?      0 = NO (LEAVE DISABLED)
                                            ;                       1 = YES (ENABLE IT)
002254  000000  ECCDIS:: .WORD  0           ;DISABLE ECC?           0 = NO (LEAVE ENABLED)
                                            ;                       1 = YES (DISABLE IT)


                        ;*********** IMPORTANT NOTE ***********
                        ; THE FOLLOWING 2 SOFTWARE PARAMETERS ;
                        ; HAVE NON-STANDARD DEFAULTS.  ERRORS ;
                        ; AND END OF PASS PERFORMANCE SUMMARY ;
                        ; REPORTS WILL ** N O T ** BE PRINTED ;
                        ; UNLESS  THE  OPERATOR  SPECIFICALLY ;
                        ; REQUESTS THE PRINTOUTS VIA SUITABLE ;
                        ; ANSWERS TO THE SOFTWARE QUESTIONS.  ;
                        ;                                     ;
                        ; THIS OPERATING FEATURE WAS INCLUDED ;
                        ; IN THE ML11 PERFORMANCE EXERCISER'S ;
                        ; FUNCTIONAL SPECIFICATION AT THE RE- ;
                        ; QUEST OF FIELD SERVICE.             ;
                        ;*************************************

002256  000000  EOPSUM:: .WORD  0           ;ENABLE EOP SUMMARIES?  0 = NO PRINTOUT
                                            ;                       1 = YES
002260  000000  ERROUT:: .WORD  0           ;ENABLE ERROR PRINTOUTS? 0 = NO PRINTOUT
                                            ;                        1 = YES
002262  000000  EFN$21:: .WORD  0           ;ENABLE SOFT ERROR TESTING 0 = NO (DEFAULT)
                                            ;                          1 = YES
                                            ;THIS OPTION IS DESIGNED FOR MEMORY ENGINEERING
                                            ;FOR DMT PURPOSES

002264                          ENDSW
```

```
     002264                                                                                    L10001:
  197
  222
  248
  249
  250                                   ;++
  251                                   ; THE HARDWARE PARAMETER CODING SECTION CONTAINS MACROS
  252                                   ; THAT ARE USED BY THE SUPERVISOR TO BUILD P-TABLES.  THE
  253                                   ; MACROS ARE NOT EXECUTED AS MACHINE INSTRUCTIONS BUT ARE
  254                                   ; INTERPRETED BY THE SUPERVISOR AS DATA STRUCTURES.  THE
  255                                   ; MACROS ALLOW THE SUPERVISOR TO ESTABLISH COMMUNICATIONS
  256                                   ; WITH THE OPERATOR.
  257                                   ;--
  258
  259 002264                                    BGNHRD
      002264  000022                                                                  .WORD L10002-L$HARD/2
      002266                                                                    L$HARD::
  260
  270
  271
  272 002266                                    GPRMA   QH1,0,0,0,177777,YES
      002266  000031                                                               .WORD   T$CODE
      002270  002332                                                               .WORD   QH1
      002272  000000                                                               .WORD   T$LOLIM
      002274  177777                                                               .WORD   T$HILIM
  273 002276                                    GPRMA   QH2,2,0,0,377,YES
      002276  001031                                                               .WORD   T$CODE
      002300  002351                                                               .WORD   QH2
      002302  000000                                                               .WORD   T$LOLIM
      002304  000377                                                               .WORD   T$HILIM
  274 002306                                    GPRMD   QH3,4,0,7,1,7,YES
      002306  002032                                                               .WORD   T$CODE
      002310  002374                                                               .WORD   QH3
      002312  000007                                                               .WORD   7
      002314  000001                                                               .WORD   T$LOLIM
      002316  000007                                                               .WORD   T$HILIM
  275 002320                                    GPRMD   QH4,6,0,7,0,7,YES
      002320  003032                                                               .WORD   T$CODE
      002322  002425                                                               .WORD   QH4
      002324  000007                                                               .WORD   7
      002326  000000                                                               .WORD   T$LOLIM
      002330  000007                                                               .WORD   T$HILIM
  276
  277 002332                                    ENDHRD
                                                                                    .EVEN
      002332                                                                  L10002:
  278
  285
  286 002332      103     123     122   QH1:    .ASCIZ  /CSR ADDRESS              ?/
  287 002351      111     116     124   QH2:    .ASCIZ  /INTERRUPT VECTOR         ?/
  288 002374      102     122     040   QH3:    .ASCIZ  /BR LEVEL FOR INTERRUPT   ?/
  289 002425      104     122     111   QH4:    .ASCIZ  /DRIVE NUMBER             ?/
  290                                           .EVEN
  291
  292
  293                                   ;++
  294                                   ; THE SOFTWARE PARAMETER CODING SECTION CONTAINS MACROS
```

```
                        ; THAT ARE USED BY THE SUPERVISOR TO BUILD P-TABLES.  THE
                        ; MACROS ARE NOT EXECUTED AS MACHINE INSTRUCTIONS BUT ARE
                        ; INTERPRETED BY THE SUPERVISOR AS DATA STRUCTURES.  THE
                        ; MACROS ALLOW THE SUPERVISOR TO ESTABLISH COMMUNICATIONS
                        ; WITH THE OPERATOR.
                        ;--

002446                          BGNSFT
002446  000106                                                    .WORD L10003-L$SOFT/2
002450                                                      L$SOFT::

                        GPRML   QS1,0,1,YES
002450  000130                                                  .WORD   T$CODE
002452  002664                                                  .WORD   QS1
002454  000001                                                  .WORD   1
002456                  XFERF   6$
002456  025044                                                  .WORD   T$CODE
002460                  GPRML   QS2,2,1,YES
002460  001130                                                  .WORD   T$CODE
002462  002732                                                  .WORD   QS2
002464  000001                                                  .WORD   1
002466                  XFERF   5$
002466  014044                                                  .WORD   T$CODE
002470                  GPRMD   QS3,4,0,177777,0,177777,YES
002470  002032                                                  .WORD   T$CODE
002472  003007                                                  .WORD   QS3
002474  177777                                                  .WORD   177777
002476  000000                                                  .WORD   T$LOLIM
002500  177777                                                  .WORD   T$HILIM
002502                  GPRMD   QS4,6,0,177777,0,177777,YES
002502  003032                                                  .WORD   T$CODE
002504  003047                                                  .WORD   QS4
002506  177777                                                  .WORD   177777
002510  000000                                                  .WORD   T$LOLIM
002512  177777                                                  .WORD   T$HILIM
002514                  XFER    6$
002514  006004                                                  .WORD   T$CODE
002516  5$:     GPRMD   QS5,10,D,37,0,15..,YES
002516  004052                                                  .WORD   T$CODE
002520  003107                                                  .WORD   QS5
002522  000037                                                  .WORD   37
002524  000000                                                  .WORD   T$LOLIM
002526  000017                                                  .WORD   T$HILIM
002530  6$:     GPRML   QS6,12,1,YES
002530  005130                                                  .WORD   T$CODE
002532  003147                                                  .WORD   QS6
002534  000001                                                  .WORD   1
002536                  XFERT   7$
002536  007024                                                  .WORD   T$CODE
002540                  GPRMD   QS11,24,D,77,1,10..,YES
002540  012052                                                  .WORD   T$CODE
002542  003554                                                  .WORD   QS11
002544  000077                                                  .WORD   77
002546  000001                                                  .WORD   T$LOLIM
002550  000012                                                  .WORD   T$HILIM
002552                  XFER    13$
```

```
                002552  026004
    325         002554
                002554  006130
                002556  003454
                002560  000001
    326         002562
                002562  007130
                002564  003474
                002566  000001
    327         002570
                002570  010130
                002572  003514
                002574  000001
    328         002576
                002576  011130
                002600  003534
                002602  000001
    329         002604
                002604  006024
    330         002606
                002606  012052
                002610  003554
                002612  000077
                002614  000001
                002616  000012
    331         002620                  12$:    GPRML   QS12,26,1,YES
                002620  013130
                002622  003637
                002624  000001
    332         002626                  13$:    GPRML   QS13,30,1,YES
                002626  014130
                002630  003657
                002632  000001
    333         002634                          GPRML   QS14,32,1,YES
                002634  015130
                002636  003724
                002640  000001
    334         002642                          GPRML   QS15,34,1,YES
                002642  016130
                002644  003763
                002646  000001
    335         002650                          GPRML   QS16,36,1,YES
                002650  017130
                002652  004030
                002654  000001
    336         002656                          GPRML   QS17,40,1,YES
                002656  020130
                002660  004066
                002662  000001
    337                                         .EVEN
    338
    339         002664                          ENDSFT

                002664


                                                .WORD   T$CODE

                                                .WORD   T$CODE
                                                .WORD   QS7
                                                .WORD   1

                                                .WORD   T$CODE
                                                .WORD   QS8
                                                .WORD   1

                                                .WORD   T$CODE
                                                .WORD   QS9
                                                .WORD   1

                                                .WORD   T$CODE
                                                .WORD   QS10
                                                .WORD   1

                                                .WORD   T$CODE

                                                .WORD   T$CODE
                                                .WORD   QS11
                                                .WORD   77
                                                .WORD   T$LOLIM
                                                .WORD   T$HILIM

                                                .WORD   T$CODE
                                                .WORD   QS12
                                                .WORD   1

                                                .WORD   T$CODE
                                                .WORD   QS13
                                                .WORD   1

                                                .WORD   T$CODE
                                                .WORD   QS14
                                                .WORD   1

                                                .WORD   T$CODE
                                                .WORD   QS15
                                                .WORD   1

                                                .WORD   T$CODE
                                                .WORD   QS16
                                                .WORD   1

                                                .WORD   T$CODE
                                                .WORD   QS17
                                                .WORD   1


                                                .EVEN
                                        L10003:
```

```
347 002664     114     111     115  QS1:     .ASCIZ  /LIMIT RANGE OF SECTORS TO BE TESTED ?/
348 002732     104     117     040  QS2:     .ASCIZ  /DO YOU KNOW EXACT RANGE OF SECTORS TO TEST ?/
349 003007     106     111     122  QS3:     .ASCIZ  /FIRST  SECTOR  TO  BE  TESTED  ?/
350 003047     114     101     123  QS4:     .ASCIZ  /LAST   SECTOR  TO  BE  TESTED  ?/
351 003107     127     110     111  QS5:     .ASCIZ  /WHICH  BOARD   {0-15}  TESTED  ?/
352 003147     124     110     105  QS6:     .ASCII  /THE PROGRAM OPTIONS INCLUDE:/
353 003203     012     015     040           .ASCII  <12><15>/   1.  ADDRESS CHECK/
354 003231     012     015     040           .ASCII  <12><15>/   2.  PATTERN TEST/
355 003256     012     015     040           .ASCII  <12><15>/   3.  UNIQUE DATA CHECK/
356 003310     012     015     040           .ASCII  <12><15>/   4.  MARCH TEST/
357 003333     012     015     040           .ASCII  <12><15>/   5.  RANDOMNESS TESTS/
358 003364     012     015     104           .ASCIZ  <12><15>/DO YOU WANT TO DROP ANY OF THESE FROM THE EXERCISER ?/
359 003454     104     122     117  QS7:     .ASCIZ  /DROP OPTION 1 ?/
360 003474     104     122     117  QS8:     .ASCIZ  /DROP OPTION 2 ?/
361 003514     104     122     117  QS9:     .ASCIZ  /DROP OPTION 3 ?/
362 003534     104     122     117  QS10:    .ASCIZ  /DROP OPTION 4 ?/
363 003554     120     101     124  QS11:    .ASCIZ  /PATTERN NUMBER {1-10} TO BE USED WITH MARCH TEST ?/
364 003637     104     122     117  QS12:    .ASCIZ  /DROP OPTION 5 ?/
365 003657     105     116     101  QS13:    .ASCIZ  /ENABLE REFRESH MARGINING            ?/
366 003724     104     111     123  QS14:    .ASCIZ  /DISABLE ERROR CORRECTION            ?/
367 003763     105     116     101  QS15:    .ASCIZ  /ENABLE END OF PASS SUMMARY PRINTOUT?/
368 004030     105     116     101  QS16:    .ASCIZ  /ENABLE ERROR PRINTOUTS              ?/
369 004066     105     116     101  QS17:    .ASCIZ  /ENABLE SOFT ERROR TESTING           ?/
370                                          .EVEN
371
372
373
374                                 ;++
375                                 ; THE PROTECTION TABLE IS USED BY THE RUNTIME
376                                 ; SERVICES TO PROTECT THE LOAD MEDIA.
377                                 ;--
378
379 004126                                   BGNPROT
    004126                                                                            L$PROT::
380
381 004126  177777                           -1              ;OFFSET INTO P-TABLE FOR CSR ADDRESS
382 004130  177777                           -1              ;OFFSET INTO P-TABLE FOR MASSBUS ADDRESS
383 004132  177777                           -1              ;OFFSET INTO P-TABLE FOR DRIVE NUMBER
384
385 004134                                   ENDPROT
386
400
401
402 004134                          $PATCH:: .BLKW  32.      ;THIS LEAVES ROOM FOR THE 22 ML-11 REGISTERS TOO
403
410
411 004234                                   ENDMOD
412
413
414
415
428
```

```
                                .SBTTL  'RANDOM NUMBER GENERATOR'

                                ;ROUTINE TO GENERATE 16-BIT PSEUDO RANDOM NUMBERS

                                ;INPUTS:                NONE
                                ;IMPLIED INPUTS:        SEED1, SEED2, SEED3 are global values which will
                                ;                       be used by, but not input to the generator.
                                ;OUTPUTS:               'RANDOM'--WORD CONTAINING RANDOM 16-BIT INTEGER VALUE
                                ;IMPLIED OUTPUTS:       NONE
                                ;CALLING SEQUENCE:      JSR PC,RN

005370  010046                RN::    MOV     R0,-(SP)
005372  016700  000062                MOV     SEED2,R0
005376  000241                        CLC
005400  005367  000052                DEC     SEED1
005404  006100                        ROL     R0
005406  006100                        ROL     R0
005410  066700  000042                ADD     SEED1,R0
005414  066700  000042                ADD     SEED3,R0
005420  010067  000034                MOV     R0,SEED2
005424  006100                        ROL     R0
005426  006100                        ROL     R0
005430  066700  000026                ADD     SEED3,R0
005434  006100                        ROL     R0
005436  006100                        ROL     R0
005440  010067  000016                MOV     R0,SEED3
005444  016767  000010  000012        MOV     SEED2,RANDOM            ;SAVE NEW R.N.
005452  012600                        MOV     (SP)+,R0
005454  000207                        RTS     PC


005456  000000                SEED1:: .WORD   0
005460  001233                SEED2:: .WORD   1233
005462  007622                SEED3:: .WORD   7622

005464  000000                RANDOM:: .WORD  0
```

```
;MLX3                                                    25-Mar-1982 22:23:37    TOPS-20 Bliss-16 V2(212)
;                                                        25-Mar-1982 22:21:30    PA:<NEALE>MLX3.BLI.1 (1)

;          0001  MODULE MLX3 =
;          0002  BEGIN
;          0003
;          0004  REQUIRE 'BLSMAC.REQ';
;          1494
;          1495  %SBTTL 'REPORT CODING SECTION'
;          1496
;          1497  EXTERNAL ROUTINE
;          1498     EOP: NOVALUE;
;          1499
;          1500  BGNRPT;
;          1501
;          1502  EOP();
;          1503
;          1504  RETURN;
;          1505
;          1506  ENDRPT;


                                        .GLOBL  EOP


                                        .SBTTL  LRPT REPORT CODING SECTION

005466  004767  075234          LRPT:   JSR     PC,EOP                          ;                       1502
005472  000207                          RTS     PC                              ;                       1498

                                ; Routine Size:  3 words
                                ; Maximum stack depth per invocation:  0 words


                                        .SBTTL  L$RPT REPORT CODING SECTION
005474  004767  177766          L$RPT:: JSR     PC,LRPT                         ;                       1504
                                ;MLX3                                                   25-Mar-1982 22:23:37    TOPS
                                ;                       REPORT CODING SECTION           25-Mar-1982 22:21:30    PA:<
005500  104425                          TRAP    25
005502  000207                          RTS     PC

                                ; Routine Size:  4 words
                                ; Maximum stack depth per invocation:  0 words
```

```
     74 ;MLX3
     75 ;                    AUTODROP SECTION                                    25-Mar-1982 22:23:37    TOPS-20 Bliss-16 V2(212)
     76 ;                                                                        25-Mar-1982 22:21:30    PA:<NEALE>MLX3.BLI.1 (2)
     77 ;            1507  %SBTTL 'AUTODROP SECTION'
     78 ;            1508
     79 ;            1509  !++
     80 ;            1510  !  THIS SECTION IS OPTIONALLY EXECUTED IMMEDIATELY AFTER THE INITIALIZATION
     81 ;            1511  !  CODE IF THE /ADR FLAG WAS SET.  THE INTENT IS TO EXAMINE THE UNITS UNDER
     82 ;            1512  !  TEST TO SEE IF THEY WILL RESPOND.  THOSE THAT DON'T RESPOND ARE DROPPED.
     83 ;            1513  !
     84 ;            1514  !  THIS FUNCTION IS AN INTEGRAL PART OF THE INITIALIZATION CODE ITSELF, AND
     85 ;            1515  !  THEREFORE A SEPARATE AUTODROP SECTION IS REDUNDANT AND NOT PROVIDED.
     86 ;            1516  !--
     87 ;            1517
     88 ;            1518  BGNAUTO;
     89 ;            1519
     90 ;            1520  RETURN;
     91 ;            1521
     92 ;            1522  ENDAUTO;
     96
     97                                        .SBTTL  LAUTO AUTODROP SECTION
    101 005504  000207                  LAUTO: RTS     PC                          ;                                   1506
    102
    103                                 ; Routine Size:  1 word
    104                                 ; Maximum stack depth per invocation:  0 words
    109
    110
    114
    115                                        .SBTTL  L$AUTO AUTODROP SECTION
    119 005506  004767  177772          L$AUTO::JSR     PC,LAUTO                    ;                                   1520
    120 005512  104461                         TRAP    61
    121 005514  000207                         RTS     PC
    122
    123                                 ; Routine Size:  4 words
    124                                 ; Maximum stack depth per invocation:  0 words
```

```
;MLX3
;                DROP UNIT SECTION                                  25-Mar-1982 22:23:37    TOPS-20 Bliss-16 V2(212)
;                                                                   25-Mar-1982 22:21:30    PA:<NEALE>MLX3.BLI.1 (3)

;          1523  %SBTTL 'DROP UNIT SECTION'
;          1524
;          1525  !++
;          1526  !  THE DROP-UNIT SECTION CONTAINS THE CODING THAT CAUSES A DRIVE TO NO
;          1527  !  LONGER BE TESTED.  WHEN THIS HAPPENS, THE FOLLOWING ACTIONS WILL BE
;          1528  !  TAKEN:
;          1529  !
;          1530  !  (1) SET THE DRIVE STATUS TO INACTIVE AND SET THE DRIVE'S DROP FLAG.
;          1531  !
;          1532  !  (2) DECREMENT THE NUMBER OF ACTIVE DRIVES, AND IF ZERO ABORT PASS.
;          1533  !--
;          1534
;          1535  BGNDU;
;          1536
;          1537  !+
;          1538  ! DELCARE THE EXTERNAL SUMMARY REPORT CODE
;          1539  ! DATA STORAGE LOCATIONS TO THIS SECTION
;          1540  !-
;          1541
;          1542  EXTERNAL
;          1543     LSLUN,                              !DEFINES NUMBER OF UUT'S ATTACHED
;          1544     DRIVE_STATUS:BITVECTOR[8],          !DRIVE STATUS FLAG REGISTER
;          1545     DROPT_DRIVES:BITVECTOR[8],          !DRIVE DROP STATUS FLAG REGISTER
;          1546     NUM_DRIVES;                         !INDICATES THE NUMBER OF ATTACHED UUT'S TO TEST
;          1547
;          1548
;          1549  LITERAL
;          1550     INACTIVE = 0,                       !CODE TO DESELECT UUT'S
;          1551     ACTIVE = 1;                         !CODE TO SELECT UUT'S
;          1552
;          1553
;          1554  !+
;          1555  ! THIS SECTION IS CALLED FOR THE PURPOSE OF
;          1556  ! DROPPING THIS UUT FROM FURTHER TESTING.
;          1557  !
;          1558  ! THEREFORE CLEAR THIS UUT'S DRIVE ACTIVE FLAG BIT
;          1559  ! AND SET THIS UUT'S DRIVE DROPPED FLAG.
;          1560  !-
;          1561
;          1562  DRIVE_STATUS[.L$LUN] = INACTIVE;        !CLEAR THE DRIVE ACTIVE FLAG
;          1563  DROPT_DRIVES[.L$LUN] = ACTIVE;          !SET THE DRIVES DROPPED FLAG
;          1564
;          1565  !+
;          1566  ! DECRIMENT THE GLOBAL VARIABLE 'NUM_DRIVES' WHICH
;          1567  ! STORES THE NUMBER OF UUT'S WHICH ARE TESTED BY THIS
;          1568  ! EXERCISER.
;          1569  !
;          1570  ! IF THE NUMBER OF DRIVES REMAINING TO TEST IS ZERO
;          1571  ! THEN CALL THE CLEAN UP CODE SECTION AND EXIT THIS
;          1572  ! PROGRAM ELSE CONTINUE TESTING THE REMAINING ACTIVE
;          1573  ! UUT'S.
;          1574  !-
;MLX3
;                DROP UNIT SECTION                                  25-Mar-1982 22:23:37    TOPS-20 Bliss-16 V2(212)
;                                                                   25-Mar-1982 22:21:30    PA:<NEALE>MLX3.BLI.1 (3)
```

```
187
188 ;          1575
189 ;          1576  NUM_DRIVES = .NUM_DRIVES - 1;           !DECRIMENT THE NUMBER OF DRIVES REMAINING TO TEST
190 ;          1577
191 ;          1578
192 ;          1579  IF .NUM_DRIVES EQL 0 THEN  DOCLN;       !DO CLEAN UP CODE AND EXIT PROG IF ZERO
193 ;          1580
194 ;          1581  RETURN;
195 ;          1582
196 ;          1583  ENDDU;
200
201                                      .GLOBL  L$LUN, DRIVE.STATUS, DROPT.DRIVES
202                                      .GLOBL  NUM.DRIVES
203
204
205                                      .SBTTL  LDU DROP UNIT SECTION
```

SEQ 0044

```
                                ;+
                                ; LDU DROP UNIT SECTION
                                ;
                                ; Routine Size:  41 words
                                ; Maximum stack depth per invocation:  7 words
                                ;-
                                .SBTTL  LSDU DROP UNIT SECTION
                                ;+
                                ; LSDU DROP UNIT SECTION
                                ;
                                ; Routine Size:  4 words
                                ; Maximum stack depth per invocation:  0 words
                                ;-
```

```
 210 005516  016700  174352     LDU:    MOV     L$LUN,R0                        ;  1562
 211 005522  006200                     ASR     R0
 212 005524  006200                     ASR     R0
 213 005526  006200                     ASR     R0
 214 005530  062700  034442             ADD     #DRIVE.STATUS,R0
 215 005534  010046                     MOV     R0,-(SP)
 216 005536  016746  174332             MOV     L$LUN,-(SP)
 217 005542  042716  177770             BIC     #177770,(SP)
 218 005546  012746  000001             MOV     #1,-(SP)
 219 005552  005046                     CLR     -(SP)
 220 005554  004767  177034             JSR     PC,BL$PU2
 221 005560  016700  174310             MOV     L$LUN,R0                        ;  1563
 222 005564  006200                     ASR     R0
 223 005566  006200                     ASR     R0
 224 005570  006200                     ASR     R0
 225 005572  062700  034444             ADD     #DROPT.DRIVES,R0
 226 005576  010016                     MOV     R0,(SP)
 227 005600  016746  174270             MOV     L$LUN,-(SP)
 228 005604  042716  177770             BIC     #177770,(SP)
 229 005610  012746  000001             MOV     #1,-(SP)
 230 005614  011646                     MOV     (SP),-(SP)
 231 005616  004767  176772             JSR     PC,BL$PU2
 232 005622  005367  026630             DEC     NUM.DRIVES                      ;  1576
 233 005626  001001                     BNE     1$                              ;  1579
 234 005630  104444                     TRAP    44
 235 005632  062706  000016     1$:     ADD     #16,SP                          ;  1522
 236 005636  000207                     RTS     PC
 237
 238
 239
 247
 248
 252
 253
 257 005640  004767  177652     LSDU::  JSR     PC,LDU                          ;  1581
 258 005644  104453                     TRAP    53
 259 005646  000207                     RTS     PC
 260
 261
 262
 267
 268
```

```
    270 ;MLX3
    271 ;                   ADD UNIT SECTION                    25-Mar-1982 22:23:37    TOPS-20 BLiss-16 V2(212)
    272 ;                                                       25-Mar-1982 22:21:30    PA:<NEALE>MLX3.BLI.1 (4)
    273 ;           1584    %SBTTL 'ADD UNIT SECTION'
    274 ;           1585
    275 ;           1586    !++
    276 ;           1587    !  THE ADD UNIT SECTION CONTAINS ANY CODE THE PROGRAMMER WISHES
    277 ;           1588    !  TO BE EXECUTED IN CONJUNCTION WITH THE ADDING OF A UNIT BACK
    278 ;           1589    !  TO THE TEST CYCLE.   SINCE THE INITIALIZATION CODE WILL HAVE
    279 ;           1590    !  TO BE EXECUTED IMMEDIATELY AFTER AN 'ADD',  THERE IS NO NEED
    280 ;           1591    !  FOR ANY CODE IN THIS SECTION.
    281 ;           1592    !--
    282 ;           1593
    283 ;           1594    BGNAU;
    284 ;           1595
    285 ;           1596    RETURN;
    286 ;           1597
    287 ;           1598    ENDAU;
    291
    292                                 .SBTTL  LAU ADD UNIT SECTION
    296 005650  000207          LAU:    RTS     PC                          ;                                   1583
    297
```

```
                                ; Routine Size:  1 word
                                ; Maximum stack depth per invocation:  0 words
                                
                                
                                
                                        .SBTTL  LSAU ADD UNIT SECTION
005652  004767  177772          LSAU::  JSR     PC,LAU                  ;                1596
005656  104452                          TRAP    52
005660  000207                          RTS     PC
                                
                                ; Routine Size:  4 words
                                ; Maximum stack depth per invocation:  0 words
;               1599
;MLX3
:                       ADD UNIT SECTION                        25-Mar-1982 22:23:37    TOPS-20 Bliss-16 V2(212)
                                                                25-Mar-1982 22:21:30    PA:<NEALE>MLX3.BLI.1 (4)
;               1600
;               1601    END
;               1602
;               1603    ELUDOM
```

```
                        ; OTS external references
                                .GLOBL  BL$PU2
;           1604
;           1605
; Size:             62 code + 0 data words
; Run Time:         00:03.3
; Elapsed Time:     00:05.2
; Memory Used:      26 pages
; Compilation Complete
```

```
;MLX4                                                    27-Mar-1982 19:24:42    TOPS-20 Bliss-16 V2(212)
;                                                        27-Mar-1982 19:23:44    PA:<NEALE>MLX4.BLI.5 (1)

;          0001  MODULE MLX4 =
;          0002  BEGIN
;          0003  !
;          0004  ! PERTTY BLF COMMANDS
;          0005  !
;          0006  !<BLF/NOERRORS>
;          0007  !<BLF/LOWERCASE_KEY>
;          0008  !
;          0009
;          0010  require 'BLSMAC.REQ';
;          1500
;MLX4                                                    27-Mar-1982 19:24:42    TOPS-20 Bliss-16 V2(212)
;                VARIABLES AND CONSTANTS                 27-Mar-1982 19:23:44    PA:<NEALE>MLX4.BLI.5 (2)

;          1501  %sbttl 'VARIABLES AND CONSTANTS'
;          1502
;          1503  Literal
;          1504  !
;          1505  ! VER CZMLBB ADDED LITERIAL TRUE, FALSE AND ZERO
;          1506  !
;          1507      ZERO = %o'000000',                  !DEFINE ZERO DATA
;          1508      TRUE = 1,                           !LOGICAL TRUE INDICATOR
;          1509      FALSE = 0,                          !LOGICAL FALSE INDICATOR
;          1510  !
;          1511  ! OTHER LITERIAL DEFINITIONS
;          1512  !
;          1513      ONE = 1,                            !NUMBER OF TIMES TO RETRY FOR DATA ERROR
;          1514      SIX = 6,                            !NUMBER OF TIMES TO RETRY FOR SYSTEM ERROR
;          1515      NUM_PATS = 10,                      !NUMBER OF REGULAR PATTERNS
;          1516      NUM_REGS = 22,                      !NUMBER OF ML11 REGISTERS
;          1517      TIMES_TO_LOOP = 2,                  !LOOP READING CONSTANT
;          1518      BUFSIZ = 256*8,                     !2048 16-BIT WORDS IN A FULL BUFFER
;          1519      !256 = NUMBER OF WORDS IN A SECTOR
;          1520      !  8 = NUMBER OF SECTORS IN THE BUFFER
;          1521      !
;          1522      !  ML11 DRIVE TYPES:
;          1523      !
;          1524      ML11A = %o'000110',                 !16K CHIPS
;          1525      ML11B = %o'000111',                 !64K CHIPS
;          1526      !
;          1527      !  FUNCTION CODES:
;          1528      !
;          1529      DRV_CLR = %o'11',
;          1530      WC_CMD = %o'51',
;          1531      WR_CMD = %o'61',
;          1532      RD_CMD = %o'71',
;          1533      !
;          1534      !  STATUS CODES:
;          1535      !
;          1536      INACTIVE = 0,
;          1537      ACTIVE = 1,
;          1538      !
;          1539      !  ERROR THRESHOLD VALUES:
;          1540      !
```

```
     63 ;          1541      S16K_LIMIT = 10,                          !SOFT ERROR, 16K ARRAYS
     64 ;          1542      H16K_LIMIT = 10,                          !HARD ERROR, 16K ARRAYS
     65 ;          1543      S64K_LIMIT = 10,                          !SOFT ERROR, 64K ARRAYS
     66 ;          1544      H64K_LIMIT = 10,                          !HARD ERROR, 64K ARRAYS
     67 ;          1545                                                !
     68 ;          1546      !  SUMMARY OF ERROR CODES:
     69 ;          1547      !
     70 ;          1548      !
     71 ;          1549      ! THE 'INTEGRITY' ROUTINE:
     72 ;          1550      !
     73 ;          1551      !WRITE COMMAND:
     74 ;          1552      !ERRDF(1,MSG1,0);   !**** INTEGRITY ROUTINE ERROR 01 ****
```

```
;MLX4
;                    VARIABLES AND CONSTANTS                    27-Mar-1982 19:24:42    TOPS-20 Bliss-16 V2(212)
;                                                               27-Mar-1982 19:23:44    PA:<NEALE>MLX4.BLI.5 (2)

;          1553      !ERRDF(2,MSG1,0);    !**** INTEGRITY ROUTINE ERROR 02 ****
;          1554      !ERRDF(3,MSG1,0);    !**** INTEGRITY ROUTINE ERROR 03 ****
;          1555      !READ COMMAND:
;          1556      !ERRDF(4,MSG1,0);    !**** INTEGRITY ROUTINE ERROR 04 ****
;          1557      !ERRDF(5,MSG1,0);    !**** INTEGRITY ROUTINE ERROR 05 ****
;          1558      !ERRDF(6,MSG1,0);    !**** INTEGRITY ROUTINE ERROR 06 ****
;          1559      !ERRDF(7,MSG1,0);    !**** INTEGRITY ROUTINE ERROR 07 ****
;          1560      !ERRDF(8,MSG2,0);    !**** INTEGRITY ROUTINE ERROR 08 ****
;          1561      !ERRHRD(9,MSG4,0);   !**** INTEGRITY ROUTINE ERROR 09 ****
;          1562      !ERRSOFT(10,MSG3,0);!**** INTEGRITY ROUTINE ERROR 10 ****
;          1563      !ERRSOFT(11,MSG3,1);!**** INTEGRITY ROUTINE ERROR 10 ****
;          1564      !WRITE CHECK COMMAND:
;          1565      !ERRDF(12,MSG1,0);   !**** INTEGRITY ROUTINE ERROR 12 ****
;          1566      !ERRDF(13,MSG1,0);   !**** INTEGRITY ROUTINE ERROR 13 ****
;          1567      !ERRDF(14,MSG1,0);   !**** INTEGRITY ROUTINE ERROR 14 ****
;          1568      !ERRDF(15,MSG2,0);   !**** INTEGRITY ROUTINE ERROR 15 ****
;          1569      !ERRHRD(16,MSG4,0);  !**** INTEGRITY ROUTINE ERROR 16 ****
;          1570      !ERRSOFT(17,MSG3,8);!**** INTEGRITY ROUTINE ERROR 17 ****
;          1571      !ERRSOFT(18,MSG3,1);!**** INTEGRITY ROUTINE ERROR 18 ****
;          1572      !
;          1573      ! OPTION 1:
;          1574      !
;          1575      !WRITE COMMAND:
;          1576      !ERRDF(101,MSG1,0);    !**** OPTION 1 ERROR 01 ****
;          1577      !ERRDF(102,MSG1,0);    !**** OPTION 1 ERROR 02 ****
;          1578      !ERRDF(103,MSG1,0);    !**** OPTION 1 ERROR 03 ****
;          1579      !CHECK OR READ:
;          1580      !ERRDF(104,MSG1,0);    !**** OPTION 1 ERROR 04 ****
;          1581      !ERRDF(105,MSG1,0);    !**** OPTION 1 ERROR 05 ****
;          1582      !ERRDF(106,MSG1,0);    !**** OPTION 1 ERROR 06 ****
;          1583      !ERRDF(107,MSG1,0);    !**** OPTION 1 ERROR 07 ****
;          1584      !ERRDF(108,MSG2,0);    !**** OPTION 1 ERROR 08 ****
;          1585      !ERRHRD(109,MSG4,0);   !**** OPTION 1 ERROR 09 ****
;          1586      !ERRSOFT(110,MSG3,0);  !**** OPTION 1 ERROR 10 ****
;          1587      !ERRSOFT(111,MSG3,0);  !**** OPTION 1 ERROR 11 ****
;          1588                                                    !
;          1589      ! OPTION 2:
;          1590      !
;          1591      !WRITE COMMAND:
;          1592      !ERRDF(201,MSG1,0);    !**** OPTION 2 ERROR 01 ****
;          1593      !ERRDF(202,MSG1,0);    !**** OPTION 2 ERROR 02 ****
;          1594      !ERRDF(203,MSG1,0);    !**** OPTION 2 ERROR 03 ****
;          1595      !CHECK OR READ:
;          1596      !ERRDF(204,MSG1,0);    !**** OPTION 2 ERROR 04 ****
;          1597      !ERRDF(205,MSG1,0);    !**** OPTION 2 ERROR 05 ****
;          1598      !ERRDF(206,MSG1,0);    !**** OPTION 2 ERROR 06 ****
;          1599      !ERRDF(207,MSG1,0);    !**** OPTION 2 ERROR 07 ****
;          1600      !ERRDF(208,MSG2,0);    !**** OPTION 2 ERROR 08 ****
;          1601      !ERRHRD(209,MSG4,0);   !**** OPTION 2 ERROR 09 ****
;          1602      !ERRSOFT(210,MSG3,0);  !**** OPTION 2 ERROR 10 ****
;          1603      !ERRSOFT(211,MSG3,1);  !**** OPTION 2 ERROR 10 ****
;          1604      !LOOP CHECK OR READ:
```

```
132 ;MLX4
133 ;                    VARIABLES AND CONSTANTS
134 ;
135 ;          1605      !ERRDF(212,MSG1,0);    !**** OPTION 2 ERROR 12 ****
136 ;          1606      !ERRDF(213,MSG1,0);    !**** OPTION 2 ERROR 13 ****
137 ;          1607      !ERRDF(214,MSG1,0);    !**** OPTION 2 ERROR 14 ****
138 ;          1608      !ERRDF(215,MSG1,0);    !**** OPTION 2 ERROR 15 ****
139 ;          1609      !ERRDF(216,MSG2,0);    !**** OPTION 2 ERROR 16 ****
140 ;          1610      !ERRHRD(217,MSG4,0);   !**** OPTION 2 ERROR 17 ****
141 ;          1611      !ERRSOFT(218,MSG3,0);  !**** OPTION 2 ERROR 18 ****
142 ;          1612      !ERRSOFT(219,MSG3,0);  !**** OPTION 2 ERROR 19 ****
143 ;          1613      !
144 ;          1614      ! OPTION 3:
145 ;          1615      !
146 ;          1616      !WRITE COMMAND:
147 ;          1617      !ERRDF(301,MSG1,0);    !**** OPTION 3 ERROR 01 ****
148 ;          1618      !ERRDF(302,MSG1,0);    !**** OPTION 3 ERROR 02 ****
149 ;          1619      !ERRDF(303,MSG1,0);    !**** OPTION 3 ERROR 03 ****
150 ;          1620      !CHECK OR READ:
151 ;          1621      !ERRDF(304,MSG1,0);    !**** OPTION 3 ERROR 04 ****
152 ;          1622      !ERRDF(305,MSG1,0);    !**** OPTION 3 ERROR 05 ****
153 ;          1623      !ERRDF(306,MSG1,0);    !**** OPTION 3 ERROR 06 ****
154 ;          1624      !ERRDF(307,MSG1,0);    !**** OPTION 3 ERROR 07 ****
155 ;          1625      !ERRDF(308,MSG2,0);    !**** OPTION 3 ERROR 08 ****
156 ;          1626      !ERRHRD(309,MSG4,0);   !**** OPTION 3 ERROR 09 ****
157 ;          1627      !ERRSOFT(310,MSG3,0);  !**** OPTION 3 ERROR 10 ****
158 ;          1628      !ERRSOFT(311,MSG3,0);  !**** OPTION 3 ERROR 11 ****
159 ;          1629      !
160 ;          1630      ! OPTION 4:
161 ;          1631      !
162 ;          1632      !MARCHING UP:
163 ;          1633      !WRITE DATA:
164 ;          1634      !ERRDF(401,MSG1,0);    !**** OPTION 4 ERROR 01 ****
165 ;          1635      !ERRDF(402,MSG1,0);    !**** OPTION 4 ERROR 02 ****
166 ;          1636      !ERRDF(403,MSG1,0);    !**** OPTION 4 ERROR 03 ****
167 ;          1637      !MARCHING UP:
168 ;          1638      !CHECK OR READ DATA:
169 ;          1639      !ERRDF(404,MSG1,0);    !**** OPTION 4 ERROR 04 ****
170 ;          1640      !ERRDF(405,MSG1,0);    !**** OPTION 4 ERROR 05 ****
171 ;          1641      !ERRDF(406,MSG1,0);    !**** OPTION 4 ERROR 06 ****
172 ;          1642      !ERRDF(407,MSG1,0);    !**** OPTION 4 ERROR 07 ****
173 ;          1643      !ERRDF(408,MSG2,0);    !**** OPTION 4 ERROR 08 ****
174 ;          1644      !ERRHRD(409,MSG4,0);   !**** OPTION 4 ERROR 09 ****
175 ;          1645      !ERRSOFT(410,MSG3,0);  !**** OPTION 4 ERROR 10 ****
176 ;          1646      !ERRSOFT(411,MSG3,0);  !**** OPTION 4 ERROR 11 ****
177 ;          1647      !WRITE COMP:
178 ;          1648      !ERRDF(412,MSG1,0);    !**** OPTION 4 ERROR 12 ****
179 ;          1649      !ERRDF(413,MSG1,0);    !**** OPTION 4 ERROR 13 ****
180 ;          1650      !ERRDF(414,MSG1,0);    !**** OPTION 4 ERROR 14 ****
181 ;          1651      !MARCHING DOWN:
182 ;          1652      !CHECK OR READ COMP:
183 ;          1653      !ERRDF(415,MSG1,0);    !**** OPTION 4 ERROR 15 ****
184 ;          1654      !ERRDF(416,MSG1,0);    !**** OPTION 4 ERROR 16 ****
185 ;          1655      !ERRDF(417,MSG1,0);    !**** OPTION 4 ERROR 17 ****
186 ;          1656      !ERRDF(418,MSG1,0);    !**** OPTION 4 ERROR 18 ****
```

```
;MLX4
;                    VARIABLES AND CONSTANTS                    27-Mar-1982 19:24:42    TOPS-20 Bliss-16 V2(212)
;                                                               27-Mar-1982 19:23:44    PA:<NEALE>MLX4.BLI.5 (2)

;          1657      !ERRDF(419,MSG2,0);    !**** OPTION 4 ERROR 19 ****
;          1658      !ERRHRD(420,MSG4,0);   !**** OPTION 4 ERROR 20 ****
;          1659      !ERRSOFT(421,MSG3,0);  !**** OPTION 4 ERROR 21 ****
;          1660      !ERRSOFT(422,MSG3,0);  !**** OPTION 4 ERROR 22 ****
;          1661      !WRITE DATA:
;          1662      !ERRDF(423,MSG1,0);    !**** OPTION 4 ERROR 23 ****
;          1663      !ERRDF(424,MSG1,0);    !**** OPTION 4 ERROR 24 ****
;          1664      !ERRDF(425,MSG1,0);    !**** OPTION 4 ERROR 25 ****
;          1665      !MARCHING UP:
;          1666      !CHECK OR READ DATA:
;          1667      !ERRDF(426,MSG1,0);    !**** OPTION 4 ERROR 26 ****
;          1668      !ERRDF(427,MSG1,0);    !**** OPTION 4 ERROR 27 ****
;          1669      !ERRDF(428,MSG1,0);    !**** OPTION 4 ERROR 28 ****
;          1670      !ERRDF(429,MSG1,0);    !**** OPTION 4 ERROR 29 ****
;          1671      !ERRDF(430,MSG2,0);    !**** OPTION 4 ERROR 30 ****
;          1672      !ERRHRD(431,MSG4,0);   !**** OPTION 4 ERROR 31 ****
;          1673      !ERRSOFT(432,MSG3,0);  !**** OPTION 4 ERROR 32 ****
;          1674      !ERRSOFT(433,MSG3,0);  !**** OPTION 4 ERROR 33 ****
;          1675      !
;          1676      ! OPTION 5:
;          1677      !
;          1678      !'RAND1' ROUTINE:
;          1679      !WRITE COMMAND:
;          1680      !ERRDF(5101,MSG1,0);    !**** OPTION 5, RAND1 ERROR 01 ****
;          1681      !ERRDF(5102,MSG1,0);    !**** OPTION 5, RAND1 ERROR 02 ****
;          1682      !ERRDF(5103,MSG1,0);    !**** OPTION 5, RAND1 ERROR 03 ****
;          1683      !CHECK OR READ:
;          1684      !ERRDF(5104,MSG1,0);    !**** OPTION 5, RAND1 ERROR 04 ****
;          1685      !ERRDF(5105,MSG1,0);    !**** OPTION 5, RAND1 ERROR 05 ****
;          1686      !ERRDF(5106,MSG1,0);    !**** OPTION 5, RAND1 ERROR 06 ****
;          1687      !ERRDF(5107,MSG1,0);    !**** OPTION 5, RAND1 ERROR 07 ****
;          1688      !ERRDF(5108,MSG2,0);    !**** OPTION 5, RAND1 ERROR 08 ****
;          1689      !ERRHRD(5109,MSG4,0);   !**** OPTION 5, RAND1 ERROR 09 ****
;          1690      !ERRSOFT(5110,MSG3,0);  !**** OPTION 5, RAND1 ERROR 10 ****
;          1691      !ERRSOFT(5111,MSG3,0);  !**** OPTION 5, RAND1 ERROR 11 ****
;          1692      !'RAND2' ROUTINE:
;          1693      !WRITE COMMAND:
;          1694      !ERRDF(5201,MSG1,0);    !**** OPTION 5, RAND2 ERROR 01 ****
;          1695      !ERRDF(5202,MSG1,0);    !**** OPTION 5, RAND2 ERROR 02 ****
;          1696      !ERRDF(5203,MSG1,0);    !**** OPTION 5, RAND2 ERROR 03 ****
;          1697      !CHECK OR READ:
;          1698      !ERRDF(5204,MSG1,0);    !**** OPTION 5, RAND2 ERROR 04 ****
;          1699      !ERRDF(5205,MSG1,0);    !**** OPTION 5, RAND2 ERROR 05 ****
;          1700      !ERRDF(5206,MSG1,0);    !**** OPTION 5, RAND2 ERROR 06 ****
;          1701      !ERRDF(5207,MSG1,0);    !**** OPTION 5, RAND2 ERROR 07 ****
;          1702      !ERRDF(5208,MSG2,0);    !**** OPTION 5, RAND2 ERROR 08 ****
;          1703      !ERRHRD(5209,MSG4,0);   !**** OPTION 5, RAND2 ERROR 09 ****
;          1704      !ERRSOFT(5210,MSG3,0);  !**** OPTION 5, RAND2 ERROR 10 ****
;          1705      !ERRSOFT(5211,MSG3,0);  !**** OPTION 5, RAND2 ERROR 11 ****
;          1706      !'RAND3' ROUTINE:
;          1707      !WRITE COMMAND:
;          1708      !ERRDF(5301,MSG1,0);    !**** OPTION 5, RAND3 ERROR 01 ****
```

SEQ 0053


```
244 ;MLX4
245 ;                   VARIABLES AND CONSTANTS
246 ;
247 ;          1709         !ERRDF(5302,MSG1,0);    !**** OPTION 5, RAND3 ERROR 02 ****
248 ;          1710         !ERRDF(5303,MSG1,0);    !**** OPTION 5, RAND3 ERROR 03 ****
249 ;          1711         !CHECK OR READ:
250 ;          1712         !ERRDF(5304,MSG1,0);    !**** OPTION 5, RAND3 ERROR 04 ****
251 ;          1713         !ERRDF(5305,MSG1,0);    !**** OPTION 5, RAND3 ERROR 05 ****
252 ;          1714         !ERRDF(5306,MSG1,0);    !**** OPTION 5, RAND3 ERROR 06 ****
253 ;          1715         !ERRDF(5307,MSG1,0);    !**** OPTION 5, RAND3 ERROR 07 ****
254 ;          1716         !ERRDF(5308,MSG2,0);    !**** OPTION 5, RAND3 ERROR 08 ****
255 ;          1717         !ERRHRD(5309,MSG4,0);   !**** OPTION 5, RAND3 ERROR 09 ****
256 ;          1718         !ERRSOFT(5310,MSG3,0);  !**** OPTION 5, RAND3 ERROR 10 ****
257 ;          1719         !ERRSOFT(5311,MSG3,0);  !**** OPTION 5, RAND3 ERROR 11 ****
258 ;          1720         !'RAND4' ROUTINE:
259 ;          1721         !WRITE COMMAND:
260 ;          1722         !ERRDF(5401,MSG1,0);    !**** OPTION 5, RAND4 ERROR 01 ****
261 ;          1723         !ERRDF(5402,MSG1,0);    !**** OPTION 5, RAND4 ERROR 02 ****
262 ;          1724         !ERRDF(5403,MSG1,0);    !**** OPTION 5, RAND4 ERROR 03 ****
263 ;          1725         !CHECK OR READ:
264 ;          1726         !ERRDF(5404,MSG1,0);    !**** OPTION 5, RAND4 ERROR 04 ****
265 ;          1727         !ERRDF(5405,MSG1,0);    !**** OPTION 5, RAND4 ERROR 05 ****
266 ;          1728         !ERRDF(5406,MSG1,0);    !**** OPTION 5, RAND4 ERROR 06 ****
267 ;          1729         !ERRDF(5407,MSG1,0);    !**** OPTION 5, RAND4 ERROR 07 ****
268 ;          1730         !ERRDF(5408,MSG2,0);    !**** OPTION 5, RAND4 ERROR 08 ****
269 ;          1731         !ERRHRD(5409,MSG4,0);   !**** OPTION 5, RAND4 ERROR 09 ****
270 ;          1732         !ERRSOFT(5410,MSG3,0);  !**** OPTION 5, RAND4 ERROR 10 ****
271 ;          1733         !ERRSOFT(5411,MSG3,0);  !**** OPTION 5, RAND4 ERROR 11 ****
272 ;          1734         !
273 ;          1735         ! CODES FOR WHY A UNIT WAS DROPPED:
274 ;          1736         !
275 ;          1737         CODE_1 = 1,                          !DRIVE NOT POWERED UP
276 ;          1738         CODE_2 = 2,                          !DRIVE NOT AN ML11 UNIT
277 ;          1739         CODE_3 = 3,                          !OPERATOR SET TEST LIMITS INCORRECTLY
278 ;          1740         CODE_4 = 4,                          !FAILED ALL RETRIES FOR A NON-FATAL ERROR
279 ;          1741         CODE_5 = 5,                          !CONTROLLER FATAL ERROR
280 ;          1742         CODE_6 = 6,                          !DRIVE FATAL ERROR
281 ;          1743         CODE_7 = 7,                          !ECC HARD ERROR
282 ;          1744         CODE_8 = 8;                          !ECC LOGIC FAILED TO DETECT ERROR
283 ;          1745
284 ;          1746 !
285 ;          1747 ! FIELD DECLARTIONS
286 ;          1748 !
287 ;          1749
288 ;          1750 field
289 ;          1751 !
290 ;          1752 ! VER CZMLBB ADD FIELD DECLARATION
291 ;          1753 !
292 ;          1754 ! FIELD DECLARATION TO MAP THE SINGLE BIT
293 ;          1755 ! ERROR LOG TABLE.
294 ;          1756 !
295 ;          1757     SBE_TBL_MAP =
296 ;          1758         set
297 ;          1759         BNK$_SBE = [0, 0, 2, 0],             !LOG BANK OF SBE
298 ;          1760         BRD$_SBE = [0, 2, 4, 0],             !LOG BOARD OF SBE
```

```
;MLX4
;                    VARIABLES AND CONSTANTS

                1761            BIT$_SBE = [0, 6, 7, 0],                !LOG BIT OF SBE
                1762            UNIT$_SBE = [0, 13, 3, 0],              !LOG UNIT OF SBE
                1763            SUM$_SBE = [1, 0, 16, 0],               !LOG NUMBER OF TIMES OCCURING
                1764            WRD$_0 = [0, 0, 16, 0],                 !ACCESS FIRST WORD OF TABLE
                1765            WRD$_1 = [1, 0, 16, 0]                  !ACCESS SECOND WORD OF TABE
                1766            tes,
                1767    !
                1768    ! VER CZMLBB ADDED FIELD DECLARATION
                1769    !
                1770    ! FIELD DECLARATION FOR PROM MAINTENANCE
                1771    ! SUM TABLE
                1772    !
                1773        PM_SBE_MAP =
                1774            set
                1775            PM_SBE$_SUM = [0, 16, 0]                !ACCESS NUMBER OF SBE PER ARRAY
                1776            tes,
                1777    !
                1778    ! FIELD DECLARATION FOR PATTERN TABLE
                1779    !
                1780        PATMAP =
                1781            set
                1782            COUNT1 = [0, 0, 16, 0],
                1783            COUNT2 = [1, 0, 16, 0]
                1784            tes;
                1785
                1786    own
                1787        !
                1788        ! Single bit error logging and Prom Maint
                1789        ! program storage structures
                1790        !
                1791        ! VER CZMLBB ADDED FOLLOWING THREE DATA DECLARATIONS
                1792        !
                1793        SBE_LOG : blockvector [128, 2, word] field (SBE_TBL_MAP),   !SBE LOCATION TABLE
                1794        PM_SBE_CNT : blockvector [8, 16, word] field (PM_SBE_MAP),  !ARRAY SBE COUNT TALBE
                1795        SBE$_COUNT : word,                          !COUNTS NUMBER OF SBE LOCATION TABLE ENTRIES
                1796        BIT_NUM,                                    !STORS FAILING SBE CHIP NUMBER
                1797        !
                1798        ! ML-11 EXERCISER STORAGE ALLOCATION
                1799        !
                1800        WBUFF : vector [BUFSIZ] volatile,           !IMPORTANT -- WBUFF AND RBUFF MUST
                1801        RBUFF : vector [BUFSIZ] volatile,           !BE CONTIGUOUS AND IN THAT ORDER!!
                1802        WPTR : volatile,
                1803        RPTR : volatile,
                1804        QUICK : volatile,
                1805        PATTERN : volatile,
                1806        DATA_COUNT : volatile,
                1807        COMP_COUNT : volatile,
                1808        EOP_COUNT : volatile,
                1809        BASE_ADDR,
                1810        VEC,
                1811        BR_LEVEL,
                1812        SOFTS : blockvector [8, 16],                !COUNTS FOR 8 LUNS, 16 ARRAYS EACH
```

```
;MLX4
;                    VARIABLES AND CONSTANTS                    27-Mar-1982 19:24:42    TOPS-20 Bliss-16 V2(212)
;                                                               27-Mar-1982 19:23:44    PA:<NEALE>MLX4.BLI.5 (2)

;          1813      HARDS : blockvector [8, 16],               !COUNTS FOR 8 LUNS, 16 ARRAYS EACH
;          1814      TRIES : blockvector [8, 16],               !COUNTS FOR 8 LUNS, 16 ARRAYS EACH
;          1815      WR_COUNT,                                  !# BYTES TRANSFERED VIA 'WRITE'
;          1816      WR_THOUSANDS,                              !THOUSANDS OF BYTES
;          1817      WR_MILLIONS,                               !MILLIONS OF BYTES
;          1818      RD_COUNT,                                  !# BYTES TRANSFERED VIA 'READ'
;          1819      RD_THOUSANDS,
;          1820      RD_MILLIONS,
;          1821      WC_COUNT,                                  !# BYTES TRANSFERED VIA 'WRITE CHECK'
;          1822      WC_THOUSANDS,
;          1823      WC_MILLIONS,
;          1824      I_AM_DONE,
;          1825      RETRYING,
;          1826      BOARD,
;          1827      BANK;
;          1828
;          1829  global
;          1830      ML_REG : vector [NUM_REGS] volatile,
;          1831      PTABLE_ADDR : vector [8] volatile,
;          1832      DRIVE_STATUS : bitvector [8] volatile,
;          1833      DROPT_DRIVES : bitvector [8] volatile,
;          1834      WHY_DROPT : vector [8, byte],
;          1835      NUM_DRIVES,
;          1836      LOW_SECT : vector [8] volatile,
;          1837      TOP_SECT : vector [8] volatile;
;          1838
;          1839  EQUALS;
;          1840
;          1841  macro
;          1842      !
;          1843      ! TO CALCULATE THE TEST RANGES FOR A PARTICULAR LOGICAL UNIT:
;          1844      !
;        M 1845      LOWEST =
;          1846  .LOW_SECT[.LUN]%,                              !THE FIRST SECTOR TO TEST
;        M 1847      HIGHEST =
;          1848  .TOP_SECT[.LUN]%,                              !THE LAST SECTOR TO TEST
;          1849  !
;          1850      ! ADDRESS IN MAIN MEMORY THAT CONTAINS THE PHYSICAL DRIVE
;          1851      ! NUMBER THAT CORRESPONDS TO A PARTICULAR LOGICAL UNIT:
;          1852      !
;        M 1853      DRIVE =
;          1854  (.PTABLE_ADDR[.LUN] + 6)%,
;          1855      !
;          1856      !  ML-11 REGISTER NAMES:
;          1857      !
;        M 1858      MLCS1 =
;          1859  .ML_REG[0]%,                                  !CONTROL AND STATUS REGISTER 1
;        M 1860      MLWC =
;          1861  .ML_REG[1]%,                                  !WORD COUNT REGISTER
;        M 1862      MLBA =
;          1863  .ML_REG[2]%,                                  !UNIBUS ADDRESS REGISTER
;        M 1864      MLDA =
```

```
;MLX4
;                    VARIABLES AND CONSTANTS

          1865    .ML_REG[3]%,                    !DESIRED ADDRESS REGISTER
        M 1866      MLCS2 =
          1867    .ML_REG[4]%,                    !CONTROL AND STATUS REGISTER 2
        M 1868      MLDS =
          1869    .ML_REG[5]%,                    !DRIVE STATUS REGISTER
        M 1870      MLER =
          1871    .ML_REG[6]%,                    !ERROR REGISTER
        M 1872      MLAS =
          1873    .ML_REG[7]%,                    !ATTENTION SUMMARY REGISTER
        M 1874      MLPA =
          1875    .ML_REG[8]%,                    !PROM ADDRESS REGISTER
        M 1876      MLDB =
          1877    .ML_REG[9]%,                    !DATA BUFFER REGISTER
        M 1878      MLMR =
          1879    .ML_REG[10]%,                   !MAINTENANCE REGISTER
        M 1880      MLDT =
          1881    .ML_REG[11]%,                   !DRIVE TYPE REGISTER
        M 1882      MLSN =
          1883    .ML_REG[12]%,                   !SERIAL NUMBER REGISTER
        M 1884      MLE1 =
          1885    .ML_REG[13]%,                   !ECC REGISTER 1
        M 1886      MLE2 =
          1887    .ML_REG[14]%,                   !ECC REGISTER 2
        M 1888      MLD1 =
          1889    .ML_REG[15]%,                   !DATA DIAGNOSTIC REGISTER 1
        M 1890      MLD2 =
          1891    .ML_REG[16]%,                   !DATA DIAGNOSTIC REGISTER 2
        M 1892      MLEE =
          1893    .ML_REG[17]%,                   !ECC ERROR REGISTER
        M 1894      MLEL =
          1895    .ML_REG[18]%,                   !ECC ERROR LOCATION REGISTER
        M 1896      MLPD =
          1897    .ML_REG[19]%,                   !PROM DATA REGISTER
        M 1898      MLBAE =
          1899    .ML_REG[20]%,                   !BUS ADDRESS EXTENSION REGISTER
        M 1900      MLCS3 =
          1901    .ML_REG[21]%,                   !CONTROL AND STATUS REGISTER 3
          1902    !
          1903        !  BIT ASSIGNMENTS:
          1904        !
          1905        !
          1906        !  MLCS1 BITS:
          1907        !
        M 1908      SC =
          1909    (MLCS1)<15,1>%,                 !SPECIAL CONDITION
        M 1910      TRE =
          1911    (MLCS1)<14,1>%,                 !TRANSFER ERROR
        M 1912      MCPE =
          1913    (MLCS1)<13,1>%,                 !MASSBUS CONTROL BUS PARITY ERROR
        M 1914      DVA =
          1915    (MLCS1)<11,1>%,                 !DRIVE AVAILABLE
        M 1916      RDY =
```

```
;MLX4
;                    VARIABLES AND CONSTANTS                    27-Mar-1982 19:24:42    TOPS-20 Bliss-16 V2(212)
;                                                               27-Mar-1982 19:23:44    PA:<NEALE>MLX4.BLI.5 (2)

;          1917   (MLCS1)<7,1>%,                                !READY
;        M 1918      IE =
;          1919   (MLCS1)<6,1>%,                                !INTERRUPT ENABLE
;        M 1920      FUNC =
;          1921   (MLCS1)<0,6>%,                                !FUNCTION (COMMAND) CODE AND GO BIT
;          1922   !
;          1923       !  MLCS2 BITS:
;          1924       !
;        M 1925      DLT =
;          1926   (MLCS2)<15,1>%,                               !DATA LATE
;        M 1927      WCE =
;          1928   (MLCS2)<14,1>%,                               !WRITE CHECK ERROR
;        M 1929      PE =
;          1930   (MLCS2)<13,1>%,                               !PARITY ERROR
;        M 1931      NED =
;          1932   (MLCS2)<12,1>%,                               !NON-EXISTENT DRIVE
;        M 1933      NEM =
;          1934   (MLCS2)<11,1>%,                               !NON-EXISTENT MEMORY
;        M 1935      PGE =
;          1936   (MLCS2)<10,1>%,                               !PROGRAM ERROR
;        M 1937      MXF =
;          1938   (MLCS2)<9,1>%,                                !MISSED TRANSFER
;        M 1939      MDPE =
;          1940   (MLCS2)<8,1>%,                                !MASSBUS DATA BUS PARITY ERROR
;        M 1941      ORDY =
;          1942   (MLCS2)<7,1>%,                                !OUTPUT READY
;        M 1943      IRDY =
;          1944   (MLCS2)<6,1>%,                                !INPUT READY
;        M 1945      CLR =
;          1946   (MLCS2)<5,1>%,                                !CONTROLLER CLEAR
;        M 1947      PAT =
;          1948   (MLCS2)<4,1>%,                                !PARITY TEST
;        M 1949      BAI =
;          1950   (MLCS2)<3,1>%,                                !UNIBUS ADDRESS INCREMENT INHIBIT
;        M 1951      UNIT =
;          1952   (MLCS2)<0,3>%,                                !UNIT SELECT
;          1953   !
;          1954       !  MLDS BITS:
;          1955       !
;        M 1956      ATA =
;          1957   (MLDS)<15,1>%,                                !ATTENTION ACTIVE
;        M 1958      ERR =
;          1959   (MLDS)<14,1>%,                                !ERROR SUMMARY
;        M 1960      MOL =
;          1961   (MLDS)<12,1>%,                                !MEDIUM ON LINE
;        M 1962      LBT =
;          1963   (MLDS)<10,1>%,                                !LAST BLOCK TRANSFERRED
;        M 1964      DPR =
;          1965   (MLDS)<8,1>%,                                 !DRIVE PRESENT
;        M 1966      DRY =
;          1967   (MLDS)<7,1>%,                                 !DRIVE READY
;        M 1968      VV =
```

```
;MLX4
;                    VARIABLES AND CONSTANTS

                1969    (MLDS)<6,1>%,                   !VOLUME VALID
                1970    !
                1971        !  MLER BITS:
                1972        !
              M 1973        DCK =
                1974    (MLER)<15,1>%,                  !DATA CHECK
              M 1975        UNS =
                1976    (MLER)<14,1>%,                  !DRIVE UNSAFE
              M 1977        OPI =
                1978    (MLER)<13,1>%,                  !OPERATION INCOMPLETE
              M 1979        IAE =
                1980    (MLER)<10,1>%,                  !INVALID ADDRESS ERROR
              M 1981        AOE =
                1982    (MLER)<9,1>%,                   !ADDRESS OVERFLOW ERROR
              M 1983        ECH =
                1984    (MLER)<6,1>%,                   !ECC HARD ERROR
              M 1985        DPAR =
                1986    (MLER)<5,1>%,                   !DATA PARITY ERROR
              M 1987        CPAR =
                1988    (MLER)<3,1>%,                   !CONTROL PARITY ERROR
              M 1989        RMR =
                1990    (MLER)<2,1>%,                   !REGISTER MODIFICATION REFUSED
              M 1991        ILR =
                1992    (MLER)<1,1>%,                   !ILLEGAL REGISTER
              M 1993        ILF =
                1994    (MLER)<0,1>%,                   !ILLEGAL FUNCTION
                1995    !
                1996        !  MLMR BITS:
                1997        !
              M 1998        SZ =
                1999    (MLMR)<11,5>%,                  !SYSTEM SIZE (# ARRAY CARDS)
              M 2000        ARR_TYP =
                2001    (MLMR)<10,1>%,                  !ARRAY TYPE (0=16K;1=64K CHIPS)
              M 2002        TRT =
                2003    (MLMR)<8,2>%,                   !TRANSFER RATE
              M 2004        REF_MAR =
                2005    (MLMR)<7,1>%,                   !REFRESH MARGIN
              M 2006        PROM_RW =
                2007    (MLMR)<6,1>%,                   !PROM READ/WRITE
              M 2008        PROM_DIS =
                2009    (MLMR)<5,1>%,                   !PROM DISABLE
              M 2010        DAT_CLK =
                2011    (MLMR)<4,1>%,                   !DATA CLOCK
              M 2012        DAT_DM =
                2013    (MLMR)<3,1>%,                   !DATA DIAGNOSTIC MODE
              M 2014        DCK_EN =
                2015    (MLMR)<2,1>%,                   !DATA CHECK ENABLE
              M 2016        ECC_DIS =
                2017    (MLMR)<1,1>%,                   !ECC DISABLE
              M 2018        ECC_DM =
                2019    (MLMR)<0,1>%,                   !ECC DIAGNOSTIC MODE
                2020    !
```

```
;MLX4
;                    VARIABLES AND CONSTANTS                    27-Mar-1982 19:24:42    TOPS-20 Bliss-16 V2(212)
;                                                               27-Mar-1982 19:23:44    PA:<NEALE>MLX4.BLI.5 (2)

;          2021          !  MLSN BITS:
;          2022          !
;        M 2023          SN3 =
;          2024       (MLSN)<12,4>%,                          !HIGH ORDER DECADE
;        M 2025          SN2 =
;          2026       (MLSN)<8,4>%,                           !THIRD DECADE
;        M 2027          SN1 =
;          2028       (MLSN)<4,4>%,                           !SECOND DECADE
;        M 2029          SN0 =
;          2030       (MLSN)<0,4>%,                           !LOW ORDER DECADE
;          2031    !
;          2032          !  MLEE BITS:
;          2033          !
;        M 2034          UNC =
;          2035       (MLEE)<15,1>%,                          !UNCORRECTABLE ERROR
;        M 2036          SGL =
;          2037       (MLEE)<14,1>%,                          !SINGLE ERROR
;        M 2038          CRC =
;          2039       (MLEE)<13,1>%,                          !CRC ERROR
;        M 2040          CHAN =
;          2041       (MLEE)<6,6>%,                           !CHANNEL IN ERROR
;        M 2042          EFUN =
;          2043       (MLEE)<0,6>%;                           !ERROR FUNCTION
;          2044
;          2045    external
;          2046          !
;          2047          !HEADER INFORMATION:
;          2048          !
;          2049          EFN$21,                                 ! VER CZMLBB EVENT FLAG TO TURN ON SBE TESTING
;          2050          L$UNIT,
;          2051          L$LUN,
;          2052          !
;          2053          !ALL OF THE SOFTWARE P-TABLE LOCATIONS:
;          2054          !
;          2055          LIMIT,
;          2056          RANGE,
;          2057          LSECT,
;          2058          TSECT,
;          2059          ONLY,
;          2060          DROPNE,
;          2061          DROP1,
;          2062          DROP2,
;          2063          DROP3,
;          2064          DROP4,
;          2065          DROP5,
;          2066          MARPAT,
;          2067          REFRESH,
;          2068          ECCDIS,
;          2069          EOPSUM,
;          2070          ERROUT,
;          2071          !
;          2072          !RANDOM NUMBER GENERATION VALUES:
```

```
;MLX4
;                    VARIABLES AND CONSTANTS          27-Mar-1982 19:24:42   TOPS-20 Bliss-16 V2(212)
;                                                     27-Mar-1982 19:23:44   PA:<NEALE>MLX4.BLI.5 (2)

            2073      !
            2074      SEED1,
            2075      SEED2,
            2076      SEED3,
            2077      RANDOM;
            2078
            2079  external routine
            2080      RN : novalue;                   !FOR 16-BIT RANDOM NUMBER GENERATION
            2081
            2082  !+
            2083  !
            2084  !   THE PATTERN TABLE:
            2085  !
            2086  !
            2087  !                       R E G U L A R   P A T T E R N S
            2088  !
            2089  !   PATTERN
            2090  !   NUMBER                  THE TWO ASSOCIATED COUNTS              VALUE OF DATA   VALUE OF COMP
            2091  !
            2092  !      1        0 NIBBLES OF DATA, 1024 NIBBLES OF COMP            0101            1010
            2093  !      2        1 NIBBLE  OF DATA,    1 NIBBLE  OF COMP            0101            1010
            2094  !      3        4 NIBBLES OF DATA,    4 NIBBLES OF COMP            0101            1010
            2095  !      4        9 NIBBLES OF DATA,    9 NIBBLES OF COMP            0101            1010
            2096  !      5     1024 NIBBLES OF DATA, 1024 NIBBLES OF COMP            0101            1010
            2097  !
            2098  !      6        0 NIBBLES OF DATA, 1024 NIBBLES OF COMP            0000            1111
            2099  !      7        1 NIBBLE  OF DATA,    1 NIBBLE  OF COMP            0000            1111
            2100  !      8        4 NIBBLES OF DATA,    4 NIBBLES OF COMP            0000            1111
            2101  !      9        9 NIBBLES OF DATA,    9 NIBBLES OF COMP            0000            1111
            2102  !     10     1024 NIBBLES OF DATA, 1024 NIBBLES OF COMP            0000            1111
            2103  !
            2104  !
            2105  !
            2106  !                       C O M P L E M E N T   P A T T E R N S
            2107  !
            2108  !   PATTERN
            2109  !   NUMBER                  THE TWO ASSOCIATED COUNTS              VALUE OF DATA   VALUE OF COMP
            2110  !
            2111  !     - 1        0 NIBBLES OF DATA, 1024 NIBBLES OF COMP            1010            0101
            2112  !     - 2        1 NIBBLE  OF DATA,    1 NIBBLE  OF COMP            1010            0101
            2113  !     - 3        4 NIBBLES OF DATA,    4 NIBBLES OF COMP            1010            0101
            2114  !     - 4        9 NIBBLES OF DATA,    9 NIBBLES OF COMP            1010            0101
            2115  !     - 5     1024 NIBBLES OF DATA, 1024 NIBBLES OF COMP            1010            0101
            2116  !
            2117  !     - 6        0 NIBBLES OF DATA, 1024 NIBBLES OF COMP            1111            0000
            2118  !     - 7        1 NIBBLE  OF DATA,    1 NIBBLE  OF COMP            1111            0000
            2119  !     - 8        4 NIBBLES OF DATA,    4 NIBBLES OF COMP            1111            0000
            2120  !     - 9        9 NIBBLES OF DATA,    9 NIBBLES OF COMP            1111            0000
            2121  !    -10     1024 NIBBLES OF DATA, 1024 NIBBLES OF COMP            1111            0000
            2122  !-
            2123
            2124  global
```

```
;MLX4
;                    VARIABLES AND CONSTANTS

;          2125  !<BLF/NOFORMAT>
;          2126     PATTBL: blockvector [NUM_PATS/2,2] field(PATMAP)
;          2127        preset (
;          2128           [0,COUNT1]      = %decimal'0'    ,      !FOR PATTERNS 1, -1, 6, -6
;          2129           [0,COUNT2]      = %decimal'1024',
;          2130           [1,COUNT1]      = %decimal'1'    ,      !FOR PATTERNS 2, -2, 7, -7
;          2131           [1,COUNT2]      = %decimal'1'    ,
;          2132           [2,COUNT1]      = %decimal'4'    ,      !FOR PATTERNS 3, -3, 8, -8
;          2133           [2,COUNT2]      = %decimal'4'    ,
;          2134           [3,COUNT1]      = %decimal'9'    ,      !FOR PATTERNS 4, -4, 9, -9
;          2135           [3,COUNT2]      = %decimal'9'    ,
;          2136           [4,COUNT1]      = %decimal'1024',      !FOR PATTERNS 5, -5, 10, -10
;          2137           [4,COUNT2]      = %decimal'1024'
;          2138                 );
;          2139  !<BLF/FORMAT>
;          2140
;          2141  bind
;          2142      !
;          2143      !  DEFINITIONS OF LOCATIONS WITHIN THE WRITE AND READ BUFFERS:
;          2144      !
;          2145      WDBUFF = WBUFF,                              !256-WORD WRITE DATA BUFFER
;          2146      WCBUFF = WBUFF + 512,                        !256-WORD WRITE COMP BUFFER
;          2147      RDBUFF = RBUFF,                              !256-WORD READ DATA BUFFER
;          2148      RCBUFF = RBUFF + 512,                        !256-WORD READ COMP BUFFER
;          2149      END_WBUFF = (WBUFF + BUFSIZ*2),              !JUST BEYOND END OF FULL WRITE BUFFER
;          2150      END_RBUFF = (RBUFF + BUFSIZ*2),              !JUST BEYOND END OF FULL READ  BUFFER
```

```
;MLX4
;                MESSAGES AND PRINT FORMATS                    27-Mar-1982 19:24:42    TOPS-20 Bliss-16 V2(212)
;                                                              27-Mar-1982 19:23:44    PA:<NEALE>MLX4.BLI.5 (3)

;          2151  %sbttl 'MESSAGES AND PRINT FORMATS'
;          2152
;          2153      ! SINGLE BIT ERROR LOGGING TABLE MESSAGE STRINGS
;          2154      !
;          2155      PHR21 = uplit (%asciz'                SINGLE BIT ERROR LOG SUMMARY REPORT'),
;          2156      FMT16 = uplit (%asciz'%N%D1%A(D)      %D2%A(D)      %D1%A(D)      %D2%A(D)      %D6%A(D)'),
;          2157      SBE$HEADER = uplit (%asciz'UNIT #:   ARRAY #:   BANK #:   BIT #:      COUNT #:'),
;          2158      !
;          2159      !  SPECIAL PURPOSE FORMATS (WITH SAMPLE PRINTOUTS):
;          2160      !
;          2161      FMT1A = uplit (%asciz'%N%T%S2%D2'),
;          2162      !'PATTERN NUMBER  XX'
;          2163      FMT1B = uplit (%asciz'%S7%T%S%A-%D2'),
;          2164      !'       PATTERN NUMBER -XX'
;          2165      FMT2 = uplit (%asciz'%S2%T'),
;          2166      !'  (NOT POWERED UP)'
;          2167      !'  (NOT AN ML11 UNIT)'
;          2168      !'  (OPERATOR SELECTED TEST LIMITS INCORRECTLY)'
;          2169      !'  (ALL RETRIES FAILED FOR A NON-FATAL ERROR)'
;          2170      !'  (CONTROLLER FATAL ERROR)'
;          2171      !'  (DRIVE FATAL ERROR)'
;          2172      !'  (ECC HARD ERROR)'
;          2173      !'  UP'
;          2174      !'  DOWN'
;          2175      !'  --> RUN ML11 PROM MAINTENANCE PROGRAM'
;          2176      FMT3 = uplit (%asciz'%N%A**%S%T%S%T%D3%S%A**'),
;          2177      !'** END PASS  X **'
;          2178      FMT4A = uplit (%asciz'%N2%T%A:%S%D1%S4%T%A:%S%D1'),
;          2179      !'LOGICAL UNIT: X    DRIVE: Y'
;          2180      FMT4B = uplit (%asciz'%S4%T%A:%S%O6'),
;          2181      !'    SERIAL #: ZZZZZZ'
;          2182      !'    CSR ADDRESS: XXXXXX'
;          2183      FMT4C = uplit (%asciz'%S4%T%A:%S%D1%D1%D1%D1'),
;          2184      !'    SERIAL #: DDDD'
;          2185      FMT5 = uplit (%asciz'%N%T%S3%ASECTORS UNDER TEST:%S%O6%S%ATO%S%O6'),
;          2186      !'ML11-X   SECTORS UNDER TEST: XXXXXX TO YYYYYY'
;          2187      FMT6 = uplit (%asciz'%N%ABEGAN %D4%A WORD%S%T%A AT%S%T%S%O6'),
;          2188      !'BEGAN YYYY WORD WRITE AT SECTOR ZZZZZZ'
;          2189      !'BEGAN YYYY WORD READ AT SECTOR ZZZZZZ'
;          2190      !'BEGAN YYYY WORD WRITE CHECK AT SECTOR ZZZZZZ'
;          2191      FMT7 = uplit (%asciz'%N%S2%T%A:%S%D5'),
;          2192      !'  SOFT ERROR COUNT: DDDDD'
;          2193      !'  HARD ERROR COUNT: DDDDD'
;          2194      !'  TRANSFER RETRIES: DDDDD'
;          2195      FMT8 = uplit (%asciz'%N%ATRANSFER RATE: %T%A MBYTES/SECOND'),
;          2196      !'TRANSFER RATE: X MBYTES/SECOND'
;          2197      FMT9 = uplit (%asciz'%N%S4%AARRAY%D3%A:%S%D5'),
;          2198      !'    ARRAY XX: YYYYY'
;          2199      FMT10A = uplit (%asciz'%N%T%A:%S2%T%S%O6%S2%ABOARD%D3%S2%ABANK%D2'),
;          2200      !'FAILED:  SECTOR XXXXXX  BOARD YY  BANK Z
;          2201      FMT10B = uplit (%asciz'%S2%ABIT%D3'),
;          2202      !'  BIT QQ'
```

```
;          2203      FMT11 = uplit (%asciz'%N%T%S%O6%A EXCEEDS %T%S%O6'),
;          2204      !'TOP SECTOR OF XXXXXX EXCEEDS SYSTEM LIMIT OF YYYYYY'
;          2205      !'LOW SECTOR OF XXXXXX EXCEEDS TOP SECTOR OF YYYYYY'
;          2206      FMT12A = uplit (%asciz'%N%AGOOD DATA: %O6%A AT LOCATION %O6'),
;          2207      !'GOOD DATA: XXXXXX AT LOCATION YYYYYY'
;          2208      FMT12B = uplit (%asciz'%N%ABAD DATA:  %O6%A AT LOCATION %O6'),
;          2209      !'BAD DATA:  XXXXXX AT LOCATION YYYYYY'
;          2210      FMT13 = uplit (%asciz'%N%AARRAY%D3%S2%T'),
;          2211      !'ARRAY XX  --> RUN ML11 PROM MAINTENANCE PROGRAM'
;          2212      FMT14 = uplit (%asciz'%N%D5%S%T'),
;          2213      !'XXXXX MBYTES WRITTEN'
;          2214      !'XXXXX MBYTES READ'
;          2215      !'XXXXX MBYTES WRITE CHECKED'
;          2216      FMT15 = uplit (%asciz'%S2%T'),
;          2217      !'  ECH'
;          2218      !'  NED'
;          2219      !(ANY OF THE ERROR BITS)
;          2220      CRLF = uplit (%asciz'%N'),
;          2221      !
;          2222      !  MESSAGE MAPS:
;          2223      !
;          2224      SAY1 = uplit (%asciz'%N%T'),
;          2225      SAY2 = uplit (%asciz'%N%T%S%T'),
;          2226      SAY3 = uplit (%asciz'%N%T%S%T%S%T'),
;          2227      SAY4 = uplit (%asciz'%N%T%S%T%S%T%S%T'),
;          2228      SAY5 = uplit (%asciz'%N%T%S%T%S%T%S%T%S%T'),
;          2229      !
;          2230      !  WORDS:
;          2231      !
;          2232      WRD2 = uplit (%asciz'BEGIN'),
;          2233      WRD3 = uplit (%asciz'END'),
;          2234      WRD4 = uplit (%asciz'PASS'),
;          2235      WRD6 = uplit (%asciz'ML11-A'),
;          2236      WRD7 = uplit (%asciz'ML11-B'),
;          2237      WRD11 = uplit (%asciz'DRIVE'),
;          2238      WRD15 = uplit (%asciz'SECTOR'),
;          2239      WRD16 = uplit (%asciz'WRITE'),
;          2240      WRD17 = uplit (%asciz'READ'),
;          2241      WRD18 = uplit (%asciz'RETRY'),
;          2242      WRD19 = uplit (%asciz'SUCCEEDED'),
;          2243      WRD20 = uplit (%asciz'FAILED'),
;          2244      WRD21 = uplit (%asciz'DROPPED'),
;          2245      WRD24 = uplit (%asciz'UP'),
;          2246      WRD25 = uplit (%asciz'DOWN'),
;          2247      WRD34 = uplit (%asciz'RUNNING'),
;          2248      WRD35 = uplit (%asciz'MARCHING:  '),
;          2249      WRD36 = uplit (%asciz'ENABLED'),
;          2250      WRD37 = uplit (%asciz'DISABLED'),
;          2251      WRD38 = uplit (%asciz'ECC'),
;          2252      WRD40 = uplit (%asciz'WITH'),
;          2253      WRD41 = uplit (%asciz'AND'),
;          2254      !
```

```
834 ;MLX4
835 ;
836
837 ;          2255      ! ML-11 BITS:
838 ;          2256      !
839 ;          2257      MLB2 = uplit (%asciz'NED'),
840 ;          2258      MLB3 = uplit (%asciz'NEM'),
841 ;          2259      MLB4 = uplit (%asciz'PGE'),
842 ;          2260      MLB5 = uplit (%asciz'DLT'),
843 ;          2261      MLB6 = uplit (%asciz'WCE'),
844 ;          2262      MLB7 = uplit (%asciz'PE'),
845 ;          2263      MLB8 = uplit (%asciz'MXF'),
846 ;          2264      MLB9 = uplit (%asciz'MDPE'),
847 ;          2265      MLB10 = uplit (%asciz'MCPE'),
848 ;          2266      MLB11 = uplit (%asciz'UNS'),
849 ;          2267      MLB12 = uplit (%asciz'IAE'),
850 ;          2268      MLB13 = uplit (%asciz'AOE'),
851 ;          2269      MLB14 = uplit (%asciz'RMR'),
852 ;          2270      MLB15 = uplit (%asciz'ILR'),
853 ;          2271      MLB16 = uplit (%asciz'ILF'),
854 ;          2272      MLB17 = uplit (%asciz'OPI'),
855 ;          2273      MLB18 = uplit (%asciz'DPAR'),
856 ;          2274      MLB19 = uplit (%asciz'CPAR'),
857 ;          2275      MLB20 = uplit (%asciz'DCK'),
858 ;          2276      MLB21 = uplit (%asciz'ECH'),
859 ;          2277      MLB22 = uplit (%asciz'CRC'),
860 ;          2278      MLB23 = uplit (%asciz'SGL'),
861 ;          2279      MLB24 = uplit (%asciz'UNC'),
862 ;          2280      !
863 ;          2281      ! ROUTINE NAMES:
864 ;          2282      !
865 ;          2283      RTN0 = uplit (%asciz'COMMAND INTEGRITY ROUTINE'),
866 ;          2284      RTN1 = uplit (%asciz'OPT1'),
867 ;          2285      RTN2 = uplit (%asciz'OPT2'),
868 ;          2286      RTN3 = uplit (%asciz'OPT3'),
869 ;          2287      RTN4 = uplit (%asciz'OPT4'),
870 ;          2288      RTN5 = uplit (%asciz'OPT5'),
871 ;          2289      RTN5A = uplit (%asciz'RAND1'),          !RANDOM DATA
872 ;          2290      RTN5B = uplit (%asciz'RAND2'),          !DATA & WORD COUNTS
873 ;          2291      RTN5C = uplit (%asciz'RAND3'),          !DATA, WORD COUNTS & SECTORS
874 ;          2292      RTN5D = uplit (%asciz'RAND4'),          !DATA, WORD COUNTS, SECTORS & UNITS
875 ;          2293      !
876 ;          2294      ! PHRASES:
877 ;          2295      !
878 ;          2296      PHR1 = uplit (%asciz'WRITE CHECK'),
879 ;          2297      PHR2 = uplit (%asciz'QUICK VERIFY'),
880 ;          2298      PHR3 = uplit (%asciz'REFRESH MARGINING'),
881 ;          2299      PHR4 = uplit (%asciz'CSR ADDRESS'),
882 ;          2300      PHR5 = uplit (%asciz'SOFT ERROR COUNT'),
883 ;          2301      PHR6 = uplit (%asciz'TRANSFER RETRIES'),
884 ;          2302      PHR7 = uplit (%asciz'LOGICAL UNIT'),
885 ;          2303      PHR8 = uplit (%asciz'SERIAL #'),
886 ;          2304      PHR9 = uplit (%asciz'PATTERN NUMBER'),
887 ;          2305      PHR10 = uplit (%asciz'ERROR BITS SET:'),
888 ;          2306      PHR11 = uplit (%asciz'SC SET BUT NO SYSTEM ERRORS FOUND'),
```

```
890 ;MLX4
891 ;                   MESSAGES AND PRINT FORMATS
892
893 ;          2307         PHR12 = uplit (%asciz'NUMBER OF MBYTES TRANSFERED:'),
894 ;          2308         PHR13 = uplit (%asciz'MBYTES WRITE CHECKED'),
895 ;          2309         PHR14 = uplit (%asciz'TOP SECTOR OF'),
896 ;          2310         PHR15 = uplit (%asciz'LOW SECTOR OF'),
897 ;          2311         PHR16 = uplit (%asciz'SYSTEM LIMIT OF'),
898 ;          2312         PHR17 = uplit (%asciz'PERFORMANCE SUMMARY'),
899 ;          2313         PHR18 = uplit (%asciz'HARD ERROR COUNT'),
900 ;          2314         PHR19 = uplit (%asciz'MBYTES WRITTEN'),
901 ;          2315         PHR20 = uplit (%asciz'MBYTES READ'),
902 ;          2316         !
903 ;          2317         !  TRANSFER RATES:
904 ;          2318         !
905 ;          2319         TRT00 = uplit (%asciz'2'),
906 ;          2320         TRT01 = uplit (%asciz'1'),
907 ;          2321         TRT10 = uplit (%asciz'.5'),
908 ;          2322         TRT11 = uplit (%asciz'.25'),
909 ;          2323         !
910 ;          2324         !  DROP MESSAGES:
911 ;          2325         !
912 ;          2326         CAUSE1 = uplit (%asciz'(NOT POWERED UP)'),
913 ;          2327         CAUSE2 = uplit (%asciz'(NOT AN ML11 UNIT)'),
914 ;          2328         CAUSE3 = uplit (%asciz'(OPERATOR SELECTED TEST LIMITS INCORRECTLY)'),
915 ;          2329         CAUSE4 = uplit (%asciz'(ALL RETRIES FAILED FOR A NON-FATAL ERROR)'),
916 ;          2330         CAUSE5 = uplit (%asciz'(CONTROLLER FATAL ERROR)'),
917 ;          2331         CAUSE6 = uplit (%asciz'(DRIVE FATAL ERROR)'),
918 ;          2332         CAUSE7 = uplit (%asciz'(ECC HARD ERROR)'),
919 ;          2333         CAUSE8 = uplit (%asciz'(ECC LOGIC FAILURE)'),
920 ;          2334         !
921 ;          2335         !  DIAGNOSES:
922 ;          2336         !
923 ;          2337         MSG0 = uplit (%asciz'INTERRUPT DID NOT OCCUR, BUT THE TRANSFER IS COMPLETE'),
924 ;          2338         MSG1 = uplit (%asciz'--> RUN ML11 LOGIC TEST'),
925 ;          2339         MSG2 = uplit (%asciz'--> RUN ML11 PROM MAINTENANCE PROGRAM'),
926 ;          2340         MSG3 = uplit (%asciz'SOFT ERROR'),
927 ;          2341         MSG4 = uplit (%asciz'HARD ERROR'),
928 ;          2342         MSG5 = uplit (%asciz'ECC LOGIC FAILED TO DETECT DATA ERROR');
929 ;          2343
```

```
931 ;MLX4
932 ;                   ML11 INTERRUPT SERVICE ROUTINE                  27-Mar-1982 19:24:42    TOPS-20 Bliss-16 V2(212)
933                                                                     27-Mar-1982 19:23:44    PA:<NEALE>MLX4.BLI.5 (4)
934 ;           2344  %sbttl 'ML11 INTERRUPT SERVICE ROUTINE'
935 ;           2345  BGNSRV (SERVICE);
936 ;           2346  I_AM_DONE = ACTIVE;
937 ;           2347  ~RDSRV;
941
942
943
944 005662  040  040  040  P.AAA:  .ASCII  /   /
945 005665  040  040  011          .ASCII  /   /<11>
946 005670  011  123  111          .ASCII  <11>/SI/
947 005673  116  107  114          .ASCII  /NGL/
948 005676  105  040  102          .ASCII  /E B/
949 005701  111  124  040          .ASCII  /IT /
950 005704  105  122  122          .ASCII  /ERR/
951 005707  117  122  040          .ASCII  /OR /
952 005712  114  117  107          .ASCII  /LOG/
953 005715  040  123  125          .ASCII  / SU/
954 005720  115  115  101          .ASCII  /MMA/
955 005723  122  131  040          .ASCII  /RY /
956 005726  122  105  120          .ASCII  /REP/
957 005731  117  122  124          .ASCII  /ORT/
958 005734  000  000               .ASCII  <00><00>
959 005736  045  116  045  P.AAB:  .ASCII  /%N%/
960 005741  104  061  045          .ASCII  /D1%/
961 005744  101  050  104          .ASCII  /A(D/
962 005747  051  040  040          .ASCII  /)  /
963 005752  040  040  040          .ASCII  /   /
964 005755  040  045  104          .ASCII  / %D/
965 005760  062  045  101          .ASCII  /2%A/
966 005763  050  104  051          .ASCII  /(D)/
967 005766  040  040  040          .ASCII  /   /
968 005771  040  040  040          .ASCII  /   /
969 005774  045  104  061          .ASCII  /%D1/
970 005777  045  101  050          .ASCII  /%A(/
971 006002  104  051  040          .ASCII  /D) /
972 006005  040  040  040          .ASCII  /   /
973 006010  040  040  045          .ASCII  /  %/
974 006013  104  062  045          .ASCII  /D2%/
975 006016  101  050  104          .ASCII  /A(D/
976 006021  051  040  040          .ASCII  /)  /
977 006024  040  040  040          .ASCII  /   /
978 006027  040  040  045          .ASCII  /  %/
979 006032  104  066  045          .ASCII  /D6%/
980 006035  101  050  104          .ASCII  /A(D/
981 006040  051  000               .ASCII  /)/<00>
982 006042  125  116  111  P.AAC:  .ASCII  /UNI/
983 006045  124  040  043          .ASCII  /T #/
984                     ;MLX4
985                     ;                   ML11 INTERRUPT SERVICE ROUTINE  27-Mar-1982 19:24:42    TOPS
986                                                                         27-Mar-1982 19:23:44    PA:<
987 006050  072  040  040          .ASCII  /:  /
988 006053  040  101  122          .ASCII  / AR/
989 006056  122  101  131          .ASCII  /RAY/
990 006061  040  043  072          .ASCII  / #:/
```

```
 991 006064  040  040  040          .ASCII  /   /
 992 006067  102  101  116          .ASCII  /BAN/
 993 006072  113  040  043          .ASCII  /K #/
 994 006075  072  040  040          .ASCII  /:  /
 995 006100  040  102  111          .ASCII  / BI/
 996 006103  124  040  043          .ASCII  /T #/
 997 006106  072  040  040          .ASCII  /:  /
 998 006111  040  040  040          .ASCII  /   /
 999 006114  040  103  117          .ASCII  / CO/
1000 006117  125  116  124          .ASCII  /UNT/
1001 006122  040  043  072          .ASCII  / #:/
1002 006125  000                    .ASCII  <00>
1003 006126  045  116  045  P.AAD:  .ASCII  /%N%/
1004 006131  124  045  123          .ASCII  /T%S/
1005 006134  062  045  104          .ASCII  /2%D/
1006 006137  062  000  000          .ASCII  /2/<00><00>
1007 006142  045  123  067  P.AAE:  .ASCII  /%S7/
1008 006145  045  124  045          .ASCII  /%T%/
1009 006150  123  045  101          .ASCII  /S%A/
1010 006153  055  045  104          .ASCII  /-%D/
1011 006156  062  000               .ASCII  /2/<00>
1012 006160  045  123  062  P.AAF:  .ASCII  /%S2/
1013 006163  045  124  000          .ASCII  /%T/<00>
1014 006166  045  116  045  P.AAG:  .ASCII  /%N%/
1015 006171  101  052  052          .ASCII  /A**/
1016 006174  045  123  045          .ASCII  /%S%/
1017 006177  124  045  123          .ASCII  /T%S/
1018 006202  045  124  045          .ASCII  /%T%/
1019 006205  104  063  045          .ASCII  /D3%/
1020 006210  123  045  101          .ASCII  /S%A/
1021 006213  052  052  000          .ASCII  /**/<00>
1022 006216  045  116  062  P.AAH:  .ASCII  /%N2/
1023 006221  045  124  045          .ASCII  /%T%/
1024 006224  101  072  045          .ASCII  /A:%/
1025 006227  123  045  104          .ASCII  /S%D/
1026 006232  061  045  123          .ASCII  /1%S/
1027 006235  064  045  124          .ASCII  /4%T/
1028 006240  045  101  072          .ASCII  /%A:/
1029 006243  045  123  045          .ASCII  /%S%/
1030 006246  104  061  000          .ASCII  /D1/<00>
1031 006251  000                    .ASCII  <00>
1032 006252  045  123  064  P.AAI:  .ASCII  /%S4/
1033 006255  045  124  045          .ASCII  /%T%/
1034 006260  101  072  045          .ASCII  /A:%/
1035 006263  123  045  117          .ASCII  /S%O/
1036 006266  066  000               .ASCII  /6/<00>
1037 006270  045  123  064  P.AAJ:  .ASCII  /%S4/
1038 006273  045  124  045          .ASCII  /%T%/
```

```
1040                                    ;MLX4
1041                                    ;                ML11 INTERRUPT SERVICE ROUTINE
1042
1043 006276  101  072  045              .ASCII  /A:%/
1044 006301  123  045  104              .ASCII  /S%D/
1045 006304  061  045  104              .ASCII  /1%D/
1046 006307  061  045  104              .ASCII  /1%D/
1047 006312  061  045  104              .ASCII  /1%D/
1048 006315  061  000  000              .ASCII  /1/<00><00>
1049 006320  045  116  045  P.AAK:     .ASCII  /%N%/
1050 006323  124  045  123              .ASCII  /T%S/
1051 006326  063  045  101              .ASCII  /3%A/
1052 006331  123  105  103              .ASCII  /SEC/
1053 006334  124  117  122              .ASCII  /TOR/
1054 006337  123  040  125              .ASCII  /S U/
1055 006342  116  104  105              .ASCII  /NDE/
1056 006345  122  040  124              .ASCII  /R T/
1057 006350  105  123  124              .ASCII  /EST/
1058 006353  072  045  123              .ASCII  /:%S/
1059 006356  045  117  066              .ASCII  /%O6/
1060 006361  045  123  045              .ASCII  /%S%/
1061 006364  101  124  117              .ASCII  /ATO/
1062 006367  045  123  045              .ASCII  /%S%/
1063 006372  117  066  000              .ASCII  /O6/<00>
1064 006375  000                        .ASCII  <00>
1065 006376  045  116  045  P.AAL:     .ASCII  /%N%/
1066 006401  101  102  105              .ASCII  /ABE/
1067 006404  107  101  116              .ASCII  /GAN/
1068 006407  040  045  104              .ASCII  / %D/
1069 006412  064  045  101              .ASCII  /4%A/
1070 006415  040  127  117              .ASCII  / WO/
1071 006420  122  104  045              .ASCII  /RD%/
1072 006423  123  045  124              .ASCII  /S%T/
1073 006426  045  101  040              .ASCII  /%A /
1074 006431  101  124  045              .ASCII  /AT%/
1075 006434  123  045  124              .ASCII  /S%T/
1076 006437  045  123  045              .ASCII  /%S%/
1077 006442  117  066  000              .ASCII  /O6/<00>
1078 006445  000                        .ASCII  <00>
1079 006446  045  116  045  P.AAM:     .ASCII  /%N%/
1080 006451  123  062  045              .ASCII  /S2%/
1081 006454  124  045  101              .ASCII  /T%A/
1082 006457  072  045  123              .ASCII  /:%S/
1083 006462  045  104  065              .ASCII  /%D5/
1084 006465  000                        .ASCII  <00>
1085 006466  045  116  045  P.AAN:     .ASCII  /%N%/
1086 006471  101  124  122              .ASCII  /ATR/
1087 006474  101  116  123              .ASCII  /ANS/
1088 006477  106  105  122              .ASCII  /FER/
1089 006502  040  122  101              .ASCII  / RA/
1090 006505  124  105  072              .ASCII  /TE:/
1091 006510  040  045  124              .ASCII  / %T/
1092 006513  045  101  040              .ASCII  /%A /
1093 006516  115  102  131              .ASCII  /MBY/
1094 006521  124  105  123              .ASCII  /TES/
```

```
                                        ;MLX4
                                        ;                ML11 INTERRUPT SERVICE ROUTINE
1096
1097
1098
1099  006524   057   123   105          .ASCII  <57>/SE/
1100  006527   103   117   116          .ASCII  /CON/
1101  006532   104   000                .ASCII  /D/<00>
1102  006534   045   116   045  P.AAO:  .ASCII  /%N%/
1103  006537   123   064   045          .ASCII  /S4%/
1104  006542   101   101   122          .ASCII  /AAR/
1105  006545   122   101   131          .ASCII  /RAY/
1106  006550   045   104   063          .ASCII  /%D3/
1107  006553   045   101   072          .ASCII  /%A:/
1108  006556   045   123   045          .ASCII  /%S%/
1109  006561   104   065   000          .ASCII  /D5/<00>
1110  006564   045   116   045  P.AAP:  .ASCII  /%N%/
1111  006567   124   045   101          .ASCII  /T%A/
1112  006572   072   045   123          .ASCII  /:%S/
1113  006575   062   045   124          .ASCII  /2%T/
1114  006600   045   123   045          .ASCII  /%S%/
1115  006603   117   066   045          .ASCII  /O6%/
1116  006606   123   062   045          .ASCII  /S2%/
1117  006611   101   102   117          .ASCII  /ABO/
1118  006614   101   122   104          .ASCII  /ARD/
1119  006617   045   104   063          .ASCII  /%D3/
1120  006622   045   123   062          .ASCII  /%S2/
1121  006625   045   101   102          .ASCII  /%AB/
1122  006630   101   116   113          .ASCII  /ANK/
1123  006633   045   104   062          .ASCII  /%D2/
1124  006636   000   000                .ASCII  <00><00>
1125  006640   045   123   062  P.AAQ:  .ASCII  /%S2/
1126  006643   045   101   102          .ASCII  /%AB/
1127  006646   111   124   045          .ASCII  /IT%/
1128  006651   104   063   000          .ASCII  /D3/<00>
1129  006654   045   116   045  P.AAR:  .ASCII  /%N%/
1130  006657   124   045   123          .ASCII  /T%S/
1131  006662   045   117   066          .ASCII  /%O6/
1132  006665   045   101   040          .ASCII  /%A /
1133  006670   105   130   103          .ASCII  /EXC/
1134  006673   105   105   104          .ASCII  /EED/
1135  006676   123   040   045          .ASCII  /S %/
1136  006701   124   045   123          .ASCII  /T%S/
1137  006704   045   117   066          .ASCII  /%O6/
1138  006707   000                      .ASCII  <00>
1139  006710   045   116   045  P.AAS:  .ASCII  /%N%/
1140  006713   101   107   117          .ASCII  /AGO/
1141  006716   117   104   040          .ASCII  /OD /
1142  006721   104   101   124          .ASCII  /DAT/
1143  006724   101   072   040          .ASCII  /A: /
1144  006727   045   117   066          .ASCII  /%O6/
1145  006732   045   101   040          .ASCII  /%A /
1146  006735   101   124   040          .ASCII  /AT /
1147  006740   114   117   103          .ASCII  /LOC/
1148  006743   101   124   111          .ASCII  /ATI/
1149  006746   117   116   040          .ASCII  /ON /
1150  006751   045   117   066          .ASCII  /%O6/
```

```
                        ;MLX4
                        ;                ML11 INTERRUPT SERVICE ROUTINE
```



```
1152
1153
1154
1155 006754  000  000            .ASCII  <00><00>
1156 006756  045  116  045  P.AAT: .ASCII  /%N%/
1157 006761  101  102  101       .ASCII  /ABA/
1158 006764  104  040  104       .ASCII  /D D/
1159 006767  101  124  101       .ASCII  /ATA/
1160 006772  072  040  040       .ASCII  /:  /
1161 006775  045  117  066       .ASCII  /%06/
1162 007000  045  101  040       .ASCII  /%A /
1163 007003  101  124  040       .ASCII  /AT /
1164 007006  114  117  103       .ASCII  /LOC/
1165 007011  101  124  111       .ASCII  /ATI/
1166 007014  117  116  040       .ASCII  /ON /
1167 007017  045  117  066       .ASCII  /%06/
1168 007022  000  000            .ASCII  <00><00>
1169 007024  045  116  045  P.AAU: .ASCII  /%N%/
1170 007027  101  101  122       .ASCII  /AAR/
1171 007032  122  101  131       .ASCII  /RAY/
1172 007035  045  104  063       .ASCII  /%D3/
1173 007040  045  123  062       .ASCII  /%S2/
1174 007043  045  124  000       .ASCII  /%T/<00>
1175 007046  045  116  045  P.AAV: .ASCII  /%N%/
1176 007051  104  065  045       .ASCII  /D5%/
1177 007054  123  045  124       .ASCII  /S%T/
1178 007057  000                 .ASCII  <00>
1179 007060  045  123  062  P.AAW: .ASCII  /%S2/
1180 007063  045  124  000       .ASCII  /%T/<00>
1181 007066  045  116  000  P.AAX: .ASCII  /%N/<00>
1182 007071  000                 .ASCII  <00>
1183 007072  045  116  045  P.AAY: .ASCII  /%N%/
1184 007075  124  000  000       .ASCII  /T/<00><00>
1185 007100  045  116  045  P.AAZ: .ASCII  /%N%/
1186 007103  124  045  123       .ASCII  /T%S/
1187 007106  045  124  000       .ASCII  /%T/<00>
1188 007111  000                 .ASCII  <00>
1189 007112  045  116  045  P.ABA: .ASCII  /%N%/
1190 007115  124  045  123       .ASCII  /T%S/
1191 007120  045  124  045       .ASCII  /%T%/
1192 007123  123  045  124       .ASCII  /S%T/
1193 007126  000  000            .ASCII  <00><00>
1194 007130  045  116  045  P.ABB: .ASCII  /%N%/
1195 007133  124  045  123       .ASCII  /T%S/
1196 007136  045  124  045       .ASCII  /%T%/
1197 007141  123  045  124       .ASCII  /S%T/
1198 007144  045  123  045       .ASCII  /%S%/
1199 007147  124  000  000       .ASCII  /T/<00><00>
1200 007152  045  116  045  P.ABC: .ASCII  /%N%/
1201 007155  124  045  123       .ASCII  /T%S/
1202 007160  045  124  045       .ASCII  /%T%/
1203 007163  123  045  124       .ASCII  /S%T/
1204 007166  045  123  045       .ASCII  /%S%/
1205 007171  124  045  123       .ASCII  /T%S/
1206 007174  045  124  000       .ASCII  /%T/<00>
```

```
                                        ;MLX4
                                        ;                ML11 INTERRUPT SERVICE ROUTINE
1208
1209
1210
1211 007177  000                                 .ASCII  <00>
1212 007200  102  105  107  P.ABD:               .ASCII  /BEG/
1213 007203  111  116  000                       .ASCII  /IN/<00>
1214 007206  105  116  104  P.ABE:               .ASCII  /END/
1215 007211  000                                 .ASCII  <00>
1216 007212  120  101  123  P.ABF:               .ASCII  /PAS/
1217 007215  123  000  000                       .ASCII  /S/<00><00>
1218 007220  115  114  061  P.ABG:               .ASCII  /ML1/
1219 007223  061  055  101                       .ASCII  /1-A/
1220 007226  000  000                            .ASCII  <00><00>
1221 007230  115  114  061  P.ABH:               .ASCII  /ML1/
1222 007233  061  055  102                       .ASCII  /1-B/
1223 007236  000  000                            .ASCII  <00><00>
1224 007240  104  122  111  P.ABI:               .ASCII  /DRI/
1225 007243  126  105  000                       .ASCII  /VE/<00>
1226 007246  123  105  103  P.ABJ:               .ASCII  /SEC/
1227 007251  124  117  122                       .ASCII  /TOR/
1228 007254  000  000                            .ASCII  <00><00>
1229 007256  127  122  111  P.ABK:               .ASCII  /WRI/
1230 007261  124  105  000                       .ASCII  /TE/<00>
1231 007264  122  105  101  P.ABL:               .ASCII  /REA/
1232 007267  104  000  000                       .ASCII  /D/<00><00>
1233 007272  122  105  124  P.ABM:               .ASCII  /RET/
1234 007275  122  131  000                       .ASCII  /RY/<00>
1235 007300  123  125  103  P.ABN:               .ASCII  /SUC/
1236 007303  103  105  105                       .ASCII  /CEE/
1237 007306  104  105  104                       .ASCII  /DED/
1238 007311  000                                 .ASCII  <00>
1239 007312  106  101  111  P.ABO:               .ASCII  /FAI/
1240 007315  114  105  104                       .ASCII  /LED/
1241 007320  000  000                            .ASCII  <00><00>
1242 007322  104  122  117  P.ABP:               .ASCII  /DRO/
1243 007325  120  120  105                       .ASCII  /PPE/
1244 007330  104  000                            .ASCII  /D/<00>
1245 007332  125  120  000  P.ABQ:               .ASCII  /UP/<00>
1246 007335  000                                 .ASCII  <00>
1247 007336  104  117  127  P.ABR:               .ASCII  /DOW/
1248 007341  116  000  000                       .ASCII  /N/<00><00>
1249 007344  122  125  116  P.ABS:               .ASCII  /RUN/
1250 007347  116  111  116                       .ASCII  /NIN/
1251 007352  107  000                            .ASCII  /G/<00>
1252 007354  115  101  122  P.ABT:               .ASCII  /MAR/
1253 007357  103  110  111                       .ASCII  /CHI/
1254 007362  116  107  072                       .ASCII  /NG:/
1255 007365  040  040  000                       .ASCII  /  /<00>
1256 007370  105  116  101  P.ABU:               .ASCII  /ENA/
1257 007373  102  114  105                       .ASCII  /BLE/
1258 007376  104  000                            .ASCII  /D/<00>
1259 007400  104  111  123  P.ABV:               .ASCII  /DIS/
1260 007403  101  102  114                       .ASCII  /ABL/
1261 007406  105  104  000                       .ASCII  /ED/<00>
1262 007411  000                                 .ASCII  <00>
```

```
                                        ;MLX4
                                        ;               ML11 INTERRUPT SERVICE ROUTINE
1264
1265
1266
1267 007412  105  103  103  P.ABW:  .ASCII  /ECC/
1268 007415  000                    .ASCII  <00>
1269 007416  127  111  124  P.ABX:  .ASCII  /WIT/
1270 007421  110  000  000          .ASCII  /H/<00><00>
1271 007424  101  116  104  P.ABY:  .ASCII  /AND/
1272 007427  000                    .ASCII  <00>
1273 007430  116  105  104  P.ABZ:  .ASCII  /NED/
1274 007433  000                    .ASCII  <00>
1275 007434  116  105  115  P.ACA:  .ASCII  /NEM/
1276 007437  000                    .ASCII  <00>
1277 007440  120  107  105  P.ACB:  .ASCII  /PGE/
1278 007443  000                    .ASCII  <00>
1279 007444  104  114  124  P.ACC:  .ASCII  /DLT/
1280 007447  000                    .ASCII  <00>
1281 007450  127  103  105  P.ACD:  .ASCII  /WCE/
1282 007453  000                    .ASCII  <00>
1283 007454  120  105  000  P.ACE:  .ASCII  /PE/<00>
1284 007457  000                    .ASCII  <00>
1285 007460  115  130  106  P.ACF:  .ASCII  /MXF/
1286 007463  000                    .ASCII  <00>
1287 007464  115  104  120  P.ACG:  .ASCII  /MDP/
1288 007467  105  000  000          .ASCII  /E/<00><00>
1289 007472  115  103  120  P.ACH:  .ASCII  /MCP/
1290 007475  105  000  000          .ASCII  /E/<00><00>
1291 007500  125  116  123  P.ACI:  .ASCII  /UNS/
1292 007503  000                    .ASCII  <00>
1293 007504  111  101  105  P.ACJ:  .ASCII  /IAE/
1294 007507  000                    .ASCII  <00>
1295 007510  101  117  105  P.ACK:  .ASCII  /AOE/
1296 007513  000                    .ASCII  <00>
1297 007514  122  115  122  P.ACL:  .ASCII  /RMR/
1298 007517  000                    .ASCII  <00>
1299 007520  111  114  122  P.ACM:  .ASCII  /ILR/
1300 007523  000                    .ASCII  <00>
1301 007524  111  114  106  P.ACN:  .ASCII  /ILF/
1302 007527  000                    .ASCII  <00>
1303 007530  117  120  111  P.ACO:  .ASCII  /OPI/
1304 007533  000                    .ASCII  <00>
1305 007534  104  120  101  P.ACP:  .ASCII  /DPA/
1306 007537  122  000  000          .ASCII  /R/<00><00>
1307 007542  103  120  101  P.ACQ:  .ASCII  /CPA/
1308 007545  122  000  000          .ASCII  /R/<00><00>
1309 007550  104  103  113  P.ACR:  .ASCII  /DCK/
1310 007553  000                    .ASCII  <00>
1311 007554  105  103  110  P.ACS:  .ASCII  /ECH/
1312 007557  000                    .ASCII  <00>
1313 007560  103  122  103  P.ACT:  .ASCII  /CRC/
1314 007563  000                    .ASCII  <00>
1315 007564  123  107  114  P.ACU:  .ASCII  /SGL/
1316 007567  000                    .ASCII  <00>
1317 007570  125  116  103  P.ACV:  .ASCII  /UNC/
1318 007573  000                    .ASCII  <00>
```

```
                                        ;MLX4
                                        ;               ML11 INTERRUPT SERVICE ROUTINE          27-Mar-1982 19:24:42    TOPS
                                                                                                27-Mar-1982 19:23:44    PA:<
007574  103  117  115  P.ACW:  .ASCII  /COM/
007577  115  101  116          .ASCII  /MAN/
007602  104  040  111          .ASCII  /D I/
007605  116  124  105          .ASCII  /NTE/
007610  107  122  111          .ASCII  /GRI/
007613  124  131  040          .ASCII  /TY /
007616  122  117  125          .ASCII  /ROU/
007621  124  111  116          .ASCII  /TIN/
007624  105  000              .ASCII  /E/<00>
007626  117  120  124  P.ACX:  .ASCII  /OPT/
007631  061  000  000          .ASCII  /1/<00><00>
007634  117  120  124  P.ACY:  .ASCII  /OPT/
007637  062  000  000          .ASCII  /2/<00><00>
007642  117  120  124  P.ACZ:  .ASCII  /OPT/
007645  063  000  000          .ASCII  /3/<00><00>
007650  117  120  124  P.ADA:  .ASCII  /OPT/
007653  064  000  000          .ASCII  /4/<00><00>
007656  117  120  124  P.ADB:  .ASCII  /OPT/
007661  065  000  000          .ASCII  /5/<00><00>
007664  122  101  116  P.ADC:  .ASCII  /RAN/
007667  104  061  000          .ASCII  /D1/<00>
007672  122  101  116  P.ADD:  .ASCII  /RAN/
007675  104  062  000          .ASCII  /D2/<00>
007700  122  101  116  P.ADE:  .ASCII  /RAN/
007703  104  063  000          .ASCII  /D3/<00>
007706  122  101  116  P.ADF:  .ASCII  /RAN/
007711  104  064  000          .ASCII  /D4/<00>
007714  127  122  111  P.ADG:  .ASCII  /WRI/
007717  124  105  040          .ASCII  /TE /
007722  103  110  105          .ASCII  /CHE/
007725  103  113  000          .ASCII  /CK/<00>
007730  121  125  111  P.ADH:  .ASCII  /QUI/
007733  103  113  040          .ASCII  /CK /
007736  126  105  122          .ASCII  /VER/
007741  111  106  131          .ASCII  /IFY/
007744  000  000              .ASCII  <00><00>
007746  122  105  106  P.ADI:  .ASCII  /REF/
007751  122  105  123          .ASCII  /RES/
007754  110  040  115          .ASCII  /H M/
007757  101  122  107          .ASCII  /ARG/
007762  111  116  111          .ASCII  /INI/
007765  116  107  000          .ASCII  /NG/<00>
007770  103  123  122  P.ADJ:  .ASCII  /CSR/
007773  040  101  104          .ASCII  / AD/
007776  104  122  105          .ASCII  /DRE/
010001  123  123  000          .ASCII  /SS/<00>
010004  123  117  106  P.ADK:  .ASCII  /SOF/
010007  124  040  105          .ASCII  /T E/
010012  122  122  117          .ASCII  /RRO/
010015  122  040  103          .ASCII  /R C/
010020  117  125  116          .ASCII  /OUN/
010023  124  000  000          .ASCII  /T/<00><00>
```

```
1376
1377                                    ;MLX4
1378                                    ;               ML11 INTERRUPT SERVICE ROUTINE
1379 010026    124     122     101  P.ADL:  .ASCII  /TRA/
1380 010031    116     123     106          .ASCII  /NSF/
1381 010034    105     122     040          .ASCII  /ER /
1382 010037    122     105     124          .ASCII  /RET/
1383 010042    122     111     105          .ASCII  /RIE/
1384 010045    123     000     000          .ASCII  /S/<00><00>
1385 010050    114     117     107  P.ADM:  .ASCII  /LOG/
1386 010053    111     103     101          .ASCII  /ICA/
1387 010056    114     040     125          .ASCII  /L U/
1388 010061    116     111     124          .ASCII  /NIT/
1389 010064    000     000                  .ASCII  <00><00>
1390 010066    123     105     122  P.ADN:  .ASCII  /SER/
1391 010071    111     101     114          .ASCII  /IAL/
1392 010074    040     043     000          .ASCII  / #/<00>
1393 010077    000                          .ASCII  <00>
1394 010100    120     101     124  P.ADO:  .ASCII  /PAT/
1395 010103    124     105     122          .ASCII  /TER/
1396 010106    116     040     116          .ASCII  /N N/
1397 010111    125     115     102          .ASCII  /UMB/
1398 010114    105     122     000          .ASCII  /ER/<00>
1399 010117    000                          .ASCII  <00>
1400 010120    105     122     122  P.ADP:  .ASCII  /ERR/
1401 010123    117     122     040          .ASCII  /OR /
1402 010126    102     111     124          .ASCII  /BIT/
1403 010131    123     040     123          .ASCII  /S S/
1404 010134    105     124     072          .ASCII  /ET:/
1405 010137    000                          .ASCII  <00>
1406 010140    123     103     040  P.ADQ:  .ASCII  /SC /
1407 010143    123     105     124          .ASCII  /SET/
1408 010146    040     102     125          .ASCII  / BU/
1409 010151    124     040     116          .ASCII  /T N/
1410 010154    117     040     123          .ASCII  /O S/
1411 010157    131     123     124          .ASCII  /YST/
1412 010162    105     115     040          .ASCII  /EM /
1413 010165    105     122     122          .ASCII  /ERR/
1414 010170    117     122     123          .ASCII  /ORS/
1415 010173    040     106     117          .ASCII  / FO/
1416 010176    125     116     104          .ASCII  /UND/
1417 010201    000                          .ASCII  <00>
1418 010202    116     125     115  P.ADR:  .ASCII  /NUM/
1419 010205    102     105     122          .ASCII  /BER/
1420 010210    040     117     106          .ASCII  / OF/
1421 010213    040     115     102          .ASCII  / MB/
1422 010216    131     124     105          .ASCII  /YTE/
1423 010221    123     040     124          .ASCII  /S T/
1424 010224    122     101     116          .ASCII  /RAN/
1425 010227    123     106     105          .ASCII  /SFE/
1426 010232    122     105     104          .ASCII  /RED/
1427 010235    072     000     000          .ASCII  /:/<00><00>
1428 010240    115     102     131  P.ADS:  .ASCII  /MBY/
1429 010243    124     105     123          .ASCII  /TES/
1430 010246    040     127     122          .ASCII  / WR/
```

```
                                        :MLX4
                                        :                ML11 INTERRUPT SERVICE ROUTINE
1432
1433
1434
1435 010251  111  124  105             .ASCII  /ITE/
1436 010254  040  103  110             .ASCII  / CH/
1437 010257  105  103  113             .ASCII  /ECK/
1438 010262  105  104  000             .ASCII  /ED/<00>
1439 010265  000                       .ASCII  <00>
1440 010266  124  117  120  P.ADT:     .ASCII  /TOP/
1441 010271  040  123  105             .ASCII  / SE/
1442 010274  103  124  117             .ASCII  /CTO/
1443 010277  122  040  117             .ASCII  /R O/
1444 010302  106  000                  .ASCII  /F/<00>
1445 010304  114  117  127  P.ADU:     .ASCII  /LOW/
1446 010307  040  123  105             .ASCII  / SE/
1447 010312  103  124  117             .ASCII  /CTO/
1448 010315  122  040  117             .ASCII  /R O/
1449 010320  106  000                  .ASCII  /F/<00>
1450 010322  123  131  123  P.ADV:     .ASCII  /SYS/
1451 010325  124  105  115             .ASCII  /TEM/
1452 010330  040  114  111             .ASCII  / LI/
1453 010333  115  111  124             .ASCII  /MIT/
1454 010336  040  117  106             .ASCII  / OF/
1455 010341  000                       .ASCII  <00>
1456 010342  120  105  122  P.ADW:     .ASCII  /PER/
1457 010345  106  117  122             .ASCII  /FOR/
1458 010350  115  101  116             .ASCII  /MAN/
1459 010353  103  105  040             .ASCII  /CE /
1460 010356  123  125  115             .ASCII  /SUM/
1461 010361  115  101  122             .ASCII  /MAR/
1462 010364  131  000                  .ASCII  /Y/<00>
1463 010366  110  101  122  P.ADX:     .ASCII  /HAR/
1464 010371  104  040  105             .ASCII  /D E/
1465 010374  122  122  117             .ASCII  /RRO/
1466 010377  122  040  103             .ASCII  /R C/
1467 010402  117  125  116             .ASCII  /OUN/
1468 010405  124  000  000             .ASCII  /T/<00><00>
1469 010410  115  102  131  P.ADY:     .ASCII  /MBY/
1470 010413  124  105  123             .ASCII  /TES/
1471 010416  040  127  122             .ASCII  / WR/
1472 010421  111  124  124             .ASCII  /ITT/
1473 010424  105  116  000             .ASCII  /EN/<00>
1474 010427  000                       .ASCII  <00>
1475 010430  115  102  131  P.ADZ:     .ASCII  /MBY/
1476 010433  124  105  123             .ASCII  /TES/
1477 010436  040  122  105             .ASCII  / RE/
1478 010441  101  104  000             .ASCII  /AD/<00>
1479 010444  062  000       P.AEA:     .ASCII  /2/<00>
1480 010446  061  000       P.AEB:     .ASCII  /1/<00>
1481 010450  056  065  000  P.AEC:     .ASCII  /.5/<00>
1482 010453  000                       .ASCII  <00>
1483 010454  056  062  065  P.AED:     .ASCII  /.25/
1484 010457  000                       .ASCII  <00>
1485 010460  050  116  117  P.AEE:     .ASCII  /(NO/
1486 010463  124  040  120             .ASCII  /T P/
```

27-Mar-1982 19:24:42 TOPS
27-Mar-1982 19:23:44 PA:<

```
                                        ;MLX4
                                        ;                ML11 INTERRUPT SERVICE ROUTINE
                                                                                          27-Mar-1982 19:24:42    TOPS
                                                                                          27-Mar-1982 19:23:44    PA:<
 1488
 1489
 1490
 1491 010466    117     127     105             .ASCII  /OWE/
 1492 010471    122     105     104             .ASCII  /RED/
 1493 010474    040     125     120             .ASCII  / UP/
 1494 010477    051     000     000             .ASCII  /)/<00><00>
 1495 010502    050     116     117     P.AEF:  .ASCII  /(NO/
 1496 010505    124     040     101             .ASCII  /T A/
 1497 010510    116     040     115             .ASCII  /N M/
 1498 010513    114     061     061             .ASCII  /L11/
 1499 010516    040     125     116             .ASCII  / UN/
 1500 010521    111     124     051             .ASCII  /IT)/
 1501 010524    000     000                     .ASCII  <00><00>
 1502 010526    050     117     120     P.AEG:  .ASCII  /(OP/
 1503 010531    105     122     101             .ASCII  /ERA/
 1504 010534    124     117     122             .ASCII  /TOR/
 1505 010537    040     123     105             .ASCII  / SE/
 1506 010542    114     105     103             .ASCII  /LEC/
 1507 010545    124     105     104             .ASCII  /TED/
 1508 010550    040     124     105             .ASCII  / TE/
 1509 010553    123     124     040             .ASCII  /ST /
 1510 010556    114     111     115             .ASCII  /LIM/
 1511 010561    111     124     123             .ASCII  /ITS/
 1512 010564    040     111     116             .ASCII  / IN/
 1513 010567    103     117     122             .ASCII  /COR/
 1514 010572    122     105     103             .ASCII  /REC/
 1515 010575    124     114     131             .ASCII  /TLY/
 1516 010600    051     000                     .ASCII  )/<00>
 1517 010602    050     101     114     P.AEH:  .ASCII  /(AL/
 1518 010605    114     040     122             .ASCII  /L R/
 1519 010610    105     124     122             .ASCII  /ETR/
 1520 010613    111     105     123             .ASCII  /IES/
 1521 010616    040     106     101             .ASCII  / FA/
 1522 010621    111     114     105             .ASCII  /ILE/
 1523 010624    104     040     106             .ASCII  /D F/
 1524 010627    117     122     040             .ASCII  /OR /
 1525 010632    101     040     116             .ASCII  /A N/
 1526 010635    117     116     055             .ASCII  /ON-/
 1527 010640    106     101     124             .ASCII  /FAT/
 1528 010643    101     114     040             .ASCII  /AL /
 1529 010646    105     122     122             .ASCII  /ERR/
 1530 010651    117     122     051             .ASCII  /OR)/
 1531 010654    000     000                     .ASCII  <00><00>
 1532 010656    050     103     117     P.AEI:  .ASCII  /(CO/
 1533 010661    116     124     122             .ASCII  /NTR/
 1534 010664    117     114     114             .ASCII  /OLL/
 1535 010667    105     122     040             .ASCII  /ER /
 1536 010672    106     101     124             .ASCII  /FAT/
 1537 010675    101     114     040             .ASCII  /AL /
 1538 010700    105     122     122             .ASCII  /ERR/
 1539 010703    117     122     051             .ASCII  /OR)/
 1540 010706    000     000                     .ASCII  <00><00>
 1541 010710    050     104     122     P.AEJ:  .ASCII  /(DR/
 1542 010713    111     126     105             .ASCII  /IVE/
```

```
                                        ;MLX4
                                        ;               ML11 INTERRUPT SERVICE ROUTINE
010716  040  106  101                   .ASCII  / FA/
010721  124  101  114                   .ASCII  /TAL/
010724  040  105  122                   .ASCII  / ER/
010727  122  117  122                   .ASCII  /ROR/
010732  051  000                        .ASCII  )/<00>
010734  050  105  103           P.AEK:  .ASCII  /(EC/
010737  103  040  110                   .ASCII  /C H/
010742  101  122  104                   .ASCII  /ARD/
010745  040  105  122                   .ASCII  / ER/
010750  122  117  122                   .ASCII  /ROR/
010753  051  000  000                   .ASCII  )/<00><00>
010756  050  105  103           P.AEL:  .ASCII  /(EC/
010761  103  040  114                   .ASCII  /C L/
010764  117  107  111                   .ASCII  /OGI/
010767  103  040  106                   .ASCII  /C F/
010772  101  111  114                   .ASCII  /AIL/
010775  125  122  105                   .ASCII  /URE/
011000  051  000                        .ASCII  )/<00>
011002  111  116  124           P.AEM:  .ASCII  /INT/
011005  105  122  122                   .ASCII  /ERR/
011010  125  120  124                   .ASCII  /UPT/
011013  040  104  111                   .ASCII  / DI/
011016  104  040  116                   .ASCII  /D N/
011021  117  124  040                   .ASCII  /OT /
011024  117  103  103                   .ASCII  /OCC/
011027  125  122  054                   .ASCII  /UR,/
011032  040  102  125                   .ASCII  / BU/
011035  124  040  124                   .ASCII  /T T/
011040  110  105  040                   .ASCII  /HE /
011043  124  122  101                   .ASCII  /TRA/
011046  116  123  106                   .ASCII  /NSF/
011051  105  122  040                   .ASCII  /ER /
011054  111  123  040                   .ASCII  /IS /
011057  103  117  115                   .ASCII  /COM/
011062  120  114  105                   .ASCII  /PLE/
011065  124  105  000                   .ASCII  /TE/<00>
011070  055  055  076           P.AEN:  .ASCII  /-->/
011073  040  122  125                   .ASCII  / RU/
011076  116  040  115                   .ASCII  /N M/
011101  114  061  061                   .ASCII  /L11/
011104  040  114  117                   .ASCII  / LO/
011107  107  111  103                   .ASCII  /GIC/
011112  040  124  105                   .ASCII  / TE/
011115  123  124  000                   .ASCII  /ST/<00>
011120  055  055  076           P.AEO:  .ASCII  /-->/
011123  040  122  125                   .ASCII  / RU/
011126  116  040  115                   .ASCII  /N M/
011131  114  061  061                   .ASCII  /L11/
011134  040  120  122                   .ASCII  / PR/
011137  117  115  040                   .ASCII  /OM /
011142  115  101  111                   .ASCII  /MAI/
011145  116  124  105                   .ASCII  /NTE/
```

```
                                        ;MLX4
                                        ;               ML11 INTERRUPT SERVICE ROUTINE

011150  116  101  116                   .ASCII  /NAN/
011153  103  105  040                   .ASCII  /CE /
011156  120  122  117                   .ASCII  /PRO/
011161  107  122  101                   .ASCII  /GRA/
011164  115  000                       .ASCII  /M/<00>
011166  123  117  106           P.AEP:  .ASCII  /SOF/
011171  124  040  105                   .ASCII  /T E/
011174  122  122  117                   .ASCII  /RRO/
011177  122  000  000                   .ASCII  /R/<00><00>
011202  110  101  122           P.AEQ:  .ASCII  /HAR/
011205  104  040  105                   .ASCII  /D E/
011210  122  122  117                   .ASCII  /RRO/
011213  122  000  000                   .ASCII  /R/<00><00>
011216  105  103  103           P.AER:  .ASCII  /ECC/
011221  040  114  117                   .ASCII  / LO/
011224  107  111  103                   .ASCII  /GIC/
011227  040  106  101                   .ASCII  / FA/
011232  111  114  105                   .ASCII  /ILE/
011235  104  040  124                   .ASCII  /D T/
011240  117  040  104                   .ASCII  /O D/
011243  105  124  105                   .ASCII  /ETE/
011246  103  124  040                   .ASCII  /CT /
011251  104  101  124                   .ASCII  /DAT/
011254  101  040  105                   .ASCII  /A E/
011257  122  122  117                   .ASCII  /RRO/
011262  122  000                       .ASCII  /R/<00>


011264                          SBE.LOG:.BLKW   400
012264                          PM.SBE.CNT:
012264                                  .BLKW   200
012664                          SBE$.COUNT:
012664                                  .BLKW   1
012666                          BIT.NUM:.BLKW   1
012670                          WBUFF:  .BLKW   4000
022670                          RBUFF:  .BLKW   4000
032670                          WPTR:   .BLKW   1
032672                          RPTR:   .BLKW   1
032674                          QUICK:  .BLKW   1
032676                          PATTERN:.BLKW   1
032700                          DATA.COUNT:
032700                                  .BLKW   1
032702                          COMP.COUNT:
032702                                  .BLKW   1
032704                          EOP.COUNT:
032704                                  .BLKW   1
032706                          BASE.ADDR:
032706                                  .BLKW   1
032710                          VEC:    .BLKW   1
032712                          BR.LEVEL:
```

```
                                ;MLX4
                                ;               ML11 INTERRUPT SERVICE ROUTINE

                                        .BLKW   1
                                SOFTS:  .BLKW   200
                                HARDS:  .BLKW   200
                                TRIES:  .BLKW   200
                                WR.COUNT:
                                        .BLKW   1
                                WR.THOUSANDS:
                                        .BLKW   1
                                WR.MILLIONS:
                                        .BLKW   1
                                RD.COUNT:
                                        .BLKW   1
                                RD.THOUSANDS:
                                        .BLKW   1
                                RD.MILLIONS:
                                        .BLKW   1
                                WC.COUNT:
                                        .BLKW   1
                                WC.THOUSANDS:
                                        .BLKW   1
                                WC.MILLIONS:
                                        .BLKW   1
                                I.AM.DONE:
                                        .BLKW   1
                                RETRYING:
                                        .BLKW   1
                                BOARD:  .BLKW   1
                                BANK:   .BLKW   1


                                ML.REG::.BLKW   26
                                PTABLE.ADDR::
                                        .BLKW   10
                                DRIVE.STATUS::
                                        .BLKB   1
                                        .EVEN
                                DROPT.DRIVES::
                                        .BLKB   1
                                        .EVEN
                                WHY.DROPT::
                                        .BLKW   4
                                NUM.DRIVES::
                                        .BLKW   1
                                LOW.SECT::
                                        .BLKW   10
                                TOP.SECT::
                                        .BLKW   10
034520  000000                  PATTBL::.WORD   0
034522  002000                          .WORD   2000
034524  000001                          .WORD   1
```

```
1710                                    ;MLX4
1711                                    ;               ML11 INTERRUPT SERVICE ROUTINE
1712
1713 034526  000001                             .WORD   1
1714 034530  000004                             .WORD   4
1715 034532  000004                             .WORD   4
1716 034534  000011                             .WORD   11
1717 034536  000011                             .WORD   11
1718 034540  002000                             .WORD   2000
1719 034542  002000                             .WORD   2000
1720
1721
1722                                            .GLOBL  EFN$21, L$UNIT, L$LUN, LIMIT, RANGE
1723                                            .GLOBL  LSECT, TSECT, ONLY, DROPNE, DROP1
1724                                            .GLOBL  DROP2, DROP3, DROP4, DROP5, MARPAT
1725                                            .GLOBL  REFRESH, ECCDIS, EOPSUM, ERROUT
1726                                            .GLOBL  SEED1, SEED2, SEED3, RANDOM, RN
1727
1728
1729          100000                    BIT15==         -100000
1730          040000                    BIT14==         40000
1731          020000                    BIT13==         20000
1732          010000                    BIT12==         10000
1733          004000                    BIT11==         4000
1734          002000                    BIT10==         2000
1735          001000                    BIT09==         1000
1736          000400                    BIT08==         400
1737          000200                    BIT07==         200
1738          000100                    BIT06==         100
1739          000040                    BIT05==         40
1740          000020                    BIT04==         20
1741          000010                    BIT03==         10
1742          000004                    BIT02==         4
1743          000002                    BIT01==         2
1744          000001                    BIT00==         1
1745          001000                    BIT9==          1000
1746          000400                    BIT8==          400
1747          000200                    BIT7==          200
1748          000100                    BIT6==          100
1749          000040                    BIT5==          40
1750          000020                    BIT4==          20
1751          000010                    BIT3==          10
1752          000004                    BIT2==          4
1753          000002                    BIT1==          2
1754          000001                    BIT0==          1
1755          000040                    EF.START==      40
1756          000037                    EF.RESTART==    37
1757          000036                    EF.CONTINUE==   36
1758          000035                    EF.NEW==        35
1759          000034                    EF.PWR==        34
1760          000340                    PRI07==         340
1761          000300                    PRI06==         300
1762          000240                    PRI05==         240
1763          000200                    PRI04==         200
1764          000140                    PRI03==         140
1765                                    ;MLX4
1766                                    ;               ML11 INTERRUPT SERVICE ROUTINE
```


27-Mar-1982 19:24:42 TOPS
27-Mar-1982 19:23:44 PA:<

```
1767
1768    000100          PRI02==         100
1769    000040          PRI01==         40
1770    000000          PRI00==         0
1771    000004          EVL==           4
1772    000010          LOT==           10
1773    000020          ADR==           20
1774    000040          IDU==           40
1775    000100          ISR==           100
1776    000200          UAM==           200
1777    000400          BUE==           400
1778    001000          PNT==           1000
1779    002000          PRI==           2000
1780    004000          IXE==           4000
1781    010000          IBE==           10000
1782    020000          IER==           20000
1783    040000          LOE==           40000
1784    100000          HOE==           -100000
1785    012670          WDBUFF=         WBUFF
1786    013670          WCBUFF=         WBUFF+1000
1787    022670          RDBUFF=         RBUFF
1788    023670          RCBUFF=         RBUFF+1000
1789    022670          END.WBUFF=      WBUFF+10000
1790    032670          END.RBUFF=      RBUFF+10000
1791    005662          PHR21=          P.AAA
1792    005736          FMT16=          P.AAB
1793    006042          SBE$HEADER=     P.AAC
1794    006126          FMT1A=          P.AAD
1795    006142          FMT1B=          P.AAE
1796    006160          FMT2=           P.AAF
1797    006166          FMT3=           P.AAG
1798    006216          FMT4A=          P.AAH
1799    006252          FMT4B=          P.AAI
1800    006270          FMT4C=          P.AAJ
1801    006320          FMT5=           P.AAK
1802    006376          FMT6=           P.AAL
1803    006446          FMT7=           P.AAM
1804    006466          FMT8=           P.AAN
1805    006534          FMT9=           P.AAO
1806    006564          FMT10A=         P.AAP
1807    006640          FMT10B=         P.AAQ
1808    006654          FMT11=          P.AAR
1809    006710          FMT12A=         P.AAS
1810    006756          FMT12B=         P.AAT
1811    007024          FMT13=          P.AAU
1812    007046          FMT14=          P.AAV
1813    007060          FMT15=          P.AAW
1814    007066          CRLF=           P.AAX
1815    007072          SAY1=           P.AAY
1816    007100          SAY2=           P.AAZ
1817    007112          SAY3=           P.ABA
1818    007130          SAY4=           P.ABB
1819    007152          SAY5=           P.ABC
1820                    ;MLX4                                   27-Mar-1982 19:24:42    TOPS
1821                    ;       ML11 INTERRUPT SERVICE ROUTINE  27-Mar-1982 19:23:44    PA:<
1822
1823    007200          WRD2=           P.ABD
```

```
1824    007206          WRD3=           P.ABE
1825    007212          WRD4=           P.ABF
1826    007220          WRD6=           P.ABG
1827    007230          WRD7=           P.ABH
1828    007240          WRD11=          P.ABI
1829    007246          WRD15=          P.ABJ
1830    007256          WRD16=          P.ABK
1831    007264          WRD17=          P.ABL
1832    007272          WRD18=          P.ABM
1833    007300          WRD19=          P.ABN
1834    007312          WRD20=          P.ABO
1835    007322          WRD21=          P.ABP
1836    007332          WRD24=          P.ABQ
1837    007336          WRD25=          P.ABR
1838    007344          WRD34=          P.ABS
1839    007354          WRD35=          P.ABT
1840    007370          WRD36=          P.ABU
1841    007400          WRD37=          P.ABV
1842    007412          WRD38=          P.ABW
1843    007416          WRD40=          P.ABX
1844    007424          WRD41=          P.ABY
1845    007430          MLB2=           P.ABZ
1846    007434          MLB3=           P.ACA
1847    007440          MLB4=           P.ACB
1848    007444          MLB5=           P.ACC
1849    007450          MLB6=           P.ACD
1850    007454          MLB7=           P.ACE
1851    007460          MLB8=           P.ACF
1852    007464          MLB9=           P.ACG
1853    007472          MLB10=          P.ACH
1854    007500          MLB11=          P.ACI
1855    007504          MLB12=          P.ACJ
1856    007510          MLB13=          P.ACK
1857    007514          MLB14=          P.ACL
1858    007520          MLB15=          P.ACM
1859    007524          MLB16=          P.ACN
1860    007530          MLB17=          P.ACO
1861    007534          MLB18=          P.ACP
1862    007542          MLB19=          P.ACQ
1863    007550          MLB20=          P.ACR
1864    007554          MLB21=          P.ACS
1865    007560          MLB22=          P.ACT
1866    007564          MLB23=          P.ACU
1867    007570          MLB24=          P.ACV
1868    007574          RTN0=           P.ACW
1869    007626          RTN1=           P.ACX
1870    007634          RTN2=           P.ACY
1871    007642          RTN3=           P.ACZ
1872    007650          RTN4=           P.ADA
1873    007656          RTN5=           P.ADB
1874    007664          RTN5A=          P.ADC
1875                    ;MLX4                                   27-Mar-1982 19:24:42    TOPS
1876                    ;       ML11 INTERRUPT SERVICE ROUTINE  27-Mar-1982 19:23:44    PA:<
1877
1878    007672          RTN5B=          P.ADD
1879    007700          RTN5C=          P.ADE
1880    007706          RTN5D=          P.ADF
```

```
 1881        007714                  PHR1=           P.ADG
 1882        007730                  PHR2=           P.ADH
 1883        007746                  PHR3=           P.ADI
 1884        007770                  PHR4=           P.ADJ
 1885        010004                  PHR5=           P.ADK
 1886        010026                  PHR6=           P.ADL
 1887        010050                  PHR7=           P.ADM
 1888        010066                  PHR8=           P.ADN
 1889        010100                  PHR9=           P.ADO
 1890        010120                  PHR10=          P.ADP
 1891        010140                  PHR11=          P.ADQ
 1892        010202                  PHR12=          P.ADR
 1893        010240                  PHR13=          P.ADS
 1894        010266                  PHR14=          P.ADT
 1895        010304                  PHR15=          P.ADU
 1896        010322                  PHR16=          P.ADV
 1897        010342                  PHR17=          P.ADW
 1898        010366                  PHR18=          P.ADX
 1899        010410                  PHR19=          P.ADY
 1900        010430                  PHR20=          P.ADZ
 1901        010444                  TRT00=          P.AEA
 1902        010446                  TRT01=          P.AEB
 1903        010450                  TRT10=          P.AEC
 1904        010454                  TRT11=          P.AED
 1905        010460                  CAUSE1=         P.AEE
 1906        010502                  CAUSE2=         P.AEF
 1907        010526                  CAUSE3=         P.AEG
 1908        010602                  CAUSE4=         P.AEH
 1909        010656                  CAUSE5=         P.AEI
 1910        010710                  CAUSE6=         P.AEJ
 1911        010734                  CAUSE7=         P.AEK
 1912        010756                  CAUSE8=         P.AEL
 1913        011002                  MSG0=           P.AEM
 1914        011070                  MSG1=           P.AEN
 1915        011120                  MSG2=           P.AEO
 1916        011166                  MSG3=           P.AEP
 1917        011202                  MSG4=           P.AEQ
 1918        011216                  MSG5=           P.AER
 1919
 1920
 1921                                        .SBTTL  SERVICE ML11 INTERRUPT SERVICE ROUTINE
 1922
 1926 034544                         SERVICE::
 1927 034544  012767  000001  177564          MOV     #1,I.AM.DONE                    ;                                       2346
 1928 034552  000002                          RTI                                     ;                                       2345
 1929                                :MLX4
 1930                                ;                       ML11 INTERRUPT SERVICE ROUTINE          27-Mar-1982 19:24:42    TOPS
 1931                                                                                                    27-Mar-1982 19:23:44    PA:<
 1932
 1933                                ; Routine Size:  4 words
 1934                                ; Maximum stack depth per invocation:  0 words
 1939
 1940
 1941 ;MLX4
 1942 ;                     ONCE-ONLY CODE                                  27-Mar-1982 19:24:42    TOPS-20 Bliss-16 V2(212)
 1943                                                                        27-Mar-1982 19:23:44    PA:<NEALE>MLX4.BLI.5 (5)
 1944 ;              2348   %sbttl 'ONCE-ONLY CODE'
```

```
;          2349  routine CLRTBLS : novalue =
;          2350      begin                                          !* 1 *
;          2351
;          2352  !++
;          2353  !   ROUTINE:     CLRTBLS
;          2354  !
;          2355  !   PURPOSE:     THIS ROUTINE IS CALLED BY THE INITIALIZATION CODE WHEN
;          2356  !                A 'START' OR 'RESTART' COMMAND HAS BEEN USED TO BEGIN
;          2357  !                THE PROGRAM.  THE PURPOSES OF THE ROUTINE ARE:
;          2358  !
;          2359  !                (1)  TO INITIALIZE STATISTICS TABLES
;          2360  !
;          2361  !                (2)  TO SET ALL DRIVE AND ARRAY STATUS LOCATIONS TO ACTIVE
;          2362  !                     NOTE:  EVEN IF A UNIT WAS DROPPED DURING A PREVIOUS
;          2363  !                            RUN, THE START OR RESTART COMMAND GUARANTEES
;          2364  !                            THAT THE DRIVE WILL ONCE AGAIN BE TESTABLE.
;          2365  !
;          2366  !                (3)  TO ENABLE THE RUNNING OF ALL TEST OPTIONS IF THE
;          2367  !                     OPERATOR WANTED THEM ALL TO RUN.
;          2368  !--
;          2369
;          2370  !+
;          2371  !THIS IS THE CODE FOR PURPOSE 1:
;          2372  !-
;          2373
;          2374      incr LUN from 0 to (.LSUNIT - 1) do
;          2375          begin
;          2376
;          2377          incr ARRAY from 0 to 15 do
;          2378              begin
;          2379              SOFTS [.LUN, .ARRAY, 0, 16, 0] = 0;
;          2380              HARDS [.LUN, .ARRAY, 0, 16, 0] = 0;
;          2381              TRIES [.LUN, .ARRAY, 0, 16, 0] = 0;
;          2382              end;
;          2383
;          2384          end;
;          2385
;          2386      SBES_COUNT = 0;                                ! VER CZMLBB ADDED CLEARING THIS VARIABLE
;          2387      EOP_COUNT = 0;
;          2388      NUM_DRIVES = 0;
;          2389      RETRYING = INACTIVE;
;          2390      WR_COUNT = 0;
;          2391      WR_THOUSANDS = 0;
;          2392      WR_MILLIONS = 0;
;          2393      RD_COUNT = 0;
;          2394      RD_THOUSANDS = 0;
;          2395      RD_MILLIONS = 0;
;          2396      WC_COUNT = 0;
;          2397      WC_THOUSANDS = 0;
;          2398      WC_MILLIONS = 0;
;          2399  !+
;MLX4                                                             27-Mar-1982 19:24:42    TOPS-20 Bliss-16 V2(212)
;          ONCE-ONLY CODE                                         27-Mar-1982 19:23:44    PA:<NEALE>MLX4.BLI.5 (5)
;
;          2400  !THIS IS THE CODE FOR PURPOSE 2:
;          2401  !-
;          2402
```

```
2002 ;          2403        incr LUN from 0 to (.L$UNIT - 1) do
2003 ;          2404            begin                                       !* 2 *
2004 ;          2405            L$LUN = .LUN;
2005 ;          2406            LOW_SECT [.LUN] = 0;
2006 ;          2407            DRIVE_STATUS [.LUN] = ACTIVE;
2007 ;          2408            DROPT_DRIVES [.LUN] = INACTIVE;
2008 ;          2409            WHY_DROPT [.LUN] = 0;
2009 ;          2410            end;                                        !* 2 *
2010 ;          2411
2011 ;          2412    !+
2012 ;          2413    !THIS IS THE CODE FOR PURPOSE 3:
2013 ;          2414    !-
2014 ;          2415
2015 ;          2416        if .DROPNE eql 0
2016 ;          2417        then
2017 ;          2418            begin                                       !* 3 *
2018 ;          2419            DROP1 = 0;
2019 ;          2420            DROP2 = 0;
2020 ;          2421            DROP3 = 0;
2021 ;          2422            DROP4 = 0;
2022 ;          2423            DROP5 = 0;
2023 ;          2424            end;                                        !* 3 *
2024 ;          2425
2025 ;          2426        !+
2026 ;          2427        ! VER CZMLBB ADDED CLEARING OF THIS STRUCTURE
2027 ;          2428        !
2028 ;          2429        ! The single bit error log table stores all
2029 ;          2430        ! detected sbe's detected during the run time
2030 ;          2431        ! of the exerciser.  In this table is stored
2031 ;          2432        ! the sbe's: unit #, board #, bank #, bit #
2032 ;          2433        ! and a count of how many times this sbe has
2033 ;          2434        ! reoccured.
2034 ;          2435        !
2035 ;          2436        ! This code initializes this structure to zeroes
2036 ;          2437        ! before starting execution of the exerciser.
2037 ;          2438        !-
2038 ;          2439
2039 ;          2440        incr index from 0 to 127 do                     !Clear the single bit error log table
2040 ;          2441            begin
2041 ;          2442            SBE_LOG [.index, WRD$_0] = ZERO;            !Clear word zero
2042 ;          2443            SBE_LOG [.index, WRD$_1] = ZERO;            !Clear word one
2043 ;          2444            end;
2044 ;          2445
2045 ;          2446        !+
2046 ;          2447        ! VER CZMLBB ADDED CLEARING OF THIS STRUCTURE
2047 ;          2448        !
2048 ;          2449        ! The prom maintanence program is called out
2049 ;          2450        ! to be run on a particular array module when
2050 ;          2451        ! that array module has collected 10 unique
```

```
;MLX4
;                ONCE-ONLY CODE

;          2452      ! failing chips.  This structure stores for
;          2453      ! each array within each unit its total number
;          2454      ! of unique failing chips.
;          2455      !
;          2456      ! This code initializes this structure to zeroes
;          2457      ! before the exerciser is executed.
;          2458      !-
;          2459
;          2460      incr UNIT_SEL from 0 to 7 do                !Clear each units table
;          2461
;          2462          incr ARRAY_SEL from 0 to 15 do          !Clear each array within each unit
;          2463              PM_SBE_CNT [.UNIT_SEL, .ARRAY_SEL, PM_SBE$_SUM] = ZERO;     !Clear this array sum
;          2464
;          2465      return;
;          2466      end;
2073
2074                                        .SBTTL  CLRTBLS ONCE-ONLY CODE
2078 034554  004167  150534          CLRTBLS:JSR     R1,$SAVE4                ;                          2349
2079 034560  016704  145226                  MOV     L$UNIT,R4                ;                          2374
2080 034564  005001                          CLR     R1                       ; LUN
2081 034566  000424                          BR      3$
2082 034570  010100                  1$:     MOV     R1,R0                    ; LUN,*                    2379
2083 034572  006300                          ASL     R0
2084 034574  006300                          ASL     R0
2085 034576  006300                          ASL     R0
2086 034600  006300                          ASL     R0
2087 034602  005003                          CLR     R3                       ; ARRAY                    2377
2088 034604  010002                  2$:     MOV     R0,R2                    ;                          2379
2089 034606  060302                          ADD     R3,R2                    ; ARRAY,*
2090 034610  006302                          ASL     R2
2091 034612  005062  032714                  CLR     SOFTS(R2)
2092 034616  005062  033314                  CLR     HARDS(R2)                ;                          2380
2093 034622  005062  033714                  CLR     TRIES(R2)                ;                          2381
2094 034626  005203                          INC     R3                       ; ARRAY                    2377
2095 034630  020327  000017                  CMP     R3,#17                   ; ARRAY,*
2096 034634  003763                          BLE     2$
2097 034636  005201                          INC     R1                       ; LUN                      2374
2098 034640  020104                  3$:     CMP     R1,R4                    ; LUN,*
2099 034642  002752                          BLT     1$
2100 034644  005067  156014                  CLR     SBE$.COUNT               ;                          2386
2101 034650  005067  176030                  CLR     EOP.COUNT                ;                          2387
2102 034654  005067  177576                  CLR     NUM.DRIVES               ;                          2388
2103 034660  005067  177454                  CLR     RETRYING                 ;                          2389
2104 034664  005067  177424                  CLR     WR.COUNT                 ;                          2390
2105 034670  005067  177422                  CLR     WR.THOUSANDS             ;                          2391
2106 034674  005067  177420                  CLR     WR.MILLIONS              ;                          2392
```

```
                                        ;MLX4
                                        ;                ONCE-ONLY CODE                                  27-Mar-1982 19:24:42    TOPS
                                                                                                         27-Mar-1982 19:23:44    PA:<
034700  005067  177416                          CLR     RD.COUNT                ;                                               2393
034704  005067  177414                          CLR     RD.THOUSANDS            ;                                               2394
034710  005067  177412                          CLR     RD.MILLIONS             ;                                               2395
034714  005067  177410                          CLR     WC.COUNT                ;                                               2396
034720  005067  177406                          CLR     WC.THOUSANDS            ;                                               2397
034724  005067  177404                          CLR     WC.MILLIONS             ;                                               2398
034730  005002                                  CLR     R2                      ; LUN                                           2403
034732  000445                                  BR      5$
034734  010267  145134          4$:             MOV     R2,L$LUN                ; LUN,*                                         2405
034740  010201                                  MOV     R2,R1                   ; LUN,*                                         2406
034742  006301                                  ASL     R1
034744  005061  034460                          CLR     LOW.SECT(R1)
034750  010201                                  MOV     R2,R1                   ; LUN,*                                         2407
034752  006201                                  ASR     R1
034754  006201                                  ASR     R1
034756  006201                                  ASR     R1
034760  010146                                  MOV     R1,-(SP)
034762  062716  034442                          ADD     #DRIVE.STATUS,(SP)
034766  010246                                  MOV     R2,-(SP)                ; LUN,*
034770  042716  177770                          BIC     #177770,(SP)
034774  012746  000001                          MOV     #1,-(SP)
035000  011646                                  MOV     (SP),-(SP)
035002  004767  147606                          JSR     PC,BL$PU2
035006  010116                                  MOV     R1,(SP)                 ;                                               2408
035010  062716  034444                          ADD     #DROPT.DRIVES,(SP)
035014  010246                                  MOV     R2,-(SP)                ; LUN,*
035016  042716  177770                          BIC     #177770,(SP)
035022  012746  000001                          MOV     #1,-(SP)
035026  005046                                  CLR     -(SP)
035030  004767  147560                          JSR     PC,BL$PU2
035034  105062  034446                          CLRB    WHY.DROPT(R2)           ; *(LUN)                                        2409
035040  062706  000016                          ADD     #16.SP                  ;                                               2404
035044  005202                                  INC     R2                      ; LUN                                           2403
035046  020204          5$:             CMP     R2,R4                   ; LUN,*
035050  002731                          BLT     4$
035052  005767  145156                  TST     DROPNE                  ;                                                       2416
035056  001012                          BNE     6$
035060  005067  145152                  CLR     DROP1                   ;                                                       2419
035064  005067  145150                  CLR     DROP2                   ;                                                       2420
035070  005067  145146                  CLR     DROP3                   ;                                                       2421
035074  005067  145144                  CLR     DROP4                   ;                                                       2422
035100  005067  145144                  CLR     DROP5                   ;                                                       2423
035104  005003          6$:             CLR     R3                      ; INDEX                                                 2440
035106  010302          7$:             MOV     R3,R2                   ; INDEX,*                                               2442
035110  006302                          ASL     R2
035112  006302                          ASL     R2
035114  005062  011264                  CLR     SBE.LOG(R2)
035120  005062  011266                  CLR     SBE.LOG+2(R2)
035124  005203                          INC     R3                      ;                                                       2443
035126  020327  000177                  CMP     R3,#177                 ; INDEX                                                 2440
035132  003765                          BLE     7$                      ; INDEX,*
035134  005003                          CLR     R3                      ; UNIT.SEL                                              2460
```

```
                                ;MLX4                                                                   27-Mar-1982 19:24:42    TOPS
                                ;                       ONCE-ONLY CODE                                  27-Mar-1982 19:23:44    PA:<

035136  010300          8$:   MOV     R3,R0                   ; UNIT.SEL,*                                            2463
035140  006300                ASL     R0
035142  006300                ASL     R0
035144  006300                ASL     R0
035146  006300                ASL     R0
035150  005002                CLR     R2                      ; ARRAY.SEL                                             2462
035152  010001          9$:   MOV     R0,R1                   ;                                                       2463
035154  060201                ADD     R2,R1                   ; ARRAY.SEL,*
035156  006301                ASL     R1
035160  005061  012264        CLR     PM.SBE.CNT(R1)
035164  005202                INC     R2                      ; ARRAY.SEL                                             2462
035166  020227  000017        CMP     R2,#17                  ; ARRAY.SEL,*
035172  003767                BLE     9$
035174  005203                INC     R3                      ; UNIT.SEL                                              2460
035176  020327  000007        CMP     R3,#7                   ; UNIT.SEL,*
035202  003755                BLE     8$
035204  000207                RTS     PC                      ;                                                       2349

                                ; Routine Size:  141 words
                                ; Maximum stack depth per invocation:  12 words
```

```
;MLX4
;                    ONCE-ONLY CODE                                  27-Mar-1982 19:24:42    TOPS-20 Bliss-16 V2(212)
;                                                                    27-Mar-1982 19:23:44    PA:<NEALE>MLX4.BLI.5 (6)

;          2467  routine INIT_ADDRESSES (PLOC) : novalue =
;          2468      begin                                                 !* 1 *
;          2469
;          2470  !++
;          2471  !   ROUTINE:      INIT_ADDRESSES(PLOC)
;          2472  !
;          2473  !   PURPOSE:      THIS ROUTINE IS CALLED ONLY ONCE DURING THE INITIALIZATION
;          2474  !                 CODE, EVEN IF THERE IS MORE THAN ONE DRIVE PRESENT.  THE
;          2475  !                 PURPOSES OF THE ROUTINE ARE:
;          2476  !
;          2477  !                 (1)  TO OBTAIN HARDWARE P-TABLE INFORMATION FROM
;          2478  !                      MAIN MEMORY WHICH PERTAINS TO ALL DRIVES.
;          2479  !
;          2480  !                 (2)  TO SET UP THE ADDRESSES FOR THE 22 ML-11 REGISTERS.
;          2481  !
;          2482  !                 (3)  TO SET UP THE INTERRUPT SERVICE ROUTINE AT THE
;          2483  !                      CORRECT PRIORITY, AND LOWER THE CPU PRIORITY TO
;          2484  !                      ALLOW INTERRUPTS TO OCCUR.
;          2485  !
;          2486  !   ARGUMENT:     PLOC = THE POINTER TO THE LOCATION IN MAIN MEMORY WHERE
;          2487  !                        THE HARDWARE P-TABLE IS TO BE FOUND.
;          2488  !--
;          2489
;          2490      local
;          2491          TEMP,
;          2492          PRIORITY,
;          2493          OFFSET;
;          2494
;          2495      !+
;          2496      !THIS IS THE CODE FOR PURPOSE 1:
;          2497      !-
;          2498
;          2499      BASE_ADDR = .(.PLOC + 0);
;          2500      VEC = .(.PLOC + 2);
;          2501      BR_LEVEL = .(.PLOC + 4);
;          2502      PRIORITY = .BR_LEVEL^5;
;          2503
;          2504      if .ECCDIS
;          2505      then
;          2506          TEMP = WRD37                                    !ECC IS DISABLED
;          2507      else
;          2508          TEMP = WRD36;                                   !ECC IS ENABLED (NORMAL OPERATION)
;          2509
;          2510      PRINTB (CRLF);
;          2511      PRINTB (SAY5, WRD34, WRD40, WRD38, .TEMP, WRD41);
;          2512      !'RUNNING WITH ECC ENABLED/DISABLED AND'
;          2513
;          2514      if .REFRESH
;          2515      then
;          2516          TEMP = WRD36                                    !REFRESH MARGINING IS ENABLED
;          2517      else
;          2518          TEMP = WRD37;                                   !REFRESH MARGINING IS DISABLED (NORMAL OPERATION)
```

```
                                                                    27-Mar-1982 19:24:42    TOPS-20 Bliss-16 V2(212)
                        ONCE-ONLY CODE                              27-Mar-1982 19:23:44    PA:<NEALE>MLX4.BLI.5 (6)

                2519
                2520        PRINTB (SAY3, WRD40, PHR3, .TEMP);
                2521        !'WITH REFRESH MARGINING ENABLED/DISABLED'
                2522        PRINTB (CRLF);
                2523
                2524        !+
                2525        !THIS IS THE CODE FOR PURPOSE 2:
                2526        !-
                2527
                2528        OFFSET = 0;
                2529
                2530        incr COUNT from 0 to (NUM_REGS - 1) do
                2531            begin                                   !* 2 *
                2532            ML_REG [.COUNT] = .BASE_ADDR + .OFFSET;
                2533            OFFSET = .OFFSET + 2;
                2534            end;                                    !* 2 *
                2535
                2536        !+
                2537        !THIS IS THE CODE FOR PURPOSE 3:
                2538        !-
                2539
                2540        SETPRI (PRI00);
                2541        SETVEC (.VEC, SERVICE, .PRIORITY);
                2542        return;
                2543        end;                                        !* 1 *
2281
2282                                    .SBTTL  INIT.ADDRESSES ONCE-ONLY CODE
2286 035206                     INIT.ADDRESSES:
2287 035206  004167  150102              JSR     R1,$SAVE4               ;                       2467
2288 035212  016602  000014              MOV     14(SP),R2               ; PLOC,*                2499
2289 035216  011267  175464              MOV     (R2),BASE.ADDR
2290 035222  016267  000002  175460      MOV     2(R2),VEC               ;                       2500
2291 035230  016267  000004  175454      MOV     4(R2),BR.LEVEL          ;                       2501
2292 035236  016746  175450              MOV     BR.LEVEL,-(SP)          ;                       2502
2293 035242  012746  000005              MOV     #5,-(SP)
2294 035246  004767  147742              JSR     PC,BL$SHF
2295 035252  010003                      MOV     R0,R3                   ; *,PRIORITY
2296 035254  032767  000001  144772      BIT     #1,ECCDIS               ;                       2504
2297 035262  001403                      BEQ     1$
2298 035264  012701  007400              MOV     #WRD37,R1               ; *,TEMP                2506
2299 035270  000402                      BR      2$                      ;                       2504
2300 035272  012701  007370          1$: MOV     #WRD36,R1               ; *,TEMP                2508
2301 035276  012716  007066          2$: MOV     #CRLF,(SP)              ;                       2510
2302 035302  012746  000001              MOV     #1,-(SP)
2303 035306  010600                      MOV     SP,R0                   ; SP,*
2304 035310  104414                      TRAP    14
```

```
                                        ;MLX4
                                        ;                ONCE-ONLY CODE                                  27-Mar-1982 19:24:42    TOPS
                                                                                                         27-Mar-1982 19:23:44    PA:<
2309 035312  012716  007424                        MOV     #WRD41,(SP)                     ;                                       2511
2310 035316  010146                                MOV     R1,-(SP)                        ; TEMP,*
2311 035320  012746  007412                        MOV     #WRD38,-(SP)
2312 035324  012746  007416                        MOV     #WRD40,-(SP)
2313 035330  012746  007344                        MOV     #WRD34,-(SP)
2314 035334  012746  007152                        MOV     #SAY5,-(SP)
2315 035340  012746  000006                        MOV     #6,-(SP)
2316 035344  010600                                MOV     SP,R0                           ; SP,*
2317 035346  104414                                TRAP    14
2318 035350  032767  000001  144674                BIT     #1,REFRESH                      ;                                       2514
2319 035356  001403                                BEQ     3$
2320 035360  012701  007370                        MOV     #WRD36,R1                       ; *,TEMP                                2516
2321 035364  000402                                BR      4$                              ;                                       2514
2322 035366  012701  007400              3$:       MOV     #WRD37,R1                       ; *,TEMP                                2518
2323 035372  010116                      4$:       MOV     R1,(SP)                         ; TEMP,*                                2520
2324 035374  012746  007746                        MOV     #PHR3,-(SP)
2325 035400  012746  007416                        MOV     #WRD40,-(SP)
2326 035404  012746  007112                        MOV     #SAY3,-(SP)
2327 035410  012746  000004                        MOV     #4,-(SP)
2328 035414  010600                                MOV     SP,R0                           ; SP,*
2329 035416  104414                                TRAP    14
2330 035420  012716  007066                        MOV     #CRLF,(SP)                      ;                                       2522
2331 035424  012746  000001                        MOV     #1,-(SP)
2332 035430  010600                                MOV     SP,R0                           ; SP,*
2333 035432  104414                                TRAP    14
2334 035434  005002                                CLR     R2                              ; OFFSET                                2528
2335 035436  005000                                CLR     R0                              ; COUNT                                 2530
2336 035440  010001                      5$:       MOV     R0,R1                           ; COUNT,*                               2532
2337 035442  006301                                ASL     R1
2338 035444  016704  175236                        MOV     BASE.ADDR,R4
2339 035450  060204                                ADD     R2,R4                           ; OFFSET,*
2340 035452  010461  034346                        MOV     R4,ML.REG(R1)
2341 035456  062702  000002                        ADD     #2,R2                           ; *,OFFSET                              2533
2342 035462  005200                                INC     R0                              ; COUNT                                 2530
2343 035464  020027  000025                        CMP     R0,#25                          ; COUNT,*
2344 035470  003763                                BLE     5$
2345 035472  005000                                CLR     R0                              ;                                       2540
2346 035474  104441                                TRAP    41
2347 035476  010316                                MOV     R3,(SP)                         ; PRIORITY,*                            2541
2348 035500  012746  034544                        MOV     #SERVICE,-(SP)
2349 035504  016746  175200                        MOV     VEC,-(SP)
2350 035510  012746  000003                        MOV     #3,-(SP)
2351 035514  104437                                TRAP    37
2352 035516  062706  000042                        ADD     #42,SP                          ;                                       2467
2353 035522  000207                                RTS     PC
2354
2355                                    ; Routine Size:  103 words
2356                                    ; Maximum stack depth per invocation:  22 words
```

```
;MLX4                                                    27-Mar-1982 19:24:42    TOPS-20 Bliss-16 V2(212)
;                DRIVE IDENTIFICATION ROUTINES          27-Mar-1982 19:23:44    PA:<NEALE>MLX4.BLI.5 (7)

;          2544  %sbttl 'DRIVE IDENTIFICATION ROUTINES'
;          2545  routine SAYWHO (LUN) : novalue =
;          2546      begin
;          2547
;          2548  !++
;          2549  !  ROUTINE:     SAYWHO(LUN)
;          2550  !
;          2551  !  PURPOSE:     TO PRINT OUT AN IDENTIFICATION LINE WHICH INCLUDES:
;          2552  !                  LOGICAL UNIT: X    DRIVE: Y    SERIAL #: ZZZZ
;          2553  !
;          2554  !               THE UNIT'S SERIAL NUMBER IN EITHER BCD OR OCTAL FORMAT.
;          2555  !               BCD WILL BE PRINTED AS LONG AS THE DIGITS ARE ALL VALID.
;          2556  !--
;          2557
;          2558      local
;          2559          D3,
;          2560          D2,
;          2561          D1,
;          2562          D0;
;          2563
;          2564      D3 = .SN3;
;          2565      D2 = .SN2;
;          2566      D1 = .SN1;
;          2567      D0 = .SN0;
;          2568      PRINTB (FMT4A, PHR7, .LUN, WRD11, .DRIVE);
;          2569      !'LOGICAL UNIT: X    DRIVE: Y'
;          2570
;          2571      if ((.D3 gtr 9) or (.D2 gtr 9) or (.D1 gtr 9) or (.D0 gtr 9))
;          2572      then
;          2573          PRINTB (FMT4B, PHR8, .MLSN)
;          2574          !'    SERIAL #: ZZZZZZ'
;          2575      else
;          2576          PRINTB (FMT4C, PHR8, .D3, .D2, .D1, .D0);
;          2577
;          2578      !'    SERIAL #: DDDD'
;          2579      return;
;          2580      end;
;
;                                .SBTTL  SAYWHO DRIVE IDENTIFICATION ROUTINES
2410 035524 004167 147604  SAYWHO: JSR     R1,$SAVE5                    ;
2411 035530 017705 176642          MOV     @#L.REG+30,R5                ; *,D3                    2545
2412 035534 006205                 ASR     R5                           ; D3                      2564
2413 035536 006205                 ASR     R5                           ; D3
2414 035540 006205                 ASR     R5                           ; D3
2415 035542 006205                 ASR     R5                           ; D3
2416 035544 000305                 SWAB    R5                           ; D3
2417                       ;MLX4                                                 27-Mar-1982 19:24:42    TOPS
2418                       ;            DRIVE IDENTIFICATION ROUTINES            27-Mar-1982 19:23:44    PA:<
2419
2420 035546 042705 177760          BIC     #177760,R5                   ; *,D3
2421 035552 017704 176620          MOV     @#L.REG+30,R4                ; *,D2                    2565
2422 035556 000304                 SWAB    R4                           ; D2
2423 035560 042704 177760          BIC     #177760,R4                   ; *,D2
2424 035564 117701 176606          MOVB    @#L.REG+30,R1                ; *,D1                    2566
```

```
2425 035570  006201                    ASR     R1                      ; D1
2426 035572  006201                    ASR     R1                      ; D1
2427 035574  006201                    ASR     R1                      ; D1
2428 035576  006201                    ASR     R1                      ; D1
2429 035600  042701  177760            BIC     #177760,R1              ; *,D1
2430 035604  117702  176566            MOVB    @ML.REG+30,R2           ; *,D0          2567
2431 035610  042702  177760            BIC     #177760,R2              ; *,D0
2432 035614  016603  000016            MOV     16(SP),R3               ; LUN,*         2568
2433 035620  006303                    ASL     R3
2434 035622  016303  034422            MOV     PTABLE.ADDR(R3),R3
2435 035626  016346  000006            MOV     6(R3),-(SP)
2436 035632  012746  007240            MOV     #WRD11,-(SP)
2437 035636  016646  000022            MOV     22(SP),-(SP)            ; LUN,*
2438 035642  012746  010050            MOV     #PHR7,-(SP)
2439 035646  012746  006216            MOV     #FMT4A,-(SP)
2440 035652  012746  000005            MOV     #5,-(SP)
2441 035656  010600                    MOV     SP,R0                   ; SP,*
2442 035660  104414                    TRAP    14
2443 035662  020527  000011            CMP     R5,#11                  ; D3,*          2571
2444 035666  003011                    BGT     1$
2445 035670  020427  000011            CMP     R4,#11                  ; D2,*
2446 035674  003006                    BGT     1$
2447 035676  020127  000011            CMP     R1,#11                  ; D1,*
2448 035702  003003                    BGT     1$
2449 035704  020227  000011            CMP     R2,#11                  ; D0,*
2450 035710  003413                    BLE     2$
2451 035712  017746  176460    1$:     MOV     @ML.REG+30,-(SP)        ;              2573
2452 035716  012746  010066            MOV     #PHR8,-(SP)
2453 035722  012746  006252            MOV     #FMT4B,-(SP)
2454 035726  012746  000003            MOV     #3,-(SP)
2455 035732  010600                    MOV     SP,R0                   ; SP,*
2456 035734  104414                    TRAP    14
2457 035736  000416                    BR      3$                      ;              2571
2458 035740  010246    2$:     MOV     R2,-(SP)                ; D0,*          2576
2459 035742  010146                    MOV     R1,-(SP)                ; D1,*
2460 035744  010446                    MOV     R4,-(SP)                ; D2,*
2461 035746  010546                    MOV     R5,-(SP)                ; D3,*
2462 035750  012746  010066            MOV     #PHR8,-(SP)
2463 035754  012746  006270            MOV     #FMT4C,-(SP)
2464 035760  012746  000006            MOV     #6,-(SP)
2465 035764  010600                    MOV     SP,R0                   ; SP,*
2466 035766  104414                    TRAP    14
2467 035770  062706  000006            ADD     #6,SP
2468 035774  062706  000024    3$:     ADD     #24,SP                  ;              2545
2469 036000  000207                    RTS     PC
2470
2471                                   ; Routine Size:  87 words
2472                                   ;MLX4
2473                                   ;               DRIVE IDENTIFICATION ROUTINES   27-Mar-1982 19:24:42    TOPS
2474                                                                                   27-Mar-1982 19:23:44    PA:<
2475                                   ; Maximum stack depth per invocation:  19 words
2480
2481
2482 ;MLX4                                                  27-Mar-1982 19:24:42    TOPS-20 Bliss-16 V2(212)
2483 ;                 DRIVE IDENTIFICATION ROUTINES       27-Mar-1982 19:23:44    PA:<NEALE>MLX4.BLI.5 (8)
2484
2485 ;          2581  routine CONFIG (LUN) : novalue =
```

```
2486 ;          2582      begin                                   !* 1 *
2487 ;          2583
2488 ;          2584  !++
2489 ;          2585  !  ROUTINE:     CONFIG(LUN)
2490 ;          2586  !
2491 ;          2587  !  PURPOSE:     CONFIG IS CALLED BY THE INITIALIZATION CODE FOR EACH DRIVE
2492 ;          2588  !               WHICH SUCCESSFULLY RESPONDS TO A REQUEST FOR ITS HARDWARE
2493 ;          2589  !               P-TABLE.  THE PURPOSES OF THIS ROUTINE ARE:
2494 ;          2590  !
2495 ;          2591  !               (1)  TO CHECK THAT THE DRIVE IS POWERED UP.
2496 ;          2592  !
2497 ;          2593  !               (2)  TO VERIFY THAT THE DRIVE IS AN ML11 UNIT.
2498 ;          2594  !
2499 ;          2595  !               (3)  TO VERIFY THAT THE OPERATOR DEFINED TEST RANGES ARE
2500 ;          2596  !                    WITHIN THE CALCULATED SYSTEM SIZE.
2501 ;          2597  !
2502 ;          2598  !               (4)  TO PRINT DRIVE IDENTIFICATION INFORMATION.
2503 ;          2599  !
2504 ;          2600  !  ARGUMENT:    LUN = THE LOGICAL UNIT NUMBER, COUNTING FROM 0 TO
2505 ;          2601  !                     NUMBER OF DRIVES MINUS 1.
2506 ;          2602  !--
2507 ;          2603
2508 ;          2604      local
2509 ;          2605          TYPE,
2510 ;          2606          TEMP,
2511 ;          2607          TOP;
2512 ;          2608
2513 ;          2609      !+
2514 ;          2610      !THIS IS THE CODE FOR PURPOSE 1:
2515 ;          2611      !-
2516 ;          2612
2517 ;          2613      UNIT = .DRIVE;
2518 ;          2614
2519 ;          2615      if .DPR neq 1
2520 ;          2616      then
2521 ;          2617          begin
2522 ;          2618          PRINTB (FMT4A, PHR7, .LUN, WRD11, .DRIVE);
2523 ;          2619          !'LOGICAL UNIT: X    DRIVE: Y'
2524 ;          2620          PRINTB (SAY2, WRD11, WRD21);
2525 ;          2621          !'DRIVE DROPPED'
2526 ;          2622          PRINTB (FMT2, CAUSE1);
2527 ;          2623          !'  (NOT POWERED UP)'
2528 ;          2624          WHY_DROPT [.LUN] = CODE_1;
2529 ;          2625          DODU (.LUN);
2530 ;          2626          return;
2531 ;          2627          end;
2532 ;          2628
2533 ;          2629      !+
2534 ;          2630      !THIS IS THE CODE FOR PURPOSE 2:
2535 ;          2631      !-
2536 ;          2632
```

```
2538 ;MLX4
2539 ;                  DRIVE IDENTIFICATION ROUTINES                    27-Mar-1982 19:24:42    TOPS-20 Bliss-16 V2(212)
2540 ;                                                                   27-Mar-1982 19:23:44    PA:<NEALE>MLX4.BLI.5 (8)
2541 ;          2633        SAYWHO (.LUN);
2542 ;          2634        !'LOGICAL UNIT: X    DRIVE: Y    SERIAL #: ZZZZ'
2543 ;          2635        TYPE = .MLDT;
2544 ;          2636
2545 ;          2637        if ((.TYPE neq ML11A) and (.TYPE neq ML11B))
2546 ;          2638        then
2547 ;          2639            begin
2548 ;          2640            PRINTB (SAY2, WRD11, WRD21);
2549 ;          2641            !'DRIVE DROPPED'
2550 ;          2642            PRINTB (FMT2, CAUSE2);
2551 ;          2643            !'  (NOT AN ML11 UNIT)'
2552 ;          2644            WHY_DROPT [.LUN] = CODE_2;
2553 ;          2645            DODO (.LUN);
2554 ;          2646            return;
2555 ;          2647            end
2556 ;          2648        else
2557 ;          2649            begin                                    !* 3 *
2558 ;          2650
2559 ;          2651            if .ARR_TYP eql 0
2560 ;          2652            then
2561 ;          2653                begin                                !* 4 *
2562 ;          2654                TYPE = WRD6;
2563 ;          2655                TOP = 1;
2564 ;          2656                end                                  !* 4 *
2565 ;          2657            else
2566 ;          2658                begin                                !* 5 *
2567 ;          2659                TYPE = WRD7;
2568 ;          2660                TOP = 4;
2569 ;          2661                end;                                 !* 5 *
2570 ;          2662
2571 ;          2663            !+
2572 ;          2664            !THIS IS THE CODE FOR PURPOSE 3:
2573 ;          2665            !-
2574 ;          2666
2575 ;          2667            TOP_SECT [.LUN] = (.TOP)*(512)*(.SZ) - 1;
2576 ;          2668
2577 ;          2669            if .LIMIT
2578 ;          2670            then
2579 ;          2671
2580 ;          2672            !+
2581 ;          2673            ! THE OPERATOR HAS CHOSEN LIMITS:
2582 ;          2674            !-
2583 ;          2675
2584 ;          2676                begin                                !* 6 *
2585 ;          2677
2586 ;          2678                if .RANGE eql 0
2587 ;          2679                then
2588 ;          2680
2589 ;          2681                !+
2590 ;          2682                !THE BOARD NUMBER (0-15) IS CONTAINED IN 'ONLY'
2591 ;          2683                !-
2592 ;          2684
```

```
;MLX4
;                DRIVE IDENTIFICATION ROUTINES                  27-Mar-1982 19:24:42    TOPS-20 Bliss-16 V2(212)
;                                                               27-Mar-1982 19:23:44    PA:<NEALE>MLX4.BLI.5 (8)
;          2685                  begin                          !* 10 *
;          2686  !
;          2687  ! VER CZMLBB CHANGED .TYPE TO .ARR_TYP
;          2688  !
;          2689
;          2690                  if .ARR_TYP
;          2691                  then
;          2692
;          2693                  !+
;          2694                  !THE CHIPS ARE 64K
;          2695                  !-
;          2696
;          2697                      begin                      !* 7 *
;          2698                      LSECT = 2048*.ONLY;
;          2699                      TSECT = .LSECT + 2047;
;          2700                      end                        !* 7 *
;          2701                  else
;          2702
;          2703                  !+
;          2704                  !THE CHIPS ARE 16K
;          2705                  !-
;          2706
;          2707                      begin                      !* 8 *
;          2708                      LSECT = 512*.ONLY;
;          2709                      TSECT = .LSECT + 511;
;          2710                      end;                       !* 8 *
;          2711
;          2712                  end;                           !* 10 *
;          2713
;          2714                  if .TSECT leqa .TOP_SECT [.LUN]
;          2715                  then
;          2716                      TOP_SECT [.LUN] = .TSECT
;          2717                  else
;          2718                      begin
;          2719                      PRINTB (SAY2, WRD11, WRD21);
;          2720                      !'DRIVE DROPPED'
;          2721                      PRINTB (FMT2, CAUSE3);
;          2722                      !'  (OPERATOR SELECTED TEST LIMITS INCORRECTLY)'
;          2723                      PRINTB (FMT11, PHR14, .TSECT, PHR16, .TOP_SECT [.LUN]);
;          2724                      !'TOP SECTOR OF XXXXXX EXCEEDS SYSTEM LIMIT OF YYYYYY'
;          2725                      WHY_DROPT [.LUN] = CODE_3;
;          2726                      DODO (.LUN);
;          2727                      return;
;          2728                      end;
;          2729
;          2730                  if .LSECT leqa .TSECT
;          2731                  then
;          2732                      LOW_SECT [.LUN] = .LSECT
;          2733                  else
;          2734                      begin
;          2735                      PRINTB (SAY2, WRD11, WRD21);
;          2736                      !'DRIVE DROPPED'
```

```
2650 ;MLX4
2651 ;                  DRIVE IDENTIFICATION ROUTINES                    27-Mar-1982 19:24:42    TOPS-20 Bliss-16 V2(212)
2652 ;                                                                   27-Mar-1982 19:23:44    PA:<NEALE>MLX4.BLI.5 (8)
2653 ;          2737                    PRINTB (FMT2, CAUSE3);
2654 ;          2738                    !'  (OPERATOR SELECTED TEST LIMITS INCORRECTLY)'
2655 ;          2739                    PRINTB (FMT11, PHR15, .LSECT, PHR14, .TSECT);
2656 ;          2740                    !'LOW SECTOR OF XXXXXX EXCEEDS TOP SECTOR OF YYYYYY'
2657 ;          2741                    WHY_DROPT [.LUN] = CODE_3;
2658 ;          2742                    DODO (.LUN);
2659 ;          2743                    return;
2660 ;          2744                    end;
2661 ;          2745
2662 ;          2746                end;                                 !* 6 *
2663 ;          2747
2664 ;          2748            !+
2665 ;          2749            !THIS IS THE CODE FOR PURPOSE 4:
2666 ;          2750            !-
2667 ;          2751
2668 ;          2752            PRINTB (FMT5, .TYPE, LOWEST, HIGHEST);
2669 ;          2753            !'ML11-X     SECTORS UNDER TEST: XXXXXX TO YYYYYY'
2670 ;          2754
2671 ;          2755            selectone .TRT of
2672 ;          2756                set
2673 ;          2757
2674 ;          2758                [0] :
2675 ;          2759                    TEMP = TRT00;
2676 ;          2760
2677 ;          2761                [1] :
2678 ;          2762                    TEMP = TRT01;
2679 ;          2763
2680 ;          2764                [2] :
2681 ;          2765                    TEMP = TRT10;
2682 ;          2766
2683 ;          2767                [3] :
2684 ;          2768                    TEMP = TRT11;
2685 ;          2769                tes;
2686 ;          2770
2687 ;          2771            PRINTB (FMT8, .TEMP);
2688 ;          2772            !'TRANSFER RATE: X MBYTES/SECOND'
2689 ;          2773            PRINTB (FMT4B, PHR4, .BASE_ADDR);
2690 ;          2774            !'     CSR ADDRESS: XXXXXX'
2691 ;          2775            end;                                     !* 3 *
2692 ;          2776
2693 ;          2777        return;
2694 ;          2778        end;                                         !* 1 *
2698
2699
2703 036002  004167  147326          CONFIG: JSR     R1,$SAVE5                       ;                       2581
2704 036006  016603  000016                  MOV     16(SP),R3                       ; LUN,*                 2613
```

```
                                        ;MLX4
                                        ;               DRIVE IDENTIFICATION ROUTINES                    27-Mar-1982 19:24:42    TOPS
                                                                                                         27-Mar-1982 19:23:44    PA:<
036012  010304                                MOV     R3,R4
036014  006304                                ASL     R4
036016  016401  034422                        MOV     PTABLE.ADDR(R4),R1
036022  016105  000006                        MOV     6(R1),R5
036026  042705  177770                        BIC     #177770,R5
036032  142777  000007  176316                BICB    #7,@ML.REG+10
036040  150577  176312                        BISB    R5,@ML.REG+10
036044  032777  000400  176306                BIT     #400,@ML.REG+12                 ;                2615
036052  001051                                BNE     1$
036054  016401  034422                        MOV     PTABLE.ADDR(R4),R1              ;                2618
036060  016146  000006                        MOV     6(R1),-(SP)
036064  012746  007240                        MOV     #WRD11,-(SP)
036070  010346                                MOV     R3,-(SP)
036072  012746  010050                        MOV     #PHR7,-(SP)
036076  012746  006216                        MOV     #FMT4A,-(SP)
036102  012746  000005                        MOV     #5,-(SP)
036106  010600                                MOV     SP,R0                           ; SP,*
036110  104414                                TRAP    14
036112  012716  007322                        MOV     #WRD21,(SP)                     ;                2620
036116  012746  007240                        MOV     #WRD11,-(SP)
036122  012746  007100                        MOV     #SAY2,-(SP)
036126  012746  000003                        MOV     #3,-(SP)
036132  010600                                MOV     SP,R0                           ; SP,*
036134  104414                                TRAP    14
036136  012716  010460                        MOV     #CAUSE1,(SP)                    ;                2622
036142  012746  006160                        MOV     #FMT2,-(SP)
036146  012746  000002                        MOV     #2,-(SP)
036152  010600                                MOV     SP,R0                           ; SP,*
036154  104414                                TRAP    14
036156  112763  000001  034446                MOVB    #1,WHY.DROPT(R3)                ;                2624
036164  010300                                MOV     R3,R0                           ;                2625
036166  104451                                TRAP    51
036170  062706  000026                        ADD     #26,SP                          ;                2615
036174  000207                                RTS     PC                              ;                2617
036176  010346                        1$:     MOV     R3,-(SP)                        ;                2633
036200  004767  177320                        JSR     PC,SAYWHO
036204  017702  176164                        MOV     @ML.REG+26,R2                   ; *,TYPE         2635
036210  020227  000110                        CMP     R2,#110                         ; TYPE,*         2637
036214  001435                                BEQ     2$
036216  020227  000111                        CMP     R2,#111                         ; TYPE,*
036222  001432                                BEQ     2$
036224  012746  007322                        MOV     #WRD21,-(SP)                    ;                2640
036230  012746  007240                        MOV     #WRD11,-(SP)
036234  012746  007100                        MOV     #SAY2,-(SP)
036240  012746  000003                        MOV     #3,-(SP)
036244  010600                                MOV     SP,R0                           ; SP,*
036246  104414                                TRAP    14
036250  012716  010502                        MOV     #CAUSE2,(SP)                    ;                2642
036254  012746  006160                        MOV     #FMT2,-(SP)
036260  012746  000002                        MOV     #2,-(SP)
036264  010600                                MOV     SP,R0                           ; SP,*
036266  104414                                TRAP    14
```

```
                                        ;MLX4
                                        ;               DRIVE IDENTIFICATION ROUTINES                    27-Mar-1982 19:24:42   TOPS
                                                                                                         27-Mar-1982 19:23:44   PA:<
036270  112763  000002  034446          MOVB    #2,WHY.DROPT(R3)        ;                                                       2644
036276  010300                          MOV     R3,R0                   ;                                                       2645
036300  104451                          TRAP    51
036302  062706  000016                  ADD     #16,SP                  ;                                                       2637
036306  000207                          RTS     PC                      ;                                                       2639
036310  032777  002000  176054  2$:     BIT     #2000,@ML.REG+24        ;                                                       2651
036316  001005                          BNE     3$
036320  012702  007220                  MOV     #WRD6,R2                ; *,TYPE                                                2654
036324  012701  000001                  MOV     #1,R1                   ; *,TOP                                                 2655
036330  000404                          BR      4$                      ;                                                       2651
036332  012702  007230          3$:     MOV     #WRD7,R2                ; *,TYPE                                                2659
036336  012701  000004                  MOV     #4,R1                   ; *,TOP                                                 2660
036342  012705  034500          4$:     MOV     #TOP.SECT,R5            ;                                                       2667
036346  060405                          ADD     R4,R5
036350  010146                          MOV     R1,-(SP)                ; TOP,*                                                 2649
036352  017701  176014                  MOV     @ML.REG+24,R1
036356  006201                          ASR     R1
036360  006201                          ASR     R1
036362  006201                          ASR     R1
036364  000301                          SWAB    R1
036366  042701  177740                  BIC     #177740,R1
036372  010146                          MOV     R1,-(SP)
036374  004767  146344                  JSR     PC,BL$MUL
036400  000300                          SWAB    R0                      ;                                                       2667
036402  105000                          CLRB    R0
036404  006300                          ASL     R0
036406  010001                          MOV     R0,R1
036410  005301                          DEC     R1
036412  010115                          MOV     R1,(R5)
036414  032767  000001  143600          BIT     #1,LIMIT                ;                                                       2669
036422  001575                          BEQ     11$
036424  005767  143574                  TST     RANGE                   ;                                                       2678
036430  001037                          BNE     6$
036432  032777  002000  175732          BIT     #2000,@ML.REG+24        ;                                                       2690
036440  001417                          BEQ     5$
036442  016700  143564                  MOV     ONLY,R0                 ;                                                       2698
036446  000300                          SWAB    R0
036450  105000                          CLRB    R0
036452  006300                          ASL     R0
036454  006300                          ASL     R0
036456  006300                          ASL     R0
036460  010067  143542                  MOV     R0,LSECT
036464  010067  143540                  MOV     R0,TSECT                ; LSECT,*
036470  062767  003777  143532          ADD     #3777,TSECT                                                                     2699
036476  000414                          BR      6$                      ;                                                       2690
036500  016700  143526          5$:     MOV     ONLY,R0                 ;                                                       2708
036504  000300                          SWAB    R0
036506  105000                          CLRB    R0
036510  006300                          ASL     R0
036512  010067  143510                  MOV     R0,LSECT
036516  010067  143506                  MOV     R0,TSECT                ; LSECT,*
036522  062767  000777  143500          ADD     #777,TSECT                                                                      2709
```

```
                                        ;MLX4
                                        ;                DRIVE IDENTIFICATION ROUTINES                    27-Mar-1982 19:24:42   TOPS
                                                                                                         27-Mar-1982 19:23:44   PA:<
036530  026715  143474        6$:     CMP     TSECT,(R5)                      ;                                               2714
036534  101003                        BHI     7$
036536  016715  143466                MOV     TSECT,(R5)                      ;                                               2716
036542  000445                        BR      8$                              ;                                               2714
036544  012746  007322        7$:     MOV     #WRD21,-(SP)                    ;                                               2719
036550  012746  007240                MOV     #WRD11,-(SP)
036554  012746  007100                MOV     #SAY2,-(SP)
036560  012746  000003                MOV     #3,-(SP)
036564  010600                        MOV     SP,R0                           ; SP,*
036566  104414                        TRAP    14
036570  012716  010526                MOV     #CAUSE3,(SP)                    ;                                               2721
036574  012746  006160                MOV     #FMT2,-(SP)
036600  012746  000002                MOV     #2,-(SP)
036604  010600                        MOV     SP,R0                           ; SP,*
036606  104414                        TRAP    14
036610  011516                        MOV     (R5),(SP)                       ;                                               2723
036612  012746  010322                MOV     #PHR16,-(SP)
036616  016746  143406                MOV     TSECT,-(SP)
036622  012746  010266                MOV     #PHR14,-(SP)
036626  012746  006654                MOV     #FMT11,-(SP)
036632  012746  000005                MOV     #5,-(SP)
036636  010600                        MOV     SP,R0                           ; SP,*
036640  104414                        TRAP    14
036642  112763  000003  034446        MOVB    #3,WHY.DROPT(R3)                ;                                               2725
036650  010300                        MOV     R3,R0                           ;                                               2726
036652  104451                        TRAP    51
036654  000455                        BR      10$                             ;                                               2714
036656  026767  143344  143344  8$:   CMP     LSECT,TSECT                     ;                                               2730
036664  101004                        BHI     9$
036666  016764  143334  034460        MOV     LSECT,LOW.SECT(R4)              ;                                               2732
036674  000450                        BR      11$                             ;                                               2730
036676  012746  007322        9$:     MOV     #WRD21,-(SP)                    ;                                               2735
036702  012746  007240                MOV     #WRD11,-(SP)
036706  012746  007100                MOV     #SAY2,-(SP)
036712  012746  000003                MOV     #3,-(SP)
036716  010600                        MOV     SP,R0                           ; SP,*
036720  104414                        TRAP    14
036722  012716  010526                MOV     #CAUSE3,(SP)                    ;                                               2737
036726  012746  006160                MOV     #FMT2,-(SP)
036732  012746  000002                MOV     #2,-(SP)
036736  010600                        MOV     SP,R0                           ; SP,*
036740  104414                        TRAP    14
036742  016716  143262                MOV     TSECT,(SP)                      ;                                               2739
036746  012746  010266                MOV     #PHR14,-(SP)
036752  016746  143250                MOV     LSECT,-(SP)
036756  012746  010304                MOV     #PHR15,-(SP)
036762  012746  006654                MOV     #FMT11,-(SP)
036766  012746  000005                MOV     #5,-(SP)
036772  010600                        MOV     SP,R0                           ; SP,*
036774  104414                        TRAP    14
036776  112763  000003  034446        MOVB    #3,WHY.DROPT(R3)                ;                                               2741
037004  010300                        MOV     R3,R0                           ;                                               2742
```

```
                                ;MLX4
                                ;                DRIVE IDENTIFICATION ROUTINES                    27-Mar-1982 19:24:42   TOPS
                                ;                                                                 27-Mar-1982 19:23:44   PA:<
037006  104451                          TRAP    51
037010  062706  000034        10$:    ADD     #34,SP                  ;                                                2730
037014  000207                          RTS     PC                      ;                                                2734
037016  011516                11$:    MOV     (R5),(SP)               ;                                                2752
037020  016446  034460                MOV     LOW.SECT(R4),-(SP)
037024  010246                        MOV     R2,-(SP)                ; TYPE,*
037026  012746  006320                MOV     #FMT5,-(SP)
037032  012746  000004                MOV     #4,-(SP)
037036  010600                        MOV     SP,R0                   ; SP,*
037040  104414                        TRAP    14
037042  017701  175324                MOV     @ML.REG+24,R1           ;                                                2755
037046  000301                        SWAB    R1
037050  042701  177774                BIC     #177774,R1
037054  001003                        BNE     12$
037056  012700  010444                MOV     #TRT00,R0               ; *,TEMP                                         2759
037062  000421                        BR      15$                     ;                                                2755
037064  020127  000001        12$:    CMP     R1,#1
037070  001003                        BNE     13$
037072  012700  010446                MOV     #TRT01,R0               ; *,TEMP                                         2762
037076  000413                        BR      15$                     ;                                                2755
037100  020127  000002        13$:    CMP     R1,#2
037104  001003                        BNE     14$
037106  012700  010450                MOV     #TRT10,R0               ; *,TEMP                                         2765
037112  000405                        BR      15$                     ;                                                2755
037114  020127  000003        14$:    CMP     R1,#3
037120  001002                        BNE     15$
037122  012700  010454                MOV     #TRT11,R0               ; *,TEMP                                         2768
037126  010016                15$:    MOV     R0,(SP)                 ; TEMP,*                                         2771
037130  012746  006466                MOV     #FMT8,-(SP)
037134  012746  000002                MOV     #2,-(SP)
037140  010600                        MOV     SP,R0                   ; SP,*
037142  104414                        TRAP    14
037144  016716  173536                MOV     BASE.ADDR,(SP)          ;                                                2773
037150  012746  007770                MOV     #PHR4,-(SP)
037154  012746  006252                MOV     #FMT48,-(SP)
037160  012746  000003                MOV     #3,-(SP)
037164  010600                        MOV     SP,R0                   ; SP,*
037166  104414                        TRAP    14
037170  062706  000030                ADD     #30,SP                  ;                                                2581
037174  000207                        RTS     PC

                        ; Routine Size:  318 words
                        ; Maximum stack depth per invocation:  20 words
```

```
;MLX4
;                    DRIVE IDENTIFICATION ROUTINES                    27-Mar-1982 19:24:42    TOPS-20 Bliss-16 V2(212)
;                                                                     27-Mar-1982 19:23:44    PA:<NEALE>MLX4.BLI.5 (9)

;          2779  routine DECODE (SECT) : novalue =
;          2780      begin
;          2781
;          2782  !++
;          2783  !   ROUTINE:      DECODE(SECT)
;          2784  !
;          2785  !   PURPOSE:      TO INTERPRET THE FIELDS OF THE FAILING SECTOR
;          2786  !                 FOR BOTH 16K AND 64K CHIPS.
;          2787  !
;          2788  !   ARGUMENT:     SECT = FAILING SECTOR NUMBER
;          2789  !
;          2790  !   RESULTS:      (1)  BANK = WHICH ROW OF CHIPS (0-4)
;          2791  !
;          2792  !                 (2)  BOARD = WHICH ARRAY MODULE (0-15)
;          2793  !--
;          2794
;          2795      if .MLDT
;          2796      then                                              !THE CHIPS ARE 64K
;          2797          begin
;          2798          BANK = .SECT<9, 2>;
;          2799          BOARD = .SECT<11, 4>;
;          2800          end
;          2801      else                                              !THE CHIPS ARE 16K
;          2802          begin
;          2803          BANK = .SECT<7, 2>;
;          2804          BOARD = .SECT<9, 4>;
;          2805          end;
;          2806
;          2807  !
;          2808  ! VER CZMLBB ADDED LOADING OF BIT_NUM TO ROUTINE
;          2809  !
;          2810      BIT_NUM = .CHAN;                                  !SAVE THE FAILING CHIP NUMBER
;          2811      return;
;          2812      end;
;
;                                                .SBTTL  DECODE DRIVE IDENTIFICATION ROUTINES
;2972 037176  032777  000001  175170  DECODE:  BIT     #1,@ML.REG+26                    ;                    2795
;2973 037204  001415                          BEQ     1$
;2974 037206  016600  000002                  MOV     2(SP),R0                         ; SECT,*             2798
;2975 037212  006200                          ASR     R0
;2976 037214  000300                          SWAB    R0
;2977 037216  042700  177774                  BIC     #177774,R0
;2978 037222  010067  175116                  MOV     R0,BANK
;2979 037226  016600  000002                  MOV     2(SP),R0                         ; SECT,*             2799
;2980 037232  006200                          ASR     R0
;2981 037234  006200                          ASR     R0
```

```
                                        ;MLX4                                                                   27-Mar-1982 19:24:42    TOPS
                                        ;               DRIVE IDENTIFICATION ROUTINES                           27-Mar-1982 19:23:44    PA:<

037236  000412                                  BR      2$
037240  016600  000002          1$:     MOV     2(SP),R0                        ; SECT,*                                        2803
037244  006100                          ROL     R0
037246  000300                          SWAB    R0
037250  042700  177774                  BIC     #177774,R0
037254  010067  175064                  MOV     R0,BANK
037260  016600  000002                  MOV     2(SP),R0                        ; SECT,*                                        2804
037264  006200                  2$:     ASR     R0
037266  000300                          SWAB    R0
037270  042700  177760                  BIC     #177760,R0
037274  010067  175042                  MOV     R0,BOARD
037300  017700  175104                  MOV     @ML.REG+42,R0                   ;                                               2810
037304  006200                          ASR     R0
037306  006200                          ASR     R0
037310  006200                          ASR     R0
037312  006200                          ASR     R0
037314  006200                          ASR     R0
037316  006200                          ASR     R0
037320  042700  177700                  BIC     #177700,R0
037324  010067  153336                  MOV     R0,BIT.NUM
037330  000207                          RTS     PC                              ;                                               2779

                                        ; Routine Size:  46 words
                                        ; Maximum stack depth per invocation:  0 wo ds
```

```
;MLX4
;                    DRIVE IDENTIFICATION ROUTINES                    27-Mar-1982 19:24:42    TOPS-20 Bliss-16 V2(212)
;                                                                     27-Mar-1982 19:23:44    PA:<NEALE>MLX4.BLI.5 (10)

;          2813  routine ISOLATE : novalue =
;          2814      begin
;          2815
;          2816  !++
;          2817  !  ROUTINE:      ISOLATE
;          2818  !
;          2819  !  PURPOSE:      UPON THE DETECTION OF A DATA ERROR (EITHER RECOVERABLE OR
;          2820  !                NOT) TO EXAMINE THE SECTOR ADDRESS CONTAINED IN THE ECC
;          2821  !                ERROR LOCATION REGISTER AND PINPOINT THE LOCATION OF THE
;          2822  !                ERROR.  THE SECTOR, BOARD (0-15) AND BANK (0-3) WILL BE
;          2823  !                REPORTED.
;          2824  !--
;          2825
;          2826      DECODE (.MLEL);
;          2827
;          2828      if .ERROUT then PRINTB (FMT10A, WRD20, WRD15, .MLEL, .BOARD, .BANK);
;          2829
;          2830      !'FAILED:  SECTOR XXXXXX  BOARD YY  BANK Z'
;          2831      return;
;          2832      end;
;
                                        .SBTTL  ISOLATE DRIVE IDENTIFICATION ROUTINES
037332  017746  175054          ISOLATE:MOV     @ML.REG+44,-(SP)                ;                       2826
037336  004767  177634                  JSR     PC,DECODE
037342  032767  000001  142710          BIT     #1,ERROUT                       ;                       2828
037350  001422                          BEQ     1$
037352  016746  174766                  MOV     BANK,-(SP)
037356  016746  174760                  MOV     BOARD,-(SP)
037362  017746  175024                  MOV     @ML.REG+44,-(SP)
037366  012746  007246                  MOV     #WRD15,-(SP)
037372  012746  007312                  MOV     #WRD20,-(SP)
037376  012746  006564                  MOV     #FMT10A,-(SP)
037402  012746  000006                  MOV     #6,-(SP)
037406  010600                          MOV     SP,R0                           ; SP,*
037410  104414                          TRAP    14
037412  062706  000016                  ADD     #16,SP
037416  005726                  1$:     TST     (SP)+                           ;                       2813
037420  000207                          RTS     PC

                        ; Routine Size:  28 words
                        ; Maximum stack depth per invocation:  8 words
```

```
3072 ;MLX4
3073 ;                  DRIVE IDENTIFICATION ROUTINES                    27-Mar-1982 19:24:42    TOPS-20 Bliss-16 V2(212)
3074 ;                                                                   27-Mar-1982 19:23:44    PA:<NEALE>MLX4.BLI.5 (11)
3075 ;          2833    routine UP_HARD_COUNT (LUN, ARRAY) : novalue =
3076 ;          2834        begin
3077 ;          2835
3078 ;          2836    !++
3079 ;          2837    !  ROUTINE:     UP_HARD_COUNT(LUN,ARRAY)
3080 ;          2838    !
3081 ;          2839    !  PURPOSE:     TO INCREMENT THE HARD ERROR COUNT FOR THE GIVEN
3082 ;          2840    !               ARRAY, AND TO SEE IF THE HARD ERROR THRESHOLD HAS
3083 ;          2841    !               BEEN REACHED.
3084 ;          2842    !
3085 ;          2843    !  ARGUMENTS:   LUN   = LOGICAL UNIT WHICH HAS THE HARD ERROR
3086 ;          2844    !               ARRAY = THE BOARD NUMBER (0-15)
3087 ;          2845    !--
3088 ;          2846
3089 ;          2847        local                                            ! VER CZMLBB ADDED THIS LOCAL STORAGE
3090 ;          2848            SBE_EXIST;                                   !Indicates if the searched sbe already exists
3091 ;          2849
3092 ;          2850        HARDS [.LUN, .ARRAY, 0, 16, 0] = .HARDS [.LUN, .ARRAY, 0, 16, 0] + 1;
3093 ;          2851        !
3094 ;          2852        ! VER CZMLBB ADDED FOLLOWING CODE TO LOG SBE'S
3095 ;          2853        !
3096 ;          2854        ! First search the sbe log table and see if
3097 ;          2855        ! this single bit error already exists in
3098 ;          2856        ! the table.  If it does exist then just
3099 ;          2857        ! increment its occurance count,  else
3100 ;          2858        ! record its failing location into the
3101 ;          2859        ! table.
3102 ;          2860        !
3103 ;          2861        ! Only log away 128 single bit errors.
3104 ;          2862        !
3105 ;          2863
3106 ;          2864        if .SBE$_COUNT lss 127                           !Have 128 errors been logged yet
3107 ;          2865        then
3108 ;          2866            begin                                        !There is room for at least one more sbe
3109 ;          2867            SBE_EXIST = FALSE;                           !Init the existance flag to false
3110 ;          2868
3111 ;          2869            !+
3112 ;          2870            ! Search the single bit error log table
3113 ;          2871            ! and see if this error already exist.
3114 ;          2872            !-
3115 ;          2873
3116 ;          2874            incr index from 0 to .SBE$_COUNT do
3117 ;          2875                begin
3118 ;          2876
3119 ;          2877                if (.SBE_LOG [.index, BIT$_SBE] eql .BIT_NUM) and   !Does this bit exist
3120 ;          2878                    (.SBE_LOG [.index, BNK$_SBE] eql .BANK) and     !Does this bank exist
3121 ;          2879                    (.SBE_LOG [.index, BRD$_SBE] eql .BOARD) and    !Does this board exist
3122 ;          2880                    (.SBE_LOG [.index, UNIT$_SBE] eql .LUN) !Does this unit exist
3123 ;          2881                then
3124 ;          2882                    begin                                !This sbe already exist so just up its occurance count
3125 ;          2883                    SBE_LOG [.index, SUM$_SBE] = .SBE_LOG [.index, SUM$_SBE] + 1;
3126 ;          2884                    SBE_EXIST = TRUE;                    !Indicate that this sbe already exist
```

```
;          2885                    exitloop;
;          2886                    end;
;          2887
;          2888                end;
;          2889
;          2890            !+
;          2891            ! Test to see if this sbe was already in the table
;          2892            ! by testing the flag sbe_exist.  If it did not exist then
;          2893            ! up the Prom Maintenance count, load this sbe into the table
;          2894            ! and test to see if this array module needs to be Prom Maintenanced.
;          2895            ! If the sbe did exist then return from this routine.
;          2896            !-
;          2897
;          2898            if not .SBE_EXIST                       !Was this sbe found in the table
;          2899            then
;          2900                begin                               !It was not found in the table so up the pm count
;          2901                PM_SBE_CNT [.LUN, .BOARD, PM_SBE$_SUM] = .PM_SBE_CNT [.LUN, .BOARD, PM_SBE$_SUM] + 1;
;          2902                !
;          2903                ! Now see if the Prom Maint Program needs to be
;          2904                ! run on this array module.
;          2905                !
;          2906
;          2907                if .PM_SBE_CNT [.LUN, .BOARD, PM_SBE$_SUM] eql 10   !Is the limit exceeded
;          2908                then
;          2909                    begin                           !The limit was exceeded
;          2910                    SAYWHO (.LUN);
;          2911                    PRINTB (FMT13, .ARRAY, MSG2);
;          2912                    !'ARRAY XX  --> RUN ML11 PROM MAINTENANCE PROGRAM'
;          2913                    end;
;          2914
;          2915                !+
;          2916                ! This sbe was not found in the table so
;          2917                ! load this failing sbe into the table
;          2918                ! at the bottom of the list.
;          2919                !-
;          2920
;          2921                SBE_LOG [.SBE$_COUNT, BIT$_SBE] = .BIT_NUM; !Load the failing bit
;          2922                SBE_LOG [.SBE$_COUNT, BNK$_SBE] = .BANK;    !Load the failing bank
;          2923                SBE_LOG [.SBE$_COUNT, BRD$_SBE] = .BOARD;   !Load the failing board
;          2924                SBE_LOG [.SBE$_COUNT, UNIT$_SBE] = .LUN;    !Load the failing unit
;          2925                SBE_LOG [.SBE$_COUNT, SUM$_SBE] = 1;        !Indicate this is the first one
;          2926                SBE$_COUNT = .SBE$_COUNT + 1;       !Up the count of unique sbe's detected
;          2927                end;
;          2928
;          2929            end;
;          2930
;          2931    !
;          2932    ! VER COMMENTED OUT THIS CODE
;          2933    ! REPLACED BY ABOVE CODE
;          2934    !
;          2935    !       if (((.ARR_TYP eql 1) and (.HARDS [.LUN, .ARRAY, 0, 16, 0] eql H64K_LIMIT)) or ((.ARR_TYP eql 0) and (
;          2936    !           .HARDS [.LUN, .ARRAY, 0, 16, 0] eql H16K_LIMIT)))
```

```
;MLX4
;                    DRIVE IDENTIFICATION ROUTINES                    27-Mar-1982 19:24:42    TOPS-20 Bliss-16 V2(212)
;                                                                     27-Mar-1982 19:23:44    PA:<NEALE>MLX4.BLI.5 (11)
;          2937  !     then
;          2938  !        begin
;          2939  !        SAYWHO (.LUN);
;          2940  !        PRINTB (FMT13, .ARRAY, MSG2);
;          2941  !        !'ARRAY XX  --> RUN ML11 PROM MAINTENANCE PROGRAM'
;          2942  !        end;
;          2943       return;
;          2944       end;

                                        .SBTTL  UP.HARD.COUNT DRIVE IDENTIFICATION ROUTINES
037422                          UP.HARD.COUNT:
037422  004167  145706                JSR     R1,$SAVE5                       ;                               2833
037426  016605  000020                MOV     20(SP),R5                       ; LUN,*                         2850
037432  010500                        MOV     R5,R0
037434  006300                        ASL     R0
037436  006300                        ASL     R0
037440  006300                        ASL     R0
037442  006300                        ASL     R0
037444  010002                        MOV     R0,R2
037446  066602  000016                ADD     16(SP),R2                       ; ARRAY,*
037452  006302                        ASL     R2
037454  005262  033314                INC     HARDS(R2)
037460  026727  153200  000177        CMP     SBE$.COUNT,#177                 ;                               2864
037466  002401                        BLT     1$
037470  000207                        RTS     PC
037472  005004                1$:     CLR     R4                              ; SBE.EXIST                     2867
037474  005001                        CLR     R1                              ; INDEX                         2874
037476  000463                        BR      4$
037500  010102                2$:     MOV     R1,R2                           ; INDEX,*                       2877
037502  006302                        ASL     R2
037504  006302                        ASL     R2
037506  012703  011264                MOV     #SBE.LOG,R3
037512  060203                        ADD     R2,R3
037514  016746  153146                MOV     BIT.NUM,-(SP)
037520  011346                        MOV     (R3),-(SP)
037522  006216                        ASR     (SP)
037524  006216                        ASR     (SP)
037526  006216                        ASR     (SP)
037530  006216                        ASR     (SP)
037532  006216                        ASR     (SP)
037534  006216                        ASR     (SP)
037536  042716  177600                BIC     #177600,(SP)
037542  022626                        CMP     (SP)+,(SP)+
037544  001037                        BNE     3$
037546  016746  174572                MOV     BANK,-(SP)                      ;                               2878
037552  111346                        MOVB    (R3),-(SP)
```

```
                                ;MLX4
                                ;                       DRIVE IDENTIFICATION ROUTINES                   27-Mar-1982 19:24:42    TOPS
                                ;                                                                       27-Mar-1982 19:23:44    PA:<
037554  042716  177774                BIC     #177774,(SP)
037560  022626                        CMP     (SP)+,(SP)+
037562  001030                        BNE     3$
037564  016746  174552                MOV     BOARD,-(SP)             ;                               2879
037570  111346                        MOVB    (R3),-(SP)
037572  006216                        ASR     (SP)
037574  006216                        ASR     (SP)
037576  042716  177760                BIC     #177760,(SP)
037602  022626                        CMP     (SP)+,(SP)+
037604  001017                        BNE     3$
037606  010546                        MOV     R5,-(SP)
037610  011346                        MOV     (R3),-(SP)              ;                               2880
037612  006116                        ROL     (SP)
037614  006116                        ROL     (SP)
037616  006116                        ROL     (SP)
037620  006116                        ROL     (SP)
037622  042716  177770                BIC     #177770,(SP)
037626  022626                        CMP     (SP)+,(SP)+
037630  001005                        BNE     3$
037632  005262  011266                INC     SBE.LOG+2(R2)           ;                               2883
037636  012704  000001                MOV     #1,R4                   ; *,SBE.EXIST                   2884
037642  000404                        BR      5$                      ;                               2885
037644  005201          3$:           INC     R1                      ; INDEX                         2874
037646  020167  153012  4$:           CMP     R1,SBE$.COUNT           ; INDEX,*
037652  003712                        BLE     2$
037654  006004          5$:           ROR     R4                      ; SBE.EXIST                     2898
037656  103511                        BLO     7$
037660  010002                        MOV     R0,R2                   ;                               2901
037662  066702  174454                ADD     BOARD,R2
037666  006302                        ASL     R2
037670  005262  012264                INC     PM.SBE.CNT(R2)
037674  026227  012264  000012        CMP     PM.SBE.CNT(R2),#12      ;                               2907
037702  001017                        BNE     6$
037704  010546                        MOV     R5,-(SP)                ;                               2910
037706  004767  175612                JSR     PC,SA:WHO
037712  012716  011120                MOV     #MSG2,(SP)              ;                               2911
037716  016646  000020                MOV     20(SP),-(SP)            ; ARRAY,*
037722  012746  007024                MOV     #FMT13,-(SP)
037726  012746  000003                MOV     #3,-(SP)
037732  010600                        MOV     SP,R0                   ; SP,*
037734  104414                        TRAP    14
037736  062706  000010                ADD     #10,SP                  ;
037742  016702  152716          6$:   MOV     SBE$.COUNT,R2           ;                               2909
037746  006302                        ASL     R2                                                      2921
037750  006302                        ASL     R2
037752  012701  011264                MOV     #SBE.LOG,R1
037756  060201                        ADD     R2,R1
037760  016704  152702                MOV     BIT.NUM,R4
037764  000304                        SWAB    R4
037766  106004                        RORB    R4
037770  006004                        ROR     R4
037772  006004                        ROR     R4
```

```
                                ;MLX4                                                                   27-Mar-1982 19:24:42    TOPS
                                ;                DRIVE IDENTIFICATION ROUTINES                          27-Mar-1982 19:23:44    PA:<
                                
037774  042704  160077          BIC     #160077,R4
040000  042711  017700          BIC     #17700,(R1)
040004  050411                  BIS     R4,(R1)
040006  016704  174332          MOV     BANK,R4                                 ;                       2922
040012  042704  177774          BIC     #177774,R4
040016  142711  000003          BICB    #3,(R1)
040022  150411                  BISB    R4,(R1)
040024  016704  174312          MOV     BOARD,R4                                ;                       2923
040030  006304                  ASL     R4
040032  006304                  ASL     R4
040034  042704  177703          BIC     #177703,R4
040040  142711  000074          BICB    #74,(R1)
040044  150411                  BISB    R4,(R1)
040046  006005                  ROR     R5                                      ;                       2924
040050  006005                  ROR     R5
040052  006005                  ROR     R5
040054  006005                  ROR     R5
040056  042705  017777          BIC     #17777,R5
040062  042711  160000          BIC     #160000,(R1)
040066  050511                  BIS     R5,(R1)
040070  012762  000001  011266  MOV     #1,SBE.LOG+2(R2)                        ;                       2925
040076  005267  152562          INC     SBE$.COUNT                              ;                       2926
040102  000207          7$:     RTS     PC                                      ;                       2833
                                
                                ; Routine Size:  153 words
                                ; Maximum stack depth per invocation:  10 words
```

```
;MLX4
;                DRIVE IDENTIFICATION ROUTINES                    27-Mar-1982 19:24:42    TOPS-20 Bliss-16 V2(212)
;                                                                 27-Mar-1982 19:23:44    PA:<NEALE>MLX4.BLI.5 (12)

;          2945  routine UP_SOFT_COUNT (LUN, ARRAY) : novalue =
;          2946      begin
;          2947
;          2948  !++
;          2949  !  ROUTINE:     UP_SOFT_COUNT(LUN,ARRAY)
;          2950  !
;          2951  !  PURPOSE:     TO INCREMENT THE SOFT ERROR COUNT FOR THE GIVEN
;          2952  !               ARRAY, AND TO SEE IF THE SOFT ERROR THRESHOLD HAS
;          2953  !               BEEN REACHED.
;          2954  !
;          2955  !  ARGUMENTS:   LUN   = LOGICAL UNIT WHICH HAS THE SOFT ERROR
;          2956  !               ARRAY = THE BOARD NUMBER (0-15)
;          2957  !--
;          2958
;          2959      local                                        ! VER CZMLBB ADD THIS LOCAL STORAGE
;          2960          SBE_EXIST;                               !Flag to indicate if sbe already exists
;          2961
;          2962      SOFTS [.LUN, .ARRAY, 0, 16, 0] = .SOFTS [.LUN, .ARRAY, 0, 16, 0] + 1;
;          2963      !
;          2964      ! VER CZMLBB ADDED FOLLOWING CODE TO LOG SBE'S
;          2965      !
;          2966      ! First search the sbe log table and see if
;          2967      ! this single bit error already exists in
;          2968      ! the table.  If it does exist then just
;          2969      ! increment its occurance count,  else
;          2970      ! record its failing location into the
;          2971      ! table.
;          2972      !
;          2973      ! Only log away 128 single bit errors.
;          2974      !
;          2975
;          2976      if .SBE$_COUNT lss 127                       !Have 128 errors been logged yet
;          2977      then
;          2978          begin                                    !There is room for at least one more sbe
;          2979          SBE_EXIST = FALSE;                       !Init the existence flage to false
;          2980
;          2981          !+
;          2982          ! Search the single bit error log table
;          2983          ! and see if this error already exist.
;          2984          !-
;          2985
;          2986          incr index from 0 to .SBE$_COUNT do
;          2987              begin
;          2988
;          2989              if (.SBE_LOG [.index, BIT$_SBE] eql .BIT_NUM) and   !Does this bit exist
;          2990                  (.SBE_LOG [.index, BNK$_SBE] eql .BANK) and     !Does this bank exist
;          2991                  (.SBE_LOG [.index, BRD$_SBE] eql .BOARD) and    !Does this board exist
;          2992                  (.SBE_LOG [.index, UNIT$_SBE] eql .LUN) !Does this unit exist
;          2993              then
;          2994                  begin                            !This sbe already exist so just up its occurance count
;          2995                  SBE_LOG [.index, SUM$_SBE] = .SBE_LOG [.index, SUM$_SBE] + 1;
;          2996                  SBE_EXIST = TRUE;                !Indicate that this sbe already exist
```

```
2997                    exitloop;
2998                    end;
2999
3000                end;
3001
3002            !+
3003            ! Test to see if this sbe was already in the table
3004            ! by testing the flag sbe_exist.  If it did not exist then
3005            ! enter this SBE into the table.
3006            !
3007            ! NOTE:
3008            !    Soft errors are not counted toward the Prom Maintenance
3009            !    call out.
3010            !-
3011
3012            if not .SBE_EXIST                       !Was this sbe found in the table
3013            then
3014                begin                               !It was not found in the table so up the pm count
3015
3016                !+
3017                ! This sbe was not found in the table so
3018                ! load this failing sbe into the table
3019                ! at the bottom of the list.
3020                !-
3021
3022                SBE_LOG [.SBE$_COUNT, BIT$_SBE] = .BIT_NUM; !Load the failing bit
3023                SBE_LOG [.SBE$_COUNT, BNK$_SBE] = .BANK;    !Load the failing bank
3024                SBE_LOG [.SBE$_COUNT, BRD$_SBE] = .BOARD;   !Load the failing board
3025                SBE_LOG [.SBE$_COUNT, UNIT$_SBE] = .LUN;    !Load the failing unit
3026                SBE_LOG [.SBE$_COUNT, SUM$_SBE] = 1;        !Indicate this is the first one
3027                SBE$_COUNT = .SBE$_COUNT + 1;       !Up the count of unique sbe's detected
3028                end;
3029
3030            end;
3031
3032        !
3033        ! VER CZMLBB THIS CODE WAS COMMENTED OUT AND REPLACED BY THE
3034        ! ABOVE CODE.
3035        !
3036        !    if (((.ARR_TYP eql 1) and (.SOFTS [.LUN, .ARRAY, 0, 16, 0] eql S64K_LIMIT)) or ((.ARR_TYP eql 0) and (
3037        !        .SOFTS [.LUN, .ARRAY, 0, 16, 0] eql S16K_LIMIT)))
3038        !    then
3039        !        begin
3040        !        SAYWHO (.LUN);
3041        !        PRINTB (FMT13, .ARRAY, MSG2);
3042        !        !'ARRAY XX  --> RUN ML11 PROM MAINTENANCE PROGRAM'
3043        !        end;
3044        !    return;
3045        !    end;
```

```
                                        ;MLX4
                                        ;               DRIVE IDENTIFICATION ROUTINES                    27-Mar-1982 19:24:42    TOPS
                                                                                                         27-Mar-1982 19:23:44    PA:<
                                                .SBTTL  UP.SOFT.COUNT DRIVE IDENTIFICATION ROUTINES
040104                                  UP.SOFT.COUNT:
040104  004167  145224                        JSR     R1,$SAVE5                       ;                                       2945
040110  016605  000020                        MOV     20(SP),R5                       ; LUN,*                                 2962
040114  010500                                MOV     R5,R0
040116  006300                                ASL     R0
040120  006300                                ASL     R0
040122  006300                                ASL     R0
040124  006300                                ASL     R0
040126  066600  000016                        ADD     16(SP),R0                       ; ARRAY,*
040132  006300                                ASL     R0
040134  005260  032714                        INC     SOFTS(R0)
040140  016701  152520                        MOV     SBE$.COUNT,R1                   ;                                       2976
040144  020127  000177                        CMP     R1,#177
040150  002150                                BGE     5$
040152  005003                                CLR     R3                              ; SBE.EXIST                             2979
040154  005000                                CLR     R0                              ; INDEX                                 2986
040156  000463                                BR      3$
040160  010004                          1$:   MOV     R0,R4                           ; INDEX,*                               2989
040162  006304                                ASL     R4
040164  006304                                ASL     R4
040166  012702  011264                        MOV     #SBE.LOG,R2
040172  060402                                ADD     R4,R2
040174  016746  152466                        MOV     BIT.NUM,-(SP)
040200  011246                                MOV     (R2),-(SP)
040202  006216                                ASR     (SP)
040204  006216                                ASR     (SP)
040206  006216                                ASR     (SP)
040210  006216                                ASR     (SP)
040212  006216                                ASR     (SP)
040214  006216                                ASR     (SP)
040216  042716  177600                        BIC     #177600,(SP)
040222  022626                                CMP     (SP)+,(SP)+
040224  001037                                BNE     2$
040226  016746  174112                        MOV     BANK,-(SP)                      ;                                       2990
040232  111246                                MOVB    (R2),-(SP)
040234  042716  177774                        BIC     #177774,(SP)
040240  022626                                CMP     (SP)+,(SP)+
040242  001030                                BNE     2$
040244  016746  174072                        MOV     BOARD,-(SP)                     ;                                       2991
040250  111246                                MOVB    (R2),-(SP)
040252  006216                                ASR     (SP)
040254  006216                                ASR     (SP)
040256  042716  177760                        BIC     #177760,(SP)
040262  022626                                CMP     (SP)+,(SP)+
040264  001017                                BNE     2$
040266  010546                                MOV     R5,-(SP)                        ;                                       2992
040270  011246                                MOV     (R2),-(SP)
```

```
                                ;MLX4                                                                   27-Mar-1982 19:24:42    TOPS
                                ;               DRIVE IDENTIFICATION ROUTINES                           27-Mar-1982 19:23:44    PA:<
040272  006116                          ROL     (SP)
040274  006116                          ROL     (SP)
040276  006116                          ROL     (SP)
040300  006116                          ROL     (SP)
040302  042716  177770                  BIC     #177770,(SP)
040306  022626                          CMP     (SP)+,(SP)+
040310  001005                          BNE     2$
040312  005264  011266                  INC     SBE.LOG+2(R4)           ;                               2995
040316  012703  000001                  MOV     #1,R3                   ; *,SBE.EXIST                   2996
040322  000403                          BR      4$                      ;                               2997
040324  005200                  2$:     INC     R0                      ; INDEX                         2986
040326  020001                  3$:     CMP     R0,R1                   ; INDEX,*
040330  003713                          BLE     1$
040332  006003                  4$:     ROR     R3                      ; SBE.EXIST                     3012
040334  103456                          BLO     5$
040336  006301                          ASL     R1                      ;                               3022
040340  006301                          ASL     R1
040342  012700  011264                  MOV     #SBE.LOG,R0
040346  060100                          ADD     R1,R0
040350  016704  152312                  MOV     BIT.NUM,R4
040354  000304                          SWAB    R4
040356  106004                          RORB    R4
040360  006004                          ROR     R4
040362  006004                          ROR     R4
040364  042704  160077                  BIC     #160077,R4
040370  042710  017700                  BIC     #17700,(R0)
040374  050410                          BIS     R4,(R0)
040376  016704  173742                  MOV     BANK,R4                 ;                               3023
040402  042704  177774                  BIC     #177774,R4
040406  142710  000003                  BICB    #3,(R0)
040412  150410                          BISB    R4,(R0)
040414  016704  173722                  MOV     BOARD,R4                ;                               3024
040420  006304                          ASL     R4
040422  006304                          ASL     R4
040424  042704  177703                  BIC     #177703,R4
040430  142710  000074                  BICB    #74,(R0)
040434  150410                          BISB    R4,(R0)
040436  006005                          ROR     R5                      ;                               3025
040440  006005                          ROR     R5
040442  006005                          ROR     R5
040444  006005                          ROR     R5
040446  042705  017777                  BIC     #17777,R5
040452  042710  160000                  BIC     #160000,(R0)
040456  050510                          BIS     R5,(R0)
040460  012761  000001  011266          MOV     #1,SBE.LOG+2(R1)        ;                               3026
040466  005267  152172                  INC     SBE$.COUNT              ;                               3027
040472  000207                  5$:     RTS     PC                      ;                               2945

                                ; Routine Size:  124 words
                                ; Maximum stack depth per invocation:  8 words
```

```
;MLX4
;                    INITIALIZATION CODE                                    27-Mar-1982 19:24:42    TOPS-20 Bliss-16 V2(212)
;                                                                           27-Mar-1982 19:23:44    PA:<NEALE>MLX4.BLI.5 (13)

;          3046  %sbttl 'INITIALIZATION CODE'
;          3047  BGNINIT;                                                  !* 1 *
;          3048  !       INITIALIZATION CODE IS EXECUTED AT THE BEGINNING OF EACH
;          3049  !       PASS, WHEN POWER DOWN/POWER UP HAS OCCURRED, OR WHEN THE
;          3050  !       OPERATOR HAS ISSUED A START, RESTART OR CONTINUE COMMAND.
;          3051  !
;          3052  !       DURING INITIALIZATION, THE 'GPHARD' MACRO IS USED TO GET
;          3053  !       P-TABLE INFORMATION FOR THE LOGICAL UNIT UNDER TEST. THE
;          3054  !       NUMBER OF UNITS  AVAILABLE FOR TESTING IS CONTAINED IN A
;          3055  !       HEADER LOCATION ('L$UNIT').
;          3056  !
;          3057  !       THE CODE WITHIN THE INITIALIZATION SECTION IS DIVIDED
;          3058  !       INTO THREE CATEGORIES:
;          3059  !               1)  ONCE-ONLY CODE WHICH IS EXECUTED
;          3060  !                   ONLY ON THE VERY FIRST PASS (I.E.,
;          3061  !                   WHEN THE OPERATOR HAS JUST TYPED
;          3062  !                   IN THE 'START' COMMAND;
;          3063  !               2)  CODE WHICH SHOULD NOT BE EXECUTED
;          3064  !                   ON THE FIRST PASS, BUT WHICH IS
;          3065  !                   TO BE RUN ON EVERY SUBSEQUENT PASS;
;          3066  !               3)  COMMON CODE WHICH IS TO BE EXECUTED
;          3067  !                   ON EVERY PASS (INCLUDING THE FIRST).
;          3068
;          3069  local
;          3070      PLOC;
;          3071
;          3072  if ((READEF (EF_START)) or (READEF (EF_RESTART)))
;          3073  then                                                      !THIS IS CATEGORY 1 CODE
;          3074      begin                                                 !* 2 *
;          3075      QUICK = 1;
;          3076      CLRTBLS ();
;          3077
;          3078      incr LUN from 0 to (.L$UNIT - 1) do
;          3079          begin                                             !* 3 *
;          3080          L$LUN = .LUN;
;          3081
;          3082          if GPHARD (.LUN, PLOC) neq 0
;          3083          then
;          3084              begin                                         !* 4 *
;          3085              PTABLE_ADDR [.LUN] = .PLOC;
;          3086
;          3087              if .NUM_DRIVES eql 0 then INIT_ADDRESSES (.PLOC);
;          3088
;          3089              NUM_DRIVES = .NUM_DRIVES + 1;
;          3090              DRIVE_STATUS [.LUN] = ACTIVE;
;          3091              CONFIG (.LUN),
;          3092              end                                           !* 4 *
;          3093          else
;          3094              DRIVE_STATUS [.LUN] = INACTIVE;
;          3095
;          3096          end;                                              !* 3 *
;          3097
```

```
;MLX4                                                                    27-Mar-1982 19:24:42    TOPS-20 Bliss-16 V2(212)
;                    INITIALIZATION CODE                                 27-Mar-1982 19:23:44    PA:<NEALE>MLX4.BLI.5 (13)
;
;          3098          end                                             !* 2 *
;          3099      else
;          3100          QUICK = 0;                                      !THIS IS CATEGORY 2 CODE
;          3101
;          3102      !CATEGORY 3 CODE WOULD GO HERE, BUT THERE IS NONE.
;          3103      ENDINIT;                                            !* 1 *
;
;                                        .SBTTL  LINIT INITIALIZATION CODE
040474  004167  144614          LINIT:  JSR     R1,$SAVE4                                                       ;
040500  012700  000040                  MOV     #40,R0                  ;                                       ;3045
040504  104447                          TRAP    47                      ;                                       ;3072
040506  103404                          BCS     1$
040510  012700  000037                  MOV     #37,R0
040514  104447                          TRAP    47
040516  103076                          BHIS    7$
040520  012767  000001  172146  1$:     MOV     #1,QUICK                ;                                       ;3075
040526  004767  174022                  JSR     PC,CLRTBLS              ;                                       ;3076
040532  016704  141254                  MOV     L$UNIT,R4               ;                                       ;3078
040536  005003                          CLR     R3                      ; LUN
040540  000462                          BR      6$
040542  010367  141326          2$:     MOV     R3,L$LUN                ; LUN,*                                 ;3080
040546  010302                          MOV     R3,R2                   ; LUN,*                                 ;3090
040550  006202                          ASR     R2
040552  006202                          ASR     R2
040554  006202                          ASR     R2
040556  062702  034442                  ADD     #DRIVE.STATUS,R2
040562  010300                          MOV     R3,R0                   ; LUN,*                                 ;3082
040564  104442                          TRAP    42
040566  010001                          MOV     R0,R1                   ; *,PLOC
040570  001432                          BEQ     4$
040572  010300                          MOV     R3,R0                   ; LUN,*                                 ;3085
040574  006300                          ASL     R0
040576  010160  034422                  MOV     R1,PTABLE.ADDR(R0)      ; PLOC,*
040602  005767  173650                  TST     NUM.DRIVES              ;                                       ;3087
040606  001004                          BNE     3$
040610  010146                          MOV     R1,-(SP)                ; PLOC,*
040612  004767  174370                  JSR     PC,INIT.ADDRESSES
040616  005726                          TST     (SP)+
040620  005267  173632          3$:     INC     NUM.DRIVES              ;                                       ;3089
040624  010246                          MOV     R2,-(SP)                ;                                       ;3090
040626  010346                          MOV     R3,-(SP)                ; LUN,*
040630  042716  177770                  BIC     #177770,(SP)
040634  012746  000001                  MOV     #1,-(SP)
040640  011646                          MOV     (SP),-(SP)
040642  004767  143746                  JSR     PC,BL$PU2
040646  010316                          MOV     R3,(SP)                 ; LUN,*                                 ;3091
```

```
                                ;MLX4                                                                   27-Mar-1982 19:24:42    TOPS
                                ;                       INITIALIZATION CODE                             27-Mar-1982 19:23:44    PA:<

040650  004767  175126                    JSR     PC,CONFIG
040654  000411                            BR      5$
040656  010246                4$:         MOV     R2,-(SP)                ;                                                       3082
040660  010346                            MOV     R3,-(SP)                ;                                                       3094
040662  042716  177770                    BIC     #177770,(SP)            ; LUN,*
040666  012746  000001                    MOV     #1,-(SP)
040672  005046                            CLR     -(SP)
040674  004767  143714                    JSR     PC,BL$PU2
040700  062706  000010                5$:         ADD     #10,SP                  ;                                               3079
040704  005203                            INC     R3                      ; LUN                                                   3078
040706  020304                6$:         CMP     R3,R4                   ; LUN,*
040710  002714                            BLT     2$
040712  000207                            RTS     PC
040714  005067  171754                7$:         CLR     QUICK                   ;                                               3072
040720  000207                            RTS     PC                      ;                                                       3100
                                                                                ;                                               3045
                                ; Routine Size:  75 words
                                ; Maximum stack depth per invocation:  9 words
                                .SBTTL  L$INIT INITIALIZATION CODE
040722  004767  177546        L$INIT::JSR     PC,LINIT                ;                                                       3100
040726  104411                        TRAP    11
040730  000207                        RTS     PC
                                ; Routine Size:  4 words
                                ; Maximum stack depth per invocation:  0 words
```

```
:MLX4
:                GENERATOR FOR OPTION 1                          27-Mar-1982 19:24:42    TOPS-20 Bliss-16 V2(212)
:                                                                27-Mar-1982 19:23:44    PA:<NEALE>MLX4.BLI.5 (14)

;          3104  %sbttl 'GENERATOR FOR OPTION 1'
;          3105  routine GEN1 (SECTOR, COMP_FLAG) : novalue =
;          3106      begin                                       !* 1 *
;          3107
;          3108  !++
;          3109  !  ROUTINE:     GEN1(SECTOR,COMP_FLAG)
;          3110  !
;          3111  !  PURPOSE:     TO GENERATE THE ENTIRE 4K-WORD WRITE BUFFER FOR OPTION 1.
;          3112  !               THE GENERATION IS IN 16 GROUPS OF 256 WORDS (THE SIZE
;          3113  !               OF 1 SECTOR).   EACH GROUP OF 256 WORDS RECEIVES THE
;          3114  !               SECTOR ADDRESS WHERE THE WORDS WILL BE TRANSFERRED.
;          3115  !
;          3116  !  ARGUMENTS:   (1)  SECTOR    - CURRENT SECTOR NUMBER.
;          3117  !
;          3118  !               (2)  COMP_FLAG - AN INDICATOR WHICH IS USED TO DECIDE
;          3119  !                                WHETHER THE LOCATION IN THE WRITE BUFFER
;          3120  !                                SHOULD BE COMPLEMENTED.
;          3121  !--
;          3122
;          3123      local
;          3124          OFFSET;
;          3125
;          3126      OFFSET = 0;
;          3127
;          3128      if .COMP_FLAG
;          3129      then                                        !GENERATE COMPLEMENT DATA
;          3130
;          3131          incru SECT from (.SECTOR) to ((.SECTOR) + 15) do
;          3132              begin
;          3133              BREAK;
;          3134
;          3135              incru COUNT from 1 to 256 do
;          3136                  begin                           !* 2A *
;          3137                  (WBUFF + (.OFFSET)) = not (.SECT);      !COMPLEMENT DATA
;          3138                  OFFSET = .OFFSET + 2;
;          3139                  end;                            !* 2A *
;          3140
;          3141              end
;          3142
;          3143      else                                        !GENERATE REGULAR DATA
;          3144
;          3145          incru SECT from (.SECTOR) to ((.SECTOR) + 15) do
;          3146              begin
;          3147              BREAK;
;          3148
;          3149              incru COUNT from 1 to 256 do
;          3150                  begin                           !* 2B *
;          3151                  (WBUFF + (.OFFSET)) = (.SECT);  !REGULAR DATA
;          3152                  OFFSET = .OFFSET + 2;
;          3153                  end;                            !* 2B *
;          3154
;          3155              end;
```

```
3779 ;MLX4
3780 ;                  GENERATOR FOR OPTION 1                    27-Mar-1982 19:24:42   TOPS-20 Bliss-16 V2(212)
3781                                                               27-Mar-1982 19:23:44   PA:<NEALE>MLX4.BLI.5 (14)
3782 ;          3156
3783 ;          3157       return;
3784 ;          3158       end;                                   !* 1 *
3788
3789                                        .SBTTL  GEN1 GENERATOR FOR OPTION 1
3793 040732 004167 144356          GEN1:   JSR     R1,$SAVE4                                                    ;
3794 040736 005002                         CLR     R2                                  ; OFFSET                 3105
3795 040740 016601 000016                  MOV     16(SP),R1                           ; SECTOR,*               3126
3796 040744 010104                         MOV     R1,R4                                                        3131
3797 040746 062704 000017                  ADD     #17,R4
3798 040752 032766 000001 000014           BIT     #1,14(SP)                           ; *,COMP.FLAG            3128
3799 040760 001437                         BEQ     6$
3800 040762 010103                         MOV     R1,R3                               ; *,SECT                 3131
3801 040764 000416                         BR      3$
3802 040766 104422                 1$:     TRAP    22                                                           ;
3803 040770 010301                         MOV     R3,R1                               ; SECT,*                 3132
3804 040772 005101                         COM     R1                                                           3137
3805 040774 012700 000001                  MOV     #1,R0                               ; *,COUNT                3135
3806 041000 010162 012670          2$:     MOV     R1,WBUFF(R2)                        ; *,*(OFFSET)            3137
3807 041004 062702 000002                  ADD     #2,R2                               ; *,OFFSET               3138
3808 041010 005200                         INC     R0                                  ; COUNT                  3135
3809 041012 020027 000400                  CMP     R0,#400                             ; COUNT,*
3810 041016 101770                         BLOS    2$
3811 041020 005203                         INC     R3                                  ; SECT                   3131
3812 041022 020304                 3$:     CMP     R3,R4                               ; SECT,*
3813 041024 101760                         BLOS    1$
3814 041026 000207                         RTS     PC                                                           ;
3815 041030 104422                 4$:     TRAP    22                                                           ;  3128
3816 041032 012700 000001                  MOV     #1,R0                                                        ;  3146
3817 041036 010162 012670          5$:     MOV     R1,WBUFF(R2)                        ; *,COUNT                3149
3818 041042 062702 000002                  ADD     #2,R2                               ; SECT,*(OFFSET)         3151
3819 041046 005200                         INC     R0                                  ; *,OFFSET               3152
3820 041050 020027 000400                  CMP     R0,#400                             ; COUNT                  3149
3821 041054 101770                         BLOS    5$                                  ; COUNT,*
3822 041056 005201                         INC     R1
3823 041060 020104                 6$:     CMP     R1,R4                               ; SECT                   3145
3824 041062 101762                         BLOS    4$                                  ; SECT,*
3825 041064 000207                         RTS     PC                                                           ;
3826                                                                                                            3105
3827
3828                               ; Routine Size:  46 words
                                  ; Maximum stack depth per invocation:  5 words
```

```
3838 ;          3159  %sbttl 'PATTERN NUMBER SELECTION'
3839 ;          3160  routine SELPAT : novalue =
3840 ;          3161      begin
3841 ;          3162
3842 ;          3163  !++
3843 ;          3164  !  ROUTINE:     SELPAT
3844 ;          3165  !
3845 ;          3166  !  PURPOSE:     TO SELECT THE NEXT PATTERN NUMBER IN OPTION 2.
3846 ;          3167  !
3847 ;          3168  !               THIS ROUTINE AUTOMATICALLY COMPLEMENTS THE CURRENT
3848 ;          3169  !               PATTERN NUMBER AND EXAMINES THE SIGN OF THE RESULT.
3849 ;          3170  !               IF THE RESULT IS NEGATIVE, THEN IT IS TAKEN TO BE
3850 ;          3171  !               THE PATTERN NUMBER OF A COMPLEMENT PATTERN AND A
3851 ;          3172  !               'RETURN' IS EXECUTED.  IF, HOWEVER, THE RESULT IS
3852 ;          3173  !               POSITIVE, THEN AN ENTIRELY NEW PATTERN NUMBER IS
3853 ;          3174  !               REQUIRED.  THE NEW PATTERN NUMBER IS OBTAINED BY
3854 ;          3175  !               INCREMENTING THE OLD.
3855 ;          3176  !--
3856 ;          3177
3857 ;          3178      PATTERN = -(.PATTERN);
3858 ;          3179
3859 ;          3180      if .PATTERN geq 0 then PATTERN = .PATTERN + 1;
3860 ;          3181
3861 ;          3182      return;
3865 ;          3183      end;
3866

                                    .SBTTL  SELPAT PATTERN NUMBER SELECTION
3870 041066  005467  171604  SELPAT: NEG     PATTERN                          ;3178
3871 041072  005767  171600          TST     PATTERN                          ;3180
3872 041076  002402                  BLT     1$
3873 041100  005267  171572          INC     PATTERN
3874 041104  000207          1$:     RTS     PC                               ;3160
3875
3876                                ; Routine Size:  8 words
3877                                ; Maximum stack depth per invocation:  0 words
3882
3883
```

```
3184  %sbttl 'USING THE PATTERN TABLE'
3185  routine GETCNTS (PATTERN) =
3186      begin
3187
3188  !++
3189  !   ROUTINE:     GETCNTS(PATTERN)
3190  !
3191  !   PURPOSE:     TO POINT TO THE APPROPRIATE BLOCK OF THE PATTERN
3192  !                TABLE AND GRAB THE VALUES WHICH ARE REQUIRED FOR THE
3193  !                PATTERN GENERATOR FOR OPTIONS 2 AND 4.
3194  !
3195  !   ARGUMENT:    PATTERN = THE CURRENT PATTERN NUMBER.
3196  !
3197  !   RESULTS:     (1)  'VALUE' RECEIVES THE VALUE WHICH IS RETURNED FOR
3198  !                     THE SUBROUTINE.  IT IS THE 4-BIT BINARY AMOUNT WHICH
3199  !                     WILL BE USED AS THE CONTENTS OF A NIBBLE OF COMPLEMENT
3200  !                     DATA.
3201  !
3202  !                (2)  'DATA_COUNT' RECEIVES THE NUMBER OF NIBBLES OF DATA.
3203  !
3204  !                (3)  'COMP_COUNT' RECEIVES THE NUMBER OF NIBBLES OF
3205  !                     COMPLEMENT DATA.
3206  !--
3207
3208      local
3209          PATNUM,
3210          VALUE;
3211
3212      selectone .PATTERN of
3213          set
3214
3215          [1 to 5] :
3216              begin
3217              PATNUM = .PATTERN - 1;
3218              VALUE = %b'1010';
3219              end;
3220
3221          [-5 to -1] :
3222              begin
3223              PATNUM = -(.PATTERN) - 1;
3224              VALUE = %b'0101';
3225              end;
3226
3227          [5 to 10] :
3228              begin
3229              PATNUM = .PATTERN - 6;
3230              VALUE = %b'1111';
3231              end;
3232
3233          [-10 to -6] :
3234              begin
3235              PATNUM = -(.PATTERN) - 6;
```

```
3942
3943 ;          3236                VALUE = %b'0000';
3944 ;          3237                end;
3945 ;          3238            tes;
3946 ;          3239
3947 ;          3240        DATA_COUNT = .PATTBL [.PATNUM, COUNT1];
3948 ;          3241        COMP_COUNT = .PATTBL [.PATNUM, COUNT2];
3949 ;          3242        return .VALUE;
3950 ;          3243        end;
3954
3955                                    .SBTTL  GETCNTS USING THE PATTERN TABLE
3959 041106 004167 144150          GETCNTS:JSR     R1,$SAVE2                       ;                                       3185
3960 041112 016600 000010                  MOV     10(SP),R0                       ; PATTERN,*                             3212
3961 041116 003410                         BLE     1$
3962 041120 020027 000005                  CMP     R0,#5
3963 041124 003005                         BGT     1$
3964 041126 010001                         MOV     R0,R1                           ; *,PATNUM                              3217
3965 041130 005301                         DEC     R1                              ; PATNUM
3966 041132 012702 000012                  MOV     #12,R2                          ; *,VALUE                               3218
3967 041136 000441                         BR      4$                              ;                                       3212
3968 041140 020027 177773          1$:     CMP     R0,#-5
3969 041144 002410                         BLT     2$
3970 041146 005700                         TST     R0
3971 041150 002006                         BGE     2$
3972 041152 012701 177777                  MOV     #-1,R1                          ; *,PATNUM                              3223
3973 041156 160001                         SUB     R0,R1                           ; *,PATNUM
3974 041160 012702 000005                  MOV     #5,R2                           ; *,VALUE                               3224
3975 041164 000426                         BR      4$                              ;                                       3212
3976 041166 020027 000006          2$:     CMP     R0,#6
3977 041172 002411                         BLT     3$
3978 041174 020027 000012                  CMP     R0,#12
3979 041200 003006                         BGT     3$
3980 041202 010001                         MOV     R0,R1                           ; *,PATNUM                              3229
3981 041204 162701 000006                  SUB     #6,R1                           ; *,PATNUM
3982 041210 012702 000017                  MOV     #17,R2                          ; *,VALUE                               3230
3983 041214 000412                         BR      4$                              ;                                       3212
3984 041216 020027 177766          3$:     CMP     R0,#-12
3985 041222 002407                         BLT     4$
3986 041224 020027 177772                  CMP     R0,#-6
3987 041230 003004                         BGT     4$
3988 041232 012701 177772                  MOV     #-6,R1                          ; *,PATNUM                              3235
3989 041236 160001                         SUB     R0,R1                           ; *,PATNUM
3990 041240 005002                         CLR     R2                              ; VALUE                                 3236
3991 041242 006301                 4$:     ASL     R1                              ;                                       3240
3992 041244 006301                         ASL     R1
3993 041246 016167 034520 171424           MOV     PATTBL(R1),DATA.COUNT
3994 041254 016167 034522 171420           MOV     PATTBL+2(R1),COMP.COUNT         ;                                       3241
3995                               ;MLX4
3996                               ;                       USING THE PATTERN TABLE                 27-Mar-1982 19:24:42    TOPS
3997                                                                                                  27-Mar-1982 19:23:44    PA:<
3998 041262 010200                         MOV     R2,R0                           ; VALUE,*                               3186
3999 041264 000207                         RTS     PC                              ;                                       3185
4000
4001                               ; Routine Size:  56 words
4002                               ; Maximum stack depth per invocation:  3 words
4007
4008
```

```
;MLX4
;                NIBBLE GENERATOR

;          3244  %sbttl 'NIBBLE GENERATOR'
;          3245  routine FILLER (BUFFER, WRDCNT, VALUE) : novalue =
;          3246      begin                                          !* 1 *
;          3247
;          3248  !++
;          3249  !  ROUTINE:     FILLER(BUFFER,WRDCNT,VALUE)
;          3250  !
;          3251  !  PURPOSE:     TO LOAD A CHOSEN WRITE BUFFER, ONE NIBBLE AT A TIME, WITH A
;          3252  !               PARTICULAR PATTERN, FOR A SPECIFIED NUMBER OF 16-BIT WORDS.
;          3253  !               THIS ROUTINE IS THE HEART OF THE GENERATOR FOR OPTIONS 2 AND 4.
;          3254  !
;          3255  !               THE 'FILLER' ROUTINE IS A LOOP WHICH WILL ALTERNATE BETWEEN
;          3256  !               THE TWO COUNTS AND FILL THE APPROPRIATE NUMBER OF NIBBLES OF
;          3257  !               THE CHOSEN BUFFER WITH THE CONTENTS OF 'VALUE'.  EVERY TIME
;          3258  !               A COUNT IS EXHAUSTED, THE OTHER COUNT IS STARTED UP AGAIN,
;          3259  !               AND EACH TIME A NEW COUNT IS BEGUN, 'VALUE' IS COMPLEMENTED.
;          3260  !               NOTE THAT THE FIRST COUNT THAT WILL BE USED WILL ALWAYS BE
;          3261  !               'DATA_COUNT', SO THAT THE BUFFER WILL BE REPETITIONS OF:
;          3262  !               DATA, COMP, DATA, COMP ...
;          3263  !
;          3264  !  ARGUMENTS:   (1)  BUFFER = THE STARTING ADDRESS OF THE CHOSEN WRITE BUFFER.
;          3265  !
;          3266  !               (2)  WRDCNT = THE NUMBER OF WORDS OF PATTERN TO BE GENERATED.
;          3267  !
;          3268  !               (3)  VALUE = NIBBLE OF DATA TO BE PLACED IN THE WRITE BUFFER
;          3269  !--
;          3270
;          3271      local
;          3272          NIBBLE,
;          3273          OFFSET,
;          3274          FLAG,
;          3275          THE_COUNT;
;          3276
;          3277      FLAG = 0;
;          3278      OFFSET = 0;
;          3279      NIBBLE = 0;
;          3280
;          3281      while 1 do
;          3282          begin                                      !* 2 *
;          3283          FLAG = not .FLAG;                          !FLIP THE FLAG
;          3284          VALUE = not .VALUE;                        !AND THE DATA
;          3285
;          3286          if .FLAG eql 0                             !CHOOSE A COUNT
;          3287          then
;          3288              THE_COUNT = .COMP_COUNT
;          3289          else
;          3290              THE_COUNT = .DATA_COUNT;
;          3291
;          3292          decr COUNT from .THE_COUNT to 1 do         !EXHAUST THE COUNT
;          3293              begin                                  !* 3 *
;          3294              (.BUFFER + (.OFFSET))<.NIBBLE, 4, 0> = .VALUE;      !LOAD THE NIBBLE
;          3295              NIBBLE = (((.NIBBLE) + 4) mod 16);     !CHANGE NIBBLE POINTER
```

```
4066 ;MLX4
4067 ;                  NIBBLE GENERATOR                                27-Mar-1982 19:24:42    TOPS-20 Bliss-16 V2(212)
4068                                                                     27-Mar-1982 19:23:44    PA:<NEALE>MLX4.BLI.5 (17)
4069 ;          3296
4070 ;          3297                if .NIBBLE eql 0
4071 ;          3298                then
4072 ;          3299                    begin                           !* 4 *
4073 ;          3300                    BREAK;
4074 ;          3301                    OFFSET = .OFFSET + 2;
4075 ;          3302                    WRDCNT = .WRDCNT - 1;
4076 ;          3303
4077 ;          3304                    if .WRDCNT eql 0 then return;
4078 ;          3305
4079 ;          3306                    end;                            !* 4 *
4080 ;          3307
4081 ;          3308                end;                                !* 3 *
4082 ;          3309
4083 ;          3310            end;                                    !* 2 *
4084 ;          3311
4085 ;          3312        end;                                        !* 1 *
4089
4090                                    .SBTTL  FILLER NIBBLE GENERATOR
4094 041266  004167  144042        FILLER: JSR     R1,$SAVE5
4095 041272  005001                        CLR     R1                      ;                               3245
4096 041274  005005                        CLR     R5                      ; FLAG                          3277
4097 041276  005002                        CLR     R2                      ; OFFSET                        3278
4098 041300  005101                1$:     COM     R1                      ; NIBBLE                        3279
4099 041302  005166  000016              COM     16(SP)                  ; FLAG                          3283
4100 041306  005701                        TST     R1                      ; VALUE                         3284
4101 041310  001003                        BNE     2$                      ; FLAG                          3286
4102 041312  016704  171364              MOV     COMP.COUNT,R4           ; *,THE.COUNT                   3288
4103 041316  000402                        BR      3$                      ;                               3286
4104 041320  016704  171354        2$:     MOV     DATA.COUNT,R4           ; *,THE.COUNT                   3290
4105 041324  010403                3$:     MOV     R4,R3                   ; THE.COUNT,COUNT               3292
4106 041326  003764                        BLE     1$
4107 041330  010546                4$:     MOV     R5,-(SP)                ; OFFSET,*                      3294
4108 041332  066616  000024              ADD     24(SP),(SP)             ; BUFFER,*
4109 041336  010246                        MOV     R2,-(SP)                ; NIBBLE,*
4110 041340  012746  000004              MOV     #4,-(SP)
4111 041344  016646  000024              MOV     24(SP),-(SP)            ; VALUE,*
4112 041350  004767  143240              JSR     PC,BL$PU2
4113 041354  010216                        MOV     R2,(SP)                 ; NIBBLE,*                      3295
4114 041356  062716  000004              ADD     #4,(SP)
4115 041362  012746  000020              MOV     #20,-(SP)
4116 041366  004767  143610              JSR     PC,BL$MOD
4117 041372  010002                        MOV     R0,R2                   ; *,NIBBLE
4118 041374  001011                        BNE     6$                      ;                               3297
4119 041376  104422                        TRAP    22                      ;                               3299
4120 041400  062705  000002              ADD     #2,R5                   ; *,OFFSET                      3301
```

```
                                        ;MLX4                                                                   27-Mar-1982 19:24:42    TOPS
                                        ;               NIBBLE GENERATOR                                        27-Mar-1982 19:23:44    PA:<

 4125 041404  005366  000032                    DEC     32(SP)                          ; WRDCNT                                3302
 4126 041410  001003                            BNE     6$                              ;                                       3304
 4127 041412  062706  000012                    ADD     #12,SP                          ;                                       3245
 4128 041416  000207                5$:         RTS     PC                              ;                                       3304
 4129 041420  062706  000012        6$:         ADD     #12,SP                          ;                                       3293
 4130 041424  005303                            DEC     R3                              ; COUNT                                 3292
 4131 041426  001340                            BNE     4$                              ;                                       3281
 4132 041430  000723                            BR      1$                              ;
 4133
 4134                                   ; Routine Size:  50 words
 4135                                   ; Maximum stack depth per invocation:  11 words
 4140
 4141
```

```
;MLX4
;                    GENERATOR FOR OPTION 2                    27-Mar-1982 19:24:42    TOPS-20 Bliss-16 V2(212)
;                                                              27-Mar-1982 19:23:44    PA:<NEALE>MLX4.BLI.5 (18)

;                3313  %sbttl 'GENERATOR FOR OPTION 2'
;                3314  routine GEN2 : novalue =
;                3315      begin
;                3316
;                3317  !++
;                3318  !  ROUTINE:     GEN2
;                3319  !
;                3320  !  PURPOSE:     TO GENERATE THE ENTIRE WRITE BUFFER (WHICH
;                3321  !               WILL BE USED IN CONJUNCTION WITH OPTION 2)
;                3322  !               BASED ON PATTERN TABLE ENTRIES.
;                3323  !--
;                3324
;                3325      local
;                3326          VALUE;
;                3327
;                3328      VALUE = GETCNTS (.PATTERN);
;                3329      FILLER (WBUFF, BUFSIZ, .VALUE);
;                3330      return;
;                3331      end;
;
;                                        .SBTTL  GEN2 GENERATOR FOR OPTION 2
;041432  016746  171240          GEN2:   MOV     PATTERN,-(SP)                   ;                       3328
;041436  004767  177444                  JSR     PC,GETCNTS
;041442  012716  012670                  MOV     #WBUFF,(SP)                     ;                       3329
;041446  012746  004000                  MOV     #4000,-(SP)
;041452  010046                          MOV     R0,-(SP)                        ; VALUE,*
;041454  004767  177606                  JSR     PC,FILLER
;041460  062706  000006                  ADD     #6,SP                           ;                       3314
;041464  000207                          RTS     PC
;
;                                ; Routine Size:  14 words
;                                ; Maximum stack depth per invocation:  3 words
;
;
```

SEQ 0126


```
;MLX4
;                GENERATOR FOR OPTION 3

;                3332  %sbttl 'GENERATOR FOR OPTION 3'
;                3333  routine GEN3 (COMP_FLAG) : novalue =
;                3334      begin
;                3335
;                3336  !++
;                3337  !  ROUTINE:     GEN3(COMP_FLAG)
;                3338  !
;                3339  !  PURPOSE:     TO GENERATE THE ENTIRE WRITE BUFFER WHICH WILL BE MADE
;                3340  !               UP OF WORDS WHICH CONTAIN THE UNIQUE COUNT FROM ONE TO
;                3341  !               BUFSIZ.
;                3342  !
;                3343  !  ARGUMENT:    COMP_FLAG - THIS IS AN INDICATOR OF WHETHER OR NOT TO
;                3344  !                           COMPLEMENT EVERY WORD IN THE WRITE BUFFER.
;                3345  !--
;                3346
;                3347      local
;                3348          OFFSET;
;                3349
;                3350      OFFSET = 0;
;                3351
;                3352      if .COMP_FLAG
;                3353      then                                          !GENERATE COMPLEMENT DATA
;                3354
;                3355          incru COUNT from 1 to BUFSIZ do
;                3356              begin
;                3357              (WBUFF + (.OFFSET)) = not (.COUNT); !COMPLEMENT DATA
;                3358              OFFSET = .OFFSET + 2;
;                3359              end
;                3360
;                3361      else                                          !GENERATE REGULAR DATA
;                3362
;                3363          incru COUNT from 1 to BUFSIZ do
;                3364              begin
;                3365              (WBUFF + (.OFFSET)) = (.COUNT);     !REGULAR DATA
;                3366              OFFSET = .OFFSET + 2;
;                3367              end;
;                3368
;                3369      return;
;                3370      end;
;4236
;4237
                                        .SBTTL  GEN3 GENERATOR FOR OPTION 3
;4241 041466 010146                GEN3: MOV     R1,-(SP)                ;                       3333
;4242 041470 005001                      CLR     R1                      ; OFFSET                3350
;4243 041472 032766 000001 000004        BIT     #1,4(SP)                ; *,COMP.FLAG           3352
;4244 041500 001415                      BEQ     2$
;4245 041502 012700 000001               MOV     #1,R0                   ; *,COUNT               3355
```

```
4247                                    ;MLX4
4248                                    ;                                                    27-Mar-1982 19:24:42    TOPS
4249                                    ;               GENERATOR FOR OPTION 3               27-Mar-1982 19:23:44    PA:<
4250 041506  010061  012670           1$:     MOV     R0,WBUFF(R1)                    ; COUNT,*(OFFSET)                    3357
4251 041512  005161  012670                   COM     WBUFF(R1)                       ; *(OFFSET)
4252 041516  062701  000002                   ADD     #2,R1                           ; *,OFFSET                           3358
4253 041522  005200                           INC     R0                              ; COUNT                              3355
4254 041524  020027  004000                   CMP     R0,#4000                        ; COUNT,*
4255 041530  101766                           BLOS    1$
4256 041532  000412                           BR      4$                              ;                                    3352
4257 041534  012700  000001           2$:     MOV     #1,R0                           ; *,COUNT                            3363
4258 041540  010061  012670           3$:     MOV     R0,WBUFF(R1)                    ; COUNT,*(OFFSET)                    3365
4259 041544  062701  000002                   ADD     #2,R1                           ; *,OFFSET                           3366
4260 041550  005200                           INC     R0                              ; COUNT                              3363
4261 041552  020027  004000                   CMP     R0,#4000                        ; COUNT,*
4262 041556  101770                           BLOS    3$
4263 041560  012601                   4$:     MOV     (SP)+,R1                        ;                                    3333
4264 041562  000207                           RTS     PC
4265
4266                                    ; Routine Size:  31 words
4267                                    ; Maximum stack depth per invocation:  2 words
4272
4273
```

```
;MLX4
;                       GENERATOR FOR OPTION 4                          27-Mar-1982 19:24:42    TOPS-20 Bliss-16 V2(212)
;                                                                       27-Mar-1982 19:23:44    PA:<NEALE>MLX4.BLI.5 (20)

;                3371   %sbttl 'GENERATOR FOR OPTION 4'
;                3372   routine GEN4 (MARPAT, BUFFER) : novalue =
;                3373       begin
;                3374
;                3375   !++
;                3376   !  ROUTINE:     GEN4(MARPAT,BUFFER)
;                3377   !
;                3378   !  PURPOSE:     THIS IS THE GENERATOR WHICH IS USED FOR OPTION 4, THE
;                3379   !               MARCH TEST.  IT BEGINS AT THE START OF THE CHOSEN
;                3380   !               BUFFER AND GENERATES 256 WORDS IN THE EXACT WAY THAT
;                3381   !               OPTION 2 DOES.
;                3382   !
;                3383   !  ARGUMENTS:   (1)  MARPAT - THE MARCH PATTERN NUMBER
;                3384   !
;                3385   !               (2)  BUFFER - THE STARTING ADDRESS OF THE CHOSEN BUFFER
;                3386   !--
;                3387
;                3388       local
;                3389           VALUE;
;                3390
;                3391       VALUE = GETCNTS (.MARPAT);
;                3392       FILLER (.BUFFER, 256, .VALUE);
;                3393       return;
;                3394       end;
4305
4306                                            .SBTTL  GEN4 GENERATOR FOR OPTION 4
4310 041564  016646  000004          GEN4:     MOV     4(SP),-(SP)             ; MARPAT,*              3391
4311 041570  004767  177312                    JSR     PC,GETCNTS
4312 041574  016616  000004                    MOV     4(SP),(SP)              ; BUFFER,*              3392
4313 041600  012746  000400                    MOV     #400,-(SP)
4314 041604  010046                            MOV     R0,-(SP)                ; VALUE,*
4315 041606  004767  177454                    JSR     PC,FILLER
4316 041612  062706  000006                    ADD     #6,SP                   ;                       3372
4317 041616  000207                            RTS     PC
4318
4319                                    ; Routine Size:  14 words
4320                                    ; Maximum stack depth per invocation:  3 words
4325
4326
```

```
4328 ;MLX4
4329 ;                  GENERATOR FOR OPTION 5                  27-Mar-1982 19:24:42    TOPS-20 Bliss-16 V2(212)
4330 ;                                                          27-Mar-1982 19:23:44    PA:<NEALE>MLX4.BLI.5 (21)
4331 ;          3395  %sbttl 'GENERATOR FOR OPTION 5'
4332 ;          3396  routine GEN5 : novalue =
4333 ;          3397      begin
4334 ;          3398
4335 ;          3399  !++
4336 ;          3400  !   ROUTINE:      GEN5
4337 ;          3401  !
4338 ;          3402  !   PURPOSE:      THIS IS THE GENERATOR WHICH IS USED FOR OPTION 5.
4339 ;          3403  !                 IT BEGINS AT THE START OF THE WRITE BUFFER, AND GENERATES
4340 ;          3404  !                 A FULL BUFFER OF RANDOM DATA.  THE FIRST 3 WORDS OF THE BUFFER
4341 ;          3405  !                 RECEIVE THE 3 SEEDS WHICH WERE USED IN THE GENERATION.
4342 ;          3406  !--
4343 ;          3407
4344 ;          3408      local
4345 ;          3409          OFFSET;
4346 ;          3410
4347 ;          3411      (WBUFF + 0) = .SEED1;                   !THE FIRST 3 WORDS OF THE
4348 ;          3412      (WBUFF + 2) = .SEED2;                   !WRITE BUFFER RECEIVE THE
4349 ;          3413      (WBUFF + 4) = .SEED3;                   !3 GENERATING SEEDS.
4350 ;          3414      OFFSET = 6;
4351 ;          3415
4352 ;          3416      incru COUNT from 4 to BUFSIZ do
4353 ;          3417          begin
4354 ;          3418          RN ();                              !NOTE: USING SIGNED VALUES (FULL 16 BITS)
4355 ;          3419          BREAK;
4356 ;          3420          (WBUFF + .OFFSET) = .RANDOM;
4357 ;          3421          OFFSET = .OFFSET + 2;
4358 ;          3422          end;
4359 ;          3423
4360 ;          3424      return;
4361 ;          3425      end;
4365
```

```
 4367
 4371 041620  004167  143436            GEN5:   .SBTTL  GEN5 GENERATOR FOR OPTION 5
 4372 041624  016767  143626  151036            JSR     R1,$SAVE2                ;                                                3396
 4373 041632  016767  143622  151032            MOV     SEED1,WBUFF              ;                                                3411
 4374 041640  016767  143616  151026            MOV     SEED2,WBUFF+2            ;                                                3412
 4375 041646  012701  000006                    MOV     SEED3,WBUFF+4            ;                                                3413
 4376 041652  012702  000004                    MOV     #6,R1                    ; *,OFFSET                                       3414
 4377 041656  004767  143506            1$:     MOV     #4,R2                    ; *,COUNT                                        3416
 4378 041662  104422                            JSR     PC,RN                    ;                                                3418
 4379 041664  016761  143574  012670            TRAP    22
 4380 041672  062701  000002                    MOV     RANDOM,WBUFF(R1)         ; *,*(OFFSET)                                    3420
 4381 041676  005202                            ADD     #2,R1                    ; *,OFFSET                                       3421
 4382 041700  020227  004000                    INC     R2                       ; COUNT                                          3416
 4383 041704  101764                            CMP     R2,#4000                 ; COUNT,*
 4384                                           BLOS    1$
 4385                                   ;MLX4                                                    27-Mar-1982 19:24:42    TOPS
 4386                                   ;               GENERATOR FOR OPTION 5                   27-Mar-1982 19:23:44    PA:<
 4387 041706  000207                            RTS     PC                       ;                                                3396
 4388
 4389                                   ; Routine Size:  28 words
 4390                                   ; Maximum stack depth per invocation:  3 words
 4395
 4396
```

SEQ 0131

```
4398 ;MLX4
4399 ;                  SYSTEM ERROR DETECTOR                    27-Mar-1982 19:24:42    TOPS-20 Bliss-16 V2(212)
4400 ;                                                           27-Mar-1982 19:23:44    PA:<NEALE>MLX4.BLI.5 (22)
4401 ;          3426  %sbttl 'SYSTEM ERROR DETECTOR'
4402 ;          3427  routine SYSERR (LUN) =
4403 ;          3428      begin                                  !* 1 *
4404 ;          3429
4405 ;          3430  !++
4406 ;          3431  !   ROUTINE:     SYSERR(LUN)
4407 ;          3432  !
4408 ;          3433  !   PURPOSE:     (1)  TO SCAN FOR SYSTEM ERRORS AFTER A TRANSFER HAS ENDED.
4409 ;          3434  !
4410 ;          3435  !                (2)  PRINT ERROR MESSAGES IF APPROPRIATE:
4411 ;          3436  !
4412 ;          3437  !                     A) ERROR MESSAGES DURING RETRIES WILL NOT
4413 ;          3438  !                        BE PRINTED.
4414 ;          3439  !
4415 ;          3440  !                     B) THE OPERATOR HAS THE ABILITY TO INHIBIT THE
4416 ;          3441  !                        PRINTING OF DATA ERRORS, BUT NOT SYSTEM ERRORS.
4417 ;          3442  !
4418 ;          3443  !                (3)  DECIDE ON A VALUE TO RETURN WHICH WILL INDICATE THE
4419 ;          3444  !                     RELATIVE SEVERITY OF THE ERROR.
4420 ;          3445  !
4421 ;          3446  !   ARGUMENT:    LUN = THE CURRENT LOGICAL UNIT NUMBER
4422 ;          3447  !                      [0 TO NUMBER OF DRIVES MINUS 1].
4423 ;          3448  !
4424 ;          3449  !   RESULT:      VALUE RETURNED FOR THE SUBROUTINE SHOWS TYPE OF ERROR:
4425 ;          3450  !                        0 = NO ERRORS AT ALL
4426 ;          3451  !                        1 = RETRY ALLOWED FOR THESE TYPES OF ERRORS
4427 ;          3452  !                        2 = FATAL CONTROLLER ERROR
4428 ;          3453  !                        3 = FATAL DRIVE ERROR
4429 ;          3454  !                        4 = UNCORRECTABLE DATA ERROR
4430 ;          3455  !                        5 = CORRECTABLE DATA ERROR
4431 ;          3456  !
4432 ;          3457  !   FIGURE OUT WHICH VALUE TO RETURN BY THE FOLLOWING SCHEME:
4433 ;          3458  !
4434 ;          3459  !   (1)  IF THERE IS A CHOICE BETWEEN DATA ERROR AND SYSTEM ERROR,
4435 ;          3460  !        FIRST CHOOSE THE DATA ERROR.
4436 ;          3461  !   (2)  IF THERE IS A CHOICE BETWEEN FATAL AND NON-FATAL ERROR,
4437 ;          3462  !        FIRST CHOOSE THE FATAL ERROR.
4438 ;          3463  !   (3)  IF THERE IS A CHOICE BETWEEN CONTROLLER ERROR AND DRIVE ERROR,
4439 ;          3464  !        FIRST CHOOSE CONTROLLER ERROR.
4440 ;          3465  !--
4441 ;          3466
4442 ;          3467      label
4443 ;          3468          PURPOSE_2_CODE;
4444 ;          3469
4445 ;          3470      local
4446 ;          3471          ERR2,
4447 ;          3472          ERR3,
4448 ;          3473          ERR4,
4449 ;          3474          ERR5,
4450 ;          3475          ERR6,
4451 ;          3476          ERR7,
4452 ;          3477          ERR8,                              ! VER CZMLBB ADDED THIS LOCAL
```

```
3478          ERR9;                                   ! VER CZMLBB ADDED THIS LOCAL
3479
3480      ERR2 = INACTIVE;
3481      ERR3 = INACTIVE;
3482      ERR4 = INACTIVE;
3483      ERR5 = INACTIVE;
3484      ERR6 = INACTIVE;
3485      ERR7 = INACTIVE;
3486      ERR8 = INACTIVE;                            ! VER CZMLBB
3487      ERR9 = INACTIVE;                            ! VER CZMLBB
3488
3489      !+
3490      ! THIS IS THE CODE FOR PURPOSE 1:
3491      !-
3492
3493      if .SC
3494      then                                        !THERE ARE SOME ERRORS -- FIND OUT WHAT KIND
3495          begin                                   !* 2 *
3496
3497          if ((.NED) or (.NEM) or (.PGE)) then ERR2 = ACTIVE;
3498
3499          if ((.DLT) or (.PE) or (.MXF) or (.MDPE) or (.MCPE)) then ERR3 = ACTIVE;
3500
3501  ! VER ADDED THIS ERR9
3502
3503          if (.MDPE) and ( not .SGL) then ERR9 = ACTIVE;
3504
3505  ! VER ADDED THIS ERR8
3506
3507          if ((.DLT) or (.PE) or (.MXF) or (.MCPE)) then ERR8 = ACTIVE;
3508
3509          if ((.WCE) and (.ECCDIS eql 0)) then ERR3 = ACTIVE;
3510
3511          if ((.UNS) or (.IAE) or (.AOE) or (.RMR) or (.ILR) or (.ILF)) then ERR4 = ACTIVE;
3512
3513          if ((.OPI) or (.DPAR) or (.CPAR)) then ERR5 = ACTIVE;
3514
3515          if ((.DCK) or (.crc) or (.SGL)) then ERR7 = ACTIVE;
3516
3517          if ((.WCE) and (.ECCDIS eql 1)) then ERR7 = ACTIVE;
3518
3519          if ((.ECH) or (.UNC)) then ERR6 = ACTIVE;
3520
3521      end                                         !* 2 *
3522  else                                            !THERE ARE NO ERRORS -- RETURN SUCCESSFULLY
3523      return 0;                                   !NO ERRORS AT ALL
3524
3525  !+
3526  ! THIS IS THE CODE FOR PURPOSE 2:
3527  !
3528  !     LABEL:
3529  !        BEGIN 3
```

```
4510 ;MLX4
4511 ;                  SYSTEM ERROR DETECTOR
4512
4513 ;          3530       !       IF RETRYING
4514 ;          3531       !       THEN  LEAVE LABEL (NO PRINTOUTS AT ALL)
4515 ;          3532       !       ELSE
4516 ;          3533       !          BEGIN 4
4517 ;          3534       !          IF ERROR PRINTOUTS ARE ALLOWED
4518 ;          3535       !          THEN
4519 ;          3536       !             BEGIN 5A
4520 ;          3537       !             IDENTIFY THE LOGICAL UNIT
4521 ;          3538       !             SCAN FOR & PRINT DATA ERRORS
4522 ;          3539       !             END   5A
4523 ;          3540       !          ELSE
4524 ;          3541       !             BEGIN 5B
4525 ;          3542       !             IF THERE ARE SYSTEM ERRORS TO REPORT
4526 ;          3543       !             THEN  IDENTIFY THE LOGICAL UNIT
4527 ;          3544       !             ELSE  LEAVE LABEL
4528 ;          3545       !             END   5B
4529 ;          3546       !          SCAN FOR & PRINT SYSTEM ERRORS
4530 ;          3547       !          END  4
4531 ;          3548       !       END  3
4532 ;          3549       !-
4533 ;          3550
4534 ;          3551  PURPOSE_2_CODE :
4535 ;          3552      begin                                   !* 3 *
4536 ;          3553
4537 ;          3554      if .RETRYING
4538 ;          3555      then
4539 ;          3556          leave PURPOSE_2_CODE                !(NO PRINTOUTS AT ALL)
4540 ;          3557      else
4541 ;          3558          begin                               !* 4 *
4542 ;          3559
4543 ;          3560          if .ERROUT
4544 ;          3561          then
4545 ;          3562              begin                           !* 5A *
4546 ;          3563              SAYWHO (.LUN);
4547 ;          3564              PRINTB (SAY1, PHR10);
4548 ;          3565              !'ERROR BITS SET:
4549 ;          3566
4550 ;          3567              if .ERR7
4551 ;          3568              then
4552 ;          3569                  begin
4553 ;          3570
4554 ;          3571                  if .DCK then PRINTB (FMT15, MLB20);
4555 ;          3572
4556 ;          3573                  if .crc then PRINTB (FMT15, MLB22);
4557 ;          3574
4558 ;          3575                  if .SGL then PRINTB (FMT15, MLB23);
4559 ;          3576
4560 ;          3577                  if ((.WCE) and (.ECCDIS eql 1)) then PRINTB (FMT15, MLB6);
4561 ;          3578
4562 ;          3579                  end;
4563 ;          3580
4564 ;          3581              if .ERR6
```

```
3582                then
3583                    begin
3584
3585                    if .ECH then PRINTB (FMT15, MLB21);
3586
3587                    if .UNC then PRINTB (FMT15, MLB24);
3588
3589                    end;
3590
3591                end                                         !* 5A *
3592            else
3593                begin                                       !* 5B *
3594    ! VER CZMLBB CHANGED ERR3 TO ERR8
3595
3596                if ((.ERR2) or (.ERR8) or (.ERR4) or (.ERR5))
3597                then
3598                    begin
3599                    SAYWHO (.LUN);
3600                    PRINTB (SAY1, PHR10);
3601                    !'ERROR BITS SET:
3602                    end
3603                else
3604    ! VER CZMLBB ADDED TEST OF ERR9 HERE
3605
3606                    if .ERR9
3607                    then
3608                        begin
3609                        SAYWHO (.LUN);
3610                        PRINTB (SAY1, PHR10);
3611                        !'ERROR BITS SET:
3612                        end
3613                    else
3614                        begin
3615                        leave PURPOSE_2_CODE;
3616                        end;
3617
3618                end;                                        !* 5B *
3619
3620            if .ERR2
3621            then
3622                begin
3623
3624                if .NED then PRINTB (FMT15, MLB2);
3625
3626                if .NEM then PRINTB (FMT15, MLB3);
3627
3628                if .PGE then PRINTB (FMT15, MLB4);
3629
3630                end;
3631
3632            if .ERR3
3633            then
```

```
                    begin

                    if .DLT then PRINTB (FMT15, MLB5);

                    if ((.WCE) and (.ECCDIS eql 0)) then PRINTB (FMT15, MLB6);

                    if .PE then PRINTB (FMT15, MLB7);

                    if .MXF then PRINTB (FMT15, MLB8);

! VER CZMLBB ADDED 'AND NOT .SGL'

                    if ((.MDPE) and ( not .SGL)) then PRINTB (FMT15, MLB9);

                    if .MCPE then PRINTB (FMT15, MLB10);

                    end;

                if .ERR4
                then
                    begin

                    if .UNS then PRINTB (FMT15, MLB11);

                    if .IAE then PRINTB (FMT15, MLB12);

                    if .AOE then PRINTB (FMT15, MLB13);

                    if .RMR then PRINTB (FMT15, MLB14);

                    if .ILR then PRINTB (FMT15, MLB15);

                    if .ILF then PRINTB (FMT15, MLB16);

                    end;

                if .ERR5
                then
                    begin

                    if .OPI then PRINTB (FMT15, MLB17);

                    if .DPAR then PRINTB (FMT15, MLB18);

                    if .CPAR then PRINTB (FMT15, MLB19);

                    end;

                end;                                        !* 4 *

            end;                                            !* 3 *
```

```
 4678 ;MLX4
 4679 ;                  SYSTEM ERROR DETECTOR                          27-Mar-1982 19:24:42    TOPS-20 Bliss-16 V2(212)
 4680 ;                                                                 27-Mar-1982 19:23:44    PA:<NEALE>MLX4.BLI.5 (22)
 4681 ;          3686        !+
 4682 ;          3687        ! THIS IS THE CODE FOR PURPOSE 3:
 4683 ;          3688        !-
 4684 ;          3689
 4685 ;          3690        if .ERR6                                   !UNRECOVERABLE DATA ERROR (NO RETRY ALLOWED)
 4686 ;          3691        then
 4687 ;          3692            return 4;
 4688 ;          3693
 4689 ;          3694        if .ERR7                                   !RECOVERABLE DATA ERROR (RETRY TO CLASSIFY)
 4690 ;          3695        then
 4691 ;          3696            return 5;
 4692 ;          3697
 4693 ;          3698        if .ERR2                                   !CONTROLLER FATAL ERROR
 4694 ;          3699        then
 4695 ;          3700            return 2;
 4696 ;          3701
 4697 ;          3702        if .ERR4                                   !DRIVE FATAL ERROR
 4698 ;          3703        then
 4699 ;          3704            return 3;
 4700 ;          3705
 4701 ;          3706        if (.ERR3 or .ERR5)                        !NON-FATAL ERRORS
 4702 ;          3707        then
 4703 ;          3708            return 1
 4704 ;          3709        else
 4705 ;          3710            begin
 4706 ;          3711
 4707 ;          3712            if ((.ERROUT) and ( not .RETRYING)) then PRINTB (SAY1, PHR11);
 4708 ;          3713
 4709 ;          3714            !'SC SET BUT NO SYSTEM ERRORS FOUND'
 4710 ;          3715            return 1;
 4711 ;          3716            end;
 4712 ;          3717
 4713 ;          3718        end;                                       !* 1 *
 4717
 4718
                                        .SBTTL  SYSERR SYSTEM ERROR DETECTOR
 4722 041710 004167 143420      SYSERR: JSR     R1,$SAVE5                  ;                                       3427
 4723 041714 005746                     TST     -(SP)
 4724 041716 005046                     CLR     -(SP)                      ; ERR2                                  3480
 4725 041720 005046                     CLR     -(SP)                      ; ERR3                                  3481
 4726 041722 005001                     CLR     R1                         ; ERR4                                  3482
 4727 041724 005002                     CLR     R2                         ; ERR5                                  3483
 4728 041726 005066 000004              CLR     4(SP)                      ; ERR6                                  3484
 4729 041732 005003                     CLR     R3                         ; ERR7                                  3485
 4730 041734 005004                     CLR     R4                         ; ERR8                                  3486
 4731 041736 005005                     CLR     R5                         ; ERR9                                  3487
 4732 041740 005777 172402              TST     @ML.REG                    ;                                       3493
 4733                           ;MLX4
 4734                           ;               SYSTEM ERROR DETECTOR                  27-Mar-1982 19:24:42    TOPS
 4735                                                                                  27-Mar-1982 19:23:44    PA:<
 4736 041744 100402                     BMI     1$
 4737 041746 000167 002344              JMP     53$
 4738 041752 032777 010000 172376 1$:   BIT     #10000,@ML.REG+10          ;                                       3497
 4739 041760 001010                     BNE     2$
 4740 041762 032777 004000 172366       BIT     #4000,@ML.REG+10
```

```
 4741 041770 001004                          BNE     2$
 4742 041772 032777 002000 172356            BIT     #2000,@ML.REG+10
 4743 042000 001403                          BEQ     3$
 4744 042002 012766 000001 000002 2$:        MOV     #1,2(SP)            ; *,ERR2
 4745 042010 005777 172342        3$:        TST     @ML.REG+10          ;                    3499
 4746 042014 100420                          BMI     4$
 4747 042016 032777 020000 172332            BIT     #20000,@ML.REG+10
 4748 042024 001014                          BNE     4$
 4749 042026 032777 001000 172322            BIT     #1000,@ML.REG+10
 4750 042034 001010                          BNE     4$
 4751 042036 032777 000400 172312            BIT     #400,@ML.REG+10
 4752 042044 001004                          BNE     4$
 4753 042046 032777 020000 172272            BIT     #20000,@ML.REG
 4754 042054 001402                          BEQ     5$
 4755 042056 012716 000001        4$:        MOV     #1,(SP)             ; *,ERR3
 4756 042062 032777 000400 172266 5$:        BIT     #400,@ML.REG+10     ;                    3503
 4757 042070 001406                          BEQ     6$
 4758 042072 032777 040000 172310            BIT     #40000,@ML.REG+42
 4759 042100 001002                          BNE     6$
 4760 042102 012705 000001                   MOV     #1,R5               ; *,ERR9
 4761 042106 005777 172244        6$:        TST     @ML.REG+10          ;                    3507
 4762 042112 100414                          BMI     7$
 4763 042114 032777 020000 172234            BIT     #20000,@ML.REG+10
 4764 042122 001010                          BNE     7$
 4765 042124 032777 001000 172224            BIT     #1000,@ML.REG+10
 4766 042132 001004                          BNE     7$
 4767 042134 032777 020000 172204            BIT     #20000,@ML.REG
 4768 042142 001402                          BEQ     8$
 4769 042144 012704 000001        7$:        MOV     #1,R4               ; *,ERR8
 4770 042150 032777 040000 172200 8$:        BIT     #40000,@ML.REG+10   ;                    3509
 4771 042156 001405                          BEQ     9$
 4772 042160 005767 140070                   TST     ECCDIS
 4773 042164 001002                          BNE     9$
 4774 042166 012716 000001                   MOV     #1,(SP)             ; *,ERR3
 4775 042172 032777 040000 172162 9$:        BIT     #40000,@ML.REG+14   ;                    3511
 4776 042200 001024                          BNE     10$
 4777 042202 032777 002000 172152            BIT     #2000,@ML.REG+14
 4778 042210 001020                          BNE     10$
 4779 042212 032777 001000 172142            BIT     #1000,@ML.REG+14
 4780 042220 001014                          BNE     10$
 4781 042222 132777 000004 172132            BITB    #4,@ML.REG+14
 4782 042230 001010                          BNE     10$
 4783 042232 132777 000002 172122            BITB    #2,@ML.REG+14
 4784 042240 001004                          BNE     10$
 4785 042242 132777 000001 172112            BITB    #1,@ML.REG+14
 4786 042250 001402                          BEQ     11$
 4787 042252 012701 000001        10$:       MOV     #1,R1               ; *,ERR4
 4788                              ;MLX4
 4789                              ;                                                27-Mar-1982 19:24:42  TOPS
 4790                                         SYSTEM ERROR DETECTOR                27-Mar-1982 19:23:44  PA:<
 4791 042256 032777 020000 172076 11$:       BIT     #20000,@ML.REG+14   ;                    3513
 4792 042264 001010                          BNE     12$
 4793 042266 132777 000040 172066            BITB    #40,@ML.REG+14
 4794 042274 001004                          BNE     12$
 4795 042276 132777 000010 172056            BITB    #10,@ML.REG+14
 4796 042304 001402                          BEQ     13$
 4797 042306 012702 000001        12$:       MOV     #1,R2               ; *,ERR5
```

SEQ 0138

```
4798 042312 005777 172044          13$:  TST   @ML.REG+14                ;                    3515
4799 042316 100410                       BMI   14$
4800 042320 032777 020000 172062         BIT   #20000,@ML.REG+42
4801 042326 001004                       BNE   14$
4802 042330 032777 040000 172052         BIT   #40000,@ML.REG+42
4803 042336 001402                       BEQ   15$
4804 042340 012703 000001          14$:  MOV   #1,R3                      ; *,ERR7
4805 042344 032777 040000 172004   15$:  BIT   #40000,@ML.REG+10          ;                    3517
4806 042352 001406                       BEQ   16$
4807 042354 026727 137674 000001         CMP   ECCDIS,#1
4808 042362 001002                       BNE   16$
4809 042364 012703 000001                MOV   #1,R3                      ; *,ERR7
4810 042370 132777 000100 171764   16$:  BITB  #100,@ML.REG+14            ;                    3519
4811 042376 001003                       BNE   17$
4812 042400 005777 172004                TST   @ML.REG+42
4813 042404 100003                       BPL   18$
4814 042406 012766 000001 000004   17$:  MOV   #1,4(SP)                   ; *,ERR6
4815 042414 032767 000001 171716   18$:  BIT   #1,RETRYING                ;                    3554
4816 042422 001402                       BEQ   19$                        ;                    3556
4817 042424 000167 001520                JMP   47$
4818 042430 032767 000001 137622   19$:  BIT   #1,ERROUT                  ;                    3560
4819 042436 001552                       BEQ   25$
4820 042440 016646 000024                MOV   24(SP),-(SP)               ; LUN,*              3563
4821 042444 004767 173054                JSR   PC,SAYWHO
4822 042450 012716 010120                MOV   #PHR10,(SP)                ;                    3564
4823 042454 012746 007072                MOV   #SAY1,-(SP)
4824 042460 012746 000002                MOV   #2,-(SP)
4825 042464 010600                       MOV   SP,R0
4826 042466 104414                       TRAP  14                         ; SP,*
4827 042470 032703 000001                BIT   #1,R3                      ; *,ERR7             3567
4828 042474 001473                       BEQ   23$
4829 042476 005777 171660                TST   @ML.REG+14                 ;                    3571
4830 042502 100012                       BPL   20$
4831 042504 012746 007550                MOV   #MLB20,-(SP)
4832 042510 012746 007060                MOV   #FMT15,-(SP)
4833 042514 012746 000002                MOV   #2,-(SP)
4834 042520 010600                       MOV   SP,R0
4835 042522 104414                       TRAP  14                         ; SP,*
4836 042524 062706 000006                ADD   #6,SP
4837 042530 032777 020000 171652   20$:  BIT   #20000,@ML.REG+42          ;                    3573
4838 042536 001412                       BEQ   21$
4839 042540 012746 007560                MOV   #MLB22,-(SP)
4840 042544 012746 007060                MOV   #FMT15,-(SP)
4841 042550 012746 000002                MOV   #2,-(SP)
4842 042554 010600                       MOV   SP,R0                      ; SP,*
4843                               ;MLX4
4844                               ;     SYSTEM ERROR DETECTOR                    27-Mar-1982 19:24:42  TOPS
4845                                                                              27-Mar-1982 19:23:44  PA:<
4846 042556 104414                       TRAP  14
4847 042560 062706 000006                ADD   #6,SP
4848 042564 032777 040000 171616   21$:  BIT   #40000,@ML.REG+42          ;                    3575
4849 042572 001412                       BEQ   22$
4850 042574 012746 007564                MOV   #MLB23,-(SP)
4851 042600 012746 007060                MOV   #FMT15,-(SP)
4852 042604 012746 000002                MOV   #2,-(SP)
4853 042610 010600                       MOV   SP,R0                      ; SP,*
4854 042612 104414                       TRAP  14
```

```
 4855 042614  062706  000006                    ADD     #6,SP
 4856 042620  032777  040000  171530  22$:      BIT     #40000,@ML.REG+10       ;                       3577
 4857 042626  001416                            BEQ     23$
 4858 042630  026727  137420  000001            CMP     ECCDIS,#1
 4859 042636  001012                            BNE     23$
 4860 042640  012746  007450                    MOV     #MLB6,-(SP)
 4861 042644  012746  007060                    MOV     #FMT15,-(SP)
 4862 042650  012746  000002                    MOV     #2,-(SP)
 4863 042654  010600                            MOV     SP,R0                   ; SP,*
 4864 042656  104414                            TRAP    14
 4865 042660  062706  000006                    ADD     #6,SP
 4866 042664  032766  000001  000012  23$:      BIT     #1,12(SP)               ; *,ERR6                3581
 4867 042672  001505                            BEQ     29$
 4868 042674  132777  000100  171460            BITB    #100,@ML.REG+14         ;                       3585
 4869 042702  001412                            BEQ     24$
 4870 042704  012746  007554                    MOV     #MLB21,-(SP)
 4871 042710  012746  007060                    MOV     #FMT15,-(SP)
 4872 042714  012746  000002                    MOV     #2,-(SP)
 4873 042720  010600                            MOV     SP,R0                   ; SP,*
 4874 042722  104414                            TRAP    14
 4875 042724  062706  000006                    ADD     #6,SP
 4876 042730  005777  171454          24$:      TST     @ML.REG+42              ;                       3587
 4877 042734  100064                            BPL     29$
 4878 042736  012746  007570                    MOV     #MLB24,-(SP)
 4879 042742  012746  007060                    MOV     #FMT15,-(SP)
 4880 042746  012746  000002                    MOV     #2,-(SP)
 4881 042752  010600                            MOV     SP,R0                   ; SP,*
 4882 042754  104414                            TRAP    14
 4883 042756  062706  000006                    ADD     #6,SP
 4884 042762  000451                            BR      29$                     ;                       3562
 4885 042764  032766  000001  000002  25$:      BIT     #1,2(SP)                ; *,ERR2                3596
 4886 042772  001010                            BNE     26$
 4887 042774  006004                            ROR     R4                      ; ERR8
 4888 042776  103406                            BLO     26$
 4889 043000  032701  000001                    BIT     #1,R1                   ; *,ERR4
 4890 043004  001003                            BNE     26$
 4891 043006  032702  000001                    BIT     #1,R2                   ; *,ERR5
 4892 043012  001415                            BEQ     27$
 4893 043014  016646  000024          26$:      MOV     24(SP),-(SP)            ; LUN,*                 3599
 4894 043020  004767  172500                    JSR     PC,SAYWHO
 4895 043024  012716  010120                    MOV     #PHR10,(SP)             ;                       3600
 4896 043030  012746  007072                    MOV     #SAY1,-(SP)
 4897 043034  012746  000002                    MOV     #2,-(SP)
 4898                         ;MLX4
 4899                         ;                  SYSTEM ERROR DETECTOR                  27-Mar-1982 19:24:42    TOPS
 4900                                                                                    27-Mar-1982 19:23:44    PA:<
 4901 043040  010600                            MOV     SP,R0                   ; SP,*
 4902 043042  104414                            TRAP    14
 4903 043044  000420                            BR      29$                     ;                       3598
 4904 043046  006005          27$:      ROR     R5                      ; ERR9                  3606
 4905 043050  103402                            BLO     28$
 4906 043052  000167  001072                    JMP     47$
 4907 043056  016646  000024          28$:      MOV     24(SP),-(SP)            ; LUN,*                 3609
 4908 043062  004767  172436                    JSR     PC,SAYWHO
 4909 043066  012716  010120                    MOV     #PHR10,(SP)             ;                       3610
 4910 043072  012746  007072                    MOV     #SAY1,-(SP)
 4911 043076  012746  000002                    MOV     #2,-(SP)
```

```
4912 043102 010600                          MOV   SP,R0                    ; SP,*
4913 043104 104414                          TRAP  14
4914 043106 062706 000006          29$:     ADD   #6,SP                    ;                    3608
4915 043112 032766 000001 000002            BIT   #1,2(SP)                 ; *,ERR2             3620
4916 043120 001452                          BEQ   32$
4917 043122 032777 010000 171226            BIT   #10000,@ML.REG+10        ;                    3624
4918 043130 001412                          BEQ   30$
4919 043132 012746 007430                   MOV   #MLB2,-(SP)
4920 043136 012746 007060                   MOV   #FMT15,-(SP)
4921 043142 012746 000002                   MOV   #2,-(SP)
4922 043146 010600                          MOV   SP,R0                    ; SP,*
4923 043150 104414                          TRAP  14
4924 043152 062706 000006                   ADD   #6,SP
4925 043156 032777 004000 171172   30$:     BIT   #4000,@ML.REG+10         ;                    3626
4926 043164 001412                          BEQ   31$
4927 043166 012746 007434                   MOV   #MLB3,-(SP)
4928 043172 012746 007060                   MOV   #FMT15,-(SP)
4929 043176 012746 000002                   MOV   #2,-(SP)
4930 043202 010600                          MOV   SP,R0                    ; SP,*
4931 043204 104414                          TRAP  14
4932 043206 062706 000006                   ADD   #6,SP
4933 043212 032777 002000 171136   31$:     BIT   #2000,@ML.REG+10         ;                    3628
4934 043220 001412                          BEQ   32$
4935 043222 012746 007440                   MOV   #MLB4,-(SP)
4936 043226 012746 007060                   MOV   #FMT15,-(SP)
4937 043232 012746 000002                   MOV   #2,-(SP)
4938 043236 010600                          MOV   SP,R0                    ; SP,*
4939 043240 104414                          TRAP  14
4940 043242 062706 000006                   ADD   #6,SP
4941 043246 032716 000001          32$:     BIT   #1,(SP)                  ; *,ERR3             3632
4942 043252 001532                          BEQ   38$
4943 043254 005777 171076                   TST   @ML.REG+10               ;                    3636
4944 043260 100012                          BPL   33$
4945 043262 012746 007444                   MOV   #MLB5,-(SP)
4946 043266 012746 007060                   MOV   #FMT15,-(SP)
4947 043272 012746 000002                   MOV   #2,-(SP)
4948 043276 010600                          MOV   SP,R0                    ; SP,*
4949 043300 104414                          TRAP  14
4950 043302 062706 000006                   ADD   #6,SP
4951 043306 032777 040000 171042   33$:     BIT   #40000,@ML.REG+10        ;                    3638
4952 043314 001415                          BEQ   34$
4953                               ;MLX4
4954                               ;        SYSTEM ERROR DETECTOR          27-Mar-1982 19:24:42  TOPS
4955                                                                       27-Mar-1982 19:23:44  PA:<
4956 043316 005767 136732                   TST   ECCDIS
4957 043322 001012                          BNE   34$
4958 043324 012746 007450                   MOV   #MLB6,-(SP)
4959 043330 012746 007060                   MOV   #FMT15,-(SP)
4960 043334 012746 000002                   MOV   #2,-(SP)
4961 043340 010600                          MOV   SP,R0                    ; SP,*
4962 043342 104414                          TRAP  14
4963 043344 062706 000006                   ADD   #6,SP
4964 043350 032777 020000 171000   34$:     BIT   #20000,@ML.REG+10        ;                    3640
4965 043356 001412                          BEQ   35$
4966 043360 012746 007454                   MOV   #MLB7,-(SP)
4967 043364 012746 007060                   MOV   #FMT15,-(SP)
4968 043370 012746 000002                   MOV   #2,-(SP)
```

```
 4969 043374  010600                          MOV     SP,R0                   ; SP,*
 4970 043376  104414                          TRAP    14
 4971 043400  062706  000006                  ADD     #6,SP
 4972 043404  032777  001000  170744  35$:    BIT     #1000,@ML.REG+10        ;                       3642
 4973 043412  001412                          BEQ     36$
 4974 043414  012746  007460                  MOV     #MLB8,-(SP)
 4975 043420  012746  007060                  MOV     #FMT15,-(SP)
 4976 043424  012746  000002                  MOV     #2,-(SP)
 4977 043430  010600                          MOV     SP,R0                   ; SP,*
 4978 043432  104414                          TRAP    14
 4979 043434  062706  000006                  ADD     #6,SP
 4980 043440  032777  000400  170710  36$:    BIT     #400,@ML.REG+10         ;                       3646
 4981 043446  001416                          BEQ     37$
 4982 043450  032777  040000  170732          BIT     #40000,@ML.REG+42
 4983 043456  001012                          BNE     37$
 4984 043460  012746  007464                  MOV     #MLB9,-(SP)
 4985 043464  012746  007060                  MOV     #FMT15,-(SP)
 4986 043470  012746  000002                  MOV     #2,-(SP)
 4987 043474  010600                          MOV     SP,R0                   ; SP,*
 4988 043476  104414                          TRAP    14
 4989 043500  062706  000006                  ADD     #6,SP
 4990 043504  032777  020000  170634  37$:    BIT     #20000,@ML.REG          ;                       3648
 4991 043512  001412                          BEQ     38$
 4992 043514  012746  007472                  MOV     #MLB10,-(SP)
 4993 043520  012746  007060                  MOV     #FMT15,-(SP)
 4994 043524  012746  000002                  MOV     #2,-(SP)
 4995 043530  010600                          MOV     SP,R0                   ; SP,*
 4996 043532  104414                          TRAP    14
 4997 043534  062706  000006                  ADD     #6,SP
 4998 043540  032701  000001          38$:    BIT     #1,R1                   ; *,ERR4                3652
 4999 043544  001524                          BEQ     44$
 5000 043546  032777  040000  170606          BIT     #40000,@ML.REG+14       ;                       3656
 5001 043554  001412                          BEQ     39$
 5002 043556  012746  007500                  MOV     #MLB11,-(SP)
 5003 043562  012746  007060                  MOV     #FMT15,-(SP)
 5004 043566  012746  000002                  MOV     #2,-(SP)
 5005 043572  010600                          MOV     SP,R0                   ; SP,*
 5006 043574  104414                          TRAP    14
 5007 043576  062706  000006                  ADD     #6,SP
 5008                         ;MLX4
 5009                         ;               SYSTEM ERROR DETECTOR                   27-Mar-1982 19:24:42    TOPS
 5010                                                                                 27-Mar-1982 19:23:44    PA:<
 5011 043602  032777  002000  170552  39$:    BIT     #2000,@ML.REG+14        ;                       3658
 5012 043610  001412                          BEQ     40$
 5013 043612  012746  007504                  MOV     #MLB12,-(SP)
 5014 043616  012746  007060                  MOV     #FMT15,-(SP)
 5015 043622  012746  000002                  MOV     #2,-(SP)
 5016 043626  010600                          MOV     SP,R0                   ; SP,*
 5017 043630  104414                          TRAP    14
 5018 043632  062706  000006                  ADD     #6,SP
 5019 043636  032777  001000  170516  40$:    BIT     #1000,@ML.REG+14        ;                       3660
 5020 043644  001412                          BEQ     41$
 5021 043646  012746  007510                  MOV     #MLB13,-(SP)
 5022 043652  012746  007060                  MOV     #FMT15,-(SP)
 5023 043656  012746  000002                  MOV     #2,-(SP)
 5024 043662  010600                          MOV     SP,R0                   ; SP,*
 5025 043664  104414                          TRAP    14
```

```
5026 043666  062706  000006                 ADD     #6,SP
5027 043672  132777  000004  170462  41$:   BITB    #4,@ML.REG+14          ;                    3662
5028 043700  001412                         BEQ     42$
5029 043702  012746  007514                 MOV     #MLB14,-(SP)
5030 043706  012746  007060                 MOV     #FMT15,-(SP)
5031 043712  012746  000002                 MOV     #2,-(SP)
5032 043716  010600                         MOV     SP,R0                  ; SP,*
5033 043720  104414                         TRAP    14
5034 043722  062706  000006                 ADD     #6,SP
5035 043726  132777  000002  170426  42$:   BITB    #2,@ML.REG+14          ;                    3664
5036 043734  001412                         BEQ     43$
5037 043736  012746  007520                 MOV     #MLB15,-(SP)
5038 043742  012746  007060                 MOV     #FMT15,-(SP)
5039 043746  012746  000002                 MOV     #2,-(SP)
5040 043752  010600                         MOV     SP,R0                  ; SP,*
5041 043754  104414                         TRAP    14
5042 043756  062706  000006                 ADD     #6,SP
5043 043762  132777  000001  170372  43$:   BITB    #1,@ML.REG+14          ;                    3666
5044 043770  001412                         BEQ     44$
5045 043772  012746  007524                 MOV     #MLB16,-(SP)
5046 043776  012746  007060                 MOV     #FMT15,-(SP)
5047 044002  012746  000002                 MOV     #2,-(SP)
5048 044006  010600                         MOV     SP,R0                  ; SP,*
5049 044010  104414                         TRAP    14
5050 044012  062706  000006                 ADD     #6,SP
5051 044016  032702  000001          44$:   BIT     #1,R2                  ; *,ERR5             3670
5052 044022  001452                         BEQ     47$
5053 044024  032777  020000  170330         BIT     #20000,@ML.REG+14      ;                    3674
5054 044032  001412                         BEQ     45$
5055 044034  012746  007530                 MOV     #MLB17,-(SP)
5056 044040  012746  007060                 MOV     #FMT15,-(SP)
5057 044044  012746  000002                 MOV     #2,-(SP)
5058 044050  010600                         MOV     SP,R0                  ; SP,*
5059 044052  104414                         TRAP    14
5060 044054  062706  000006                 ADD     #6,SP
5061 044060  132777  000040  170274  45$:   BITB    #40,@ML.REG+14         ;                    3676
5062 044066  001412                         BEQ     46$
5063                         ;MLX4
5064                         ;              SYSTEM ERROR DETECTOR                  27-Mar-1982 19:24:42  TOPS
5065                                                                                27-Mar-1982 19:23:44  PA:<
5066 044070  012746  007534                 MOV     #MLB18,-(SP)
5067 044074  012746  007060                 MOV     #FMT15,-(SP)
5068 044100  012746  000002                 MOV     #2,-(SP)
5069 044104  010600                         MOV     SP,R0                  ; SP,*
5070 044106  104414                         TRAP    14
5071 044110  062706  000006                 ADD     #6,SP
5072 044114  132777  000010  170240  46$:   BITB    #10,@ML.REG+14         ;                    3678
5073 044122  001412                         BEQ     47$
5074 044124  012746  007542                 MOV     #MLB19,-(SP)
5075 044130  012746  007060                 MOV     #FMT15,-(SP)
5076 044134  012746  000002                 MOV     #2,-(SP)
5077 044140  010600                         MOV     SP,R0                  ; SP,*
5078 044142  104414                         TRAP    14
5079 044144  062706  000006                 ADD     #6,SP
5080 044150  032766  000001  000004  47$:   BIT     #1,4(SP)               ; *,ERR6             3690
5081 044156  001403                         BEQ     48$
5082 044160  012700  000004                 MOV     #4,R0                  ;                    3692
```

```
 5083 044164  000455                        BR      54$
 5084 044166  006003                48$:    ROR     R3                      ; ERR7                  3694
 5085 044170  103003                        BCC     49$
 5086 044172  012700  000005                MOV     #5,R0                   ;                       3696
 5087 044176  000450                        BR      54$
 5088 044200  032766  000001  000002  49$:   BIT     #1,2(SP)                ; *,ERR2                3698
 5089 044206  001403                        BEQ     50$
 5090 044210  012700  000002                MOV     #2,R0                   ;                       3700
 5091 044214  000441                        BR      54$
 5092 044216  006001                50$:    ROR     R1                      ; ERR4                  3702
 5093 044220  103003                        BCC     51$
 5094 044222  012700  000003                MOV     #3,R0                   ;                       3704
 5095 044226  000434                        BR      54$
 5096 044230  032716  000001        51$:    BIT     #1,(SP)                 ; *,ERR3                3706
 5097 044234  001024                        BNE     52$
 5098 044236  006002                        ROR     R2                      ; ERR5
 5099 044240  103422                        BLO     52$
 5100 044242  032767  000001  136010        BIT     #1,ERROUT               ;                       3712
 5101 044250  001416                        BEQ     52$
 5102 044252  032767  000001  170060        BIT     #1,RETRYING
 5103 044260  001012                        BNE     52$
 5104 044262  012746  010140                MOV     #PHR11,-(SP)
 5105 044266  012746  007072                MOV     #SAY1,-(SP)
 5106 044272  012746  000002                MOV     #2,-(SP)
 5107 044276  010600                        MOV     SP,R0                   ; SP,*
 5108 044300  104414                        TRAP    14
 5109 044302  062706  000006                ADD     #6,SP
 5110 044306  012705  000001        52$:    MOV     #1,R5                   ;                       3706
 5111 044312  010500                        MOV     R5,R0                   ;                       3428
 5112 044314  000401                        BR      54$
 5113 044316  005000                53$:    CLR     R0                      ;                       3427
 5114 044320  062706  000006        54$:    ADD     #6,SP
 5115 044324  000207                        RTS     PC
 5116
 5117                                ; Routine Size:  647 words
 5118                                ;MLX4
 5119                                ;               SYSTEM ERROR DETECTOR   27-Mar-1982 19:24:42    TOPS
 5120                                                                        27-Mar-1982 19:23:44    PA:<
 5121
 5126                                ; Maximum stack depth per invocation:  15 words
 5127
```

```
;MLX4
;                    DATA COMPARISON ROUTINE                    27-Mar-1982 19:24:42    TOPS-20 Bliss-16 V2(212)
;                                                               27-Mar-1982 19:23:44    PA:<NEALE>MLX4.BLI.5 (23)

;          3719  %sbttl 'DATA COMPARISON ROUTINE'
;          3720  routine DOUBLE_CHECK (W_POINTER, R_POINTER, COUNT) =
;          3721      begin                                                !* 1 *
;          3722
;          3723  !++
;          3724  !   ROUTINE:     DOUBLE_CHECK(W_POINTER,R_POINTER,COUNT)
;          3725  !
;          3726  !   PURPOSE:     TO DOUBLE CHECK THE ECC DETECTION LOGIC AFTER A
;          3727  !                SUCCESSFUL READ COMMAND.  THE WRITE AND READ BUFFERS
;          3728  !                ARE COMPARED, AND NO ERRORS SHOULD BE FOUND UNDER
;          3729  !                NORMAL CIRCUMSTANCES.
;          3730  !
;          3731  !   ARGUMENTS:   W_POINTER = POINTER TO THE WRITE BUFFER
;          3732  !                R_POINTER = POINTER TO THE READ BUFFER
;          3733  !                COUNT = THE NUMBER OF WORDS TO COMPARE
;          3734  !
;          3735  !   RESULTS:     VALUE RETURNED IS EITHER:
;          3736  !                        0 = NO ERRORS WERE FOUND
;          3737  !                    N > 0 = ADDRESS IN WRITE BUFFER WHERE ERROR WAS FOUND
;          3738  !--
;          3739
;          3740      local
;          3741          VALUE,
;          3742          OFFSET,
;          3743          GOOD,
;          3744          BAD;
;          3745
;          3746      label
;          3747          LOOP;
;          3748
;          3749      OFFSET = 0;
;          3750      VALUE = 0;
;          3751  LOOP :
;          3752      begin                                                !* 2 *
;          3753
;          3754      incr I from 1 to .COUNT do
;          3755          begin                                            !* 3 *
;          3756          GOOD = .(.W_POINTER + .OFFSET);
;          3757          BAD = .(.R_POINTER + .OFFSET);
;          3758
;          3759          if .GOOD eql .BAD
;          3760          then
;          3761              OFFSET = .OFFSET + 2
;          3762          else
;          3763              begin                                        !* 4 *
;          3764              VALUE = (.W_POINTER + .OFFSET);              !ADDRESS OF GOOD
;          3765              leave LOOP;
;          3766              end;                                         !* 4 *
;          3767
;          3768          end;                                             !* 3 *
;          3769
;          3770      end;                                                 !* 2 *
```

```
5185 ;MLX4
5186 ;                  DATA COMPARISON ROUTINE                    27-Mar-1982 19:24:42    TOPS-20 Bliss-16 V2(212)
5187                                                               27-Mar-1982 19:23:44    PA:<NEALE>MLX4.BLI.5 (23)
5188 ;          3771       return .VALUE;                         !EITHER 0 OR THE ADDRESS OF GOOD DATA
5189 ;          3772       end;                                   !* 1 *
5193
5194                                        .SBTTL  DOUBLE.CHECK DATA COMPARISON ROUTINE
5198 044326                          DOUBLE.CHECK:
5199 044326  004167  141002                JSR     R1,$SAVE5                       ;                       3720
5200 044332  005746                         TST     -(SP)                           ;
5201 044334  005003                         CLR     R3                              ; OFFSET                3749
5202 044336  005001                         CLR     R1                              ; VALUE                 3750
5203 044340  005002                         CLR     R2                              ; I                     3754
5204 044342  000417                         BR      3$
5205 044344  010304                   1$:   MOV     R3,R4                           ; OFFSET,*              3756
5206 044346  066604  000024                ADD     24(SP),R4                       ; W.POINTER,*
5207 044352  011416                         MOV     (R4),(SP)                       ; *,GOOD
5208 044354  010305                         MOV     R3,R5                           ; OFFSET,*              3757
5209 044356  066605  000022                ADD     22(SP),R5                       ; R.POINTER,*
5210 044362  011500                         MOV     (R5),R0                         ; *,BAD
5211 044364  021600                         CMP     (SP),R0                         ; *,BAD
5212 044366  001003                         BNE     2$                              ;                       3759
5213 044370  062703  000002                ADD     #2,R3                           ; *,OFFSET              3761
5214 044374  000402                         BR      3$                              ;                       3759
5215 044376  010401                   2$:   MOV     R4,R1                           ; *,VALUE               3764
5216 044400  000404                         BR      4$                              ;                       3765
5217 044402  005202                   3$:   INC     R2                              ; I                     3754
5218 044404  020266  000020                CMP     R2,20(SP)                       ; I,COUNT
5219 044410  003755                         BLE     1$
5220 044412  010100                   4$:   MOV     R1,R0                           ; VALUE,*               3721
5221 044414  005726                         TST     (SP)+                           ;                       3720
5222 044416  000207                         RTS     PC
5223
5224                                   ; Routine Size:  29 words
5225                                   ; Maximum stack depth per invocation:  7 words
5230
5231
```

```
;MLX4
;                    COMMAND INITIATION AND TERMINATION          27-Mar-1982 19:24:42    TOPS-20 Bliss-16 V2(212)
;                                                                27-Mar-1982 19:23:44    PA:<NEALE>MLX4.BLI.5 (24)

;          3773  %sbttl 'COMMAND INITIATION AND TERMINATION'
;          3774  routine WAITER : novalue =
;          3775      begin
;          3776  !+
;          3777  ! THIS ROUTINE DOES NOTHING, BUT IT IS CALLED BY 'START_IT' TO
;          3778  ! WASTE TIME WHILE WAITING TO BEGIN OR END AN ML11 TRANSFER.
;          3779  !-
;          3780      BREAK;
;          3781      return;
;          3782      end;

                                .SBTTL  WAITER COMMAND INITIATION AND TERMINATION
044420  104422          WAITER: TRAP    22                              ;                       3775
044422  000207                  RTS     PC                              ;                       3774

                        ; Routine Size:  2 words
                        ; Maximum stack depth per invocation:  0 words
```

```
;MLX4
;                    COMMAND INITIATION AND TERMINATION          27-Mar-1982 19:24:42    TOPS-20 Bliss-16 V2(212)
;                                                                27-Mar-1982 19:23:44    PA:<NEALE>MLX4.BLI.5 (25)

;          3783  routine START_IT (COMMAND, LUN, WRDCNT, BUFFER, SECTOR) =
;          3784      begin
;          3785
;          3786  !++
;          3787  !   ROUTINE:     START_IT(COMMAND,LUN,WRDCNT,BUFFER,SECTOR)
;          3788  !
;          3789  !   PURPOSE:     TO INITIATE A TRANSFER TO OR FROM THE ML11,
;          3790  !                TO WAIT FOR THE TRANSFER TO COMPLETE, AND TO CALL
;          3791  !                THE 'SYSERR' ROUTINE TO LOOK FOR RESULTING ERRORS.
;          3792  !
;          3793  !   ARGUMENTS:   (1)  COMMAND - THE FUNCTION CODE FOR THE TYPE OF TRANSFER
;          3794  !                               DESIRED.  THIS CODE WILL BE SENT TO THE MLCS1
;          3795  !                               REGISTER TO START THE OPERATION, SINCE IT
;          3796  !                               ALSO CONTAINS THE GO BIT.
;          3797  !
;          3798  !                (2)  LUN     - LOGICAL UNIT NUMBER
;          3799  !
;          3800  !                (3)  WRDCNT  - NUMBER OF 16-BIT WORDS TO TRANSFER
;          3801  !
;          3802  !                (4)  BUFFER  - THE ADDRESS IN MAIN MEMORY OF THE SELECTED
;          3803  !                               WRITE OR READ BUFFER
;          3804  !
;          3805  !                (5)  SECTOR   THE TRANSFER'S STARTING ADDRESS IN THE ML11
;          3806  !
;          3807  !   RESULTS:     VALUES RETURNED ARE IDENTICAL TO THOSE DEFINED ABOVE IN
;          3808  !                THE 'SYSERR' ROUTINE.
;          3809  !--
;          3810
;          3811      local
;          3812          TEMP,
;          3813          READY_BIT,
;          3814          RTN;
;          3815
;          3816      label
;          3817          LOOP;
;          3818
;          3819      !
;          3820      ! WAIT FOR DRIVE READY:
;          3821      !
;          3822      UNIT = .DRIVE;                          !SELECT THE UNIT
;          3823      READY_BIT = 0;
;          3824
;          3825      until .READY_BIT neq 0 do               !WAIT FOR DRIVE READY
;          3826          begin
;          3827          WAITER ();
;          3828          READY_BIT = .MLCS1 and BIT7;
;          3829          end;
;          3830
;          3831      !
;          3832      ! INITIALIZE BEFORE EACH TRANSFER:
;          3833      !
;          3834      CLR = 1;                                !CONTROLLER CLEAR
```

```
;MLX4
;                    COMMAND INITIATION AND TERMINATION                    27-Mar-1982 19:24:42    TOPS-20 Bliss-16 V2(212)
;                                                                          27-Mar-1982 19:23:44    PA:<NEALE>MLX4.BLI.5 (25)
;
;          3835      DELAY (1);                                            !DELAY FOR CLEAR TO COMPLETE
;          3836      MLCS1 = DRV_CLR;                                      !DRIVE CLEAR
;          3837      DELAY (1);                                            !DELAY FOR CLEAR TO COMPLETE
;          3838      UNIT = .DRIVE;                                        !PUT BACK THE DRIVE NUMBER
;          3839      !
;          3840      ! SET UP THE REQUIRED ENABLE/DISABLE BITS:
;          3841      !
;          3842      DCK_EN = 1;                                           !ALLOW REPORTING OF DATA CHECK ERRORS
;          3843
;          3844      if .REFRESH then REF_MAR = 1;                         !TURN ON REFRESH MARGINING IF OPERATOR SELECTED IT
;          3845
;          3846      if .ECCDIS then ECC_DIS = 1;                          !TURN OFF ECC IF OPERATOR SELECTED IT
;          3847
;          3848      !
;          3849      ! SEND REQUIRED INFORMATION TO DEVICE REGISTERS:
;          3850      !
;          3851      MLWC = -(.WRDCNT);                                    !WORD COUNT REGISTER
;          3852      MLBA = .BUFFER;                                       !BUS ADDRESS REGISTER (ADDRESS IN MAIN MEMORY)
;          3853      MLDA = .SECTOR;                                       !DESIRED SECTOR ADDRESS REGISTER (ADDRESS IN ML11)
;          3854      !
;          3855      ! GO:
;          3856      !
;          3857      I_AM_DONE = INACTIVE;
;          3858      TEMP = .COMMAND + BIT6;                               !SET THE INTERRUPT ENABLE BIT WHILE LOADING THE
;          3859      MLCS1 = .TEMP;                                        !CONTROL AND STATUS REGISTER WITH THE COMMAND
;          3860      !
;          3861      ! WAIT FOR DRIVE TO FINISH:
;          3862      !
;          3863  LOOP :
;          3864      begin
;          3865      READY_BIT = 0;
;          3866
;          3867      until .I_AM_DONE do
;          3868          begin
;          3869          WAITER ();
;          3870          READY_BIT = .MLCS1 and BIT7;
;          3871
;          3872          if ((.READY_BIT neq 0) and (.I_AM_DONE eql INACTIVE))
;          3873          then
;          3874              begin
;          3875
;          3876              if ( not .RETRYING)
;          3877              then
;          3878                  begin
;          3879                  SAYWHO (.LUN);
;          3880
;          3881                  if .COMMAND eql WR_CMD then RTN = WRD16;
;          3882
;          3883                  if .COMMAND eql RD_CMD then RTN = WRD17;
;          3884
;          3885                  if .COMMAND eql WC_CMD then RTN = PHR1;
;          3886
```

```
5378  ;MLX4
5379  ;                    COMMAND INITIATION AND TERMINATION          27-Mar-1982 19:24:42    TOPS-20 Bliss-16 V2(212)
5380  ;                                                                27-Mar-1982 19:23:44    PA:<NEALE>MLX4.BLI.5 (25)
5381  ;          3887                    PRINTB (FMT6, .WRDCNT, .RTN, WRD15, .SECTOR);
5382  ;          3888                    !'BEGAN YYYY WORD (???) AT SECTOR ZZZZZZ'
5383  ;          3889                    !WHERE (???) IS EITHER 'WRITE', 'READ' OR 'WRITE CHECK'
5384  ;          3890                    PRINTB (SAY1, MSG0);
5385  ;          3891                    !INTERRUPT DID NOT OCCUR, BUT THE TRANSFER IS COMPLETE
5386  ;          3892                    end;
5387  ;          3893
5388  ;          3894                leave LOOP;
5389  ;          3895                end;
5390  ;          3896
5391  ;          3897            end;
5392  ;          3898
5393  ;          3899        end;
5394  ;          3900        return SYSERR (.LUN);                   !PASS ALONG THE ERROR VALUES THAT 'SYSERR' OBTAINED.
5395  ;          3901        end;
5399
5400                                      .GLOBL  L$DLY
5401
5402
5403                                      .SBTTL  START.IT COMMAND INITIATION AND TERMINATION
5407  044424                        START.IT:
5408  044424  004167  140664              JSR     R1,$SAVE4                       ;                               3783
5409  044430  005746                      TST     -(SP)
5410  044432  016604  000024              MOV     24(SP),R4                       ; LUN,*                         3822
5411  044436  010401                      MOV     R4,R1
5412  044440  006301                      ASL     R1
5413  044442  016102  034422              MOV     PTABLE.ADDR(R1),R2
5414  044446  016203  000006              MOV     6(R2),R3
5415  044452  042703  177770              BIC     #177770,R3
5416  044456  142777  000007  167672      BICB    #7,@ML.REG+10
5417  044464  150377  167666              BISB    R3,@ML.REG+10
5418  044470  005003                      CLR     R3
5419  044472  020707                      CMP     PC,PC                           ; READY.BIT                     3823
5420  044474  001007                      BNE     2$                              ;                               3825
5421  044476  004767  177716        1$:   JSR     PC,WAITER                       ;                               3827
5422  044502  017703  167640              MOV     @ML.REG,R3                      ; *,READY.BIT                   3828
5423  044506  042703  177577              BIC     #177577,R3                      ; *,READY.BIT
5424  044512  000770                      BR      1$                              ;
5425  044514  152777  000040  167634  2$: BISB    #40,@ML.REG+10                  ;                               3825
5426  044522  012700  000001              MOV     #1,R0                           ;                               3834
5427  044526  001410                3$:   BEQ     6$                              ; *,$STMP2                      3835
5428  044530  016702  135362              MOV     L$DLY,R2
5429  044534  001403                      BEQ     5$                              ; *,$STMP1
5430  044536  005016                4$:   CLR     (SP)                            ; $STMP
5431  044540  005302                      DEC     R2                              ; $STMP1
5432  044542  001375                      BNE     4$
```

```
                                        ;MLX4                                                                    27-Mar-1982 19:24:42    TOPS
                                        ;               COMMAND INITIATION AND TERMINATION                       27-Mar-1982 19:23:44    PA:<
                                5$:     DEC     R0                              ; $$TMP2
                                        BR      3$
                        000011  167570  6$:     MOV     #11,@ML.REG             ;                                                        3836
                        000001          7$:     MOV     #1,R0                   ; *,$$TMP2                                               3837
                                        BEQ     10$
                        135326                  MOV     L$DLY,R2                ; *,$$TMP1
                                        BEQ     9$
                                8$:     CLR     (SP)                            ; $$TMP
                                        DEC     R2                              ; $$TMP1
                                        BNE     8$
                                9$:     DEC     R0                              ; $$TMP2
                                        BR      7$
                        034422          10$:    MOV     PTABLE.ADDR(R1),R2      ;                                                        3838
                        000006                  MOV     6(R2),R1
                        177770                  BIC     #177770,R1
                        000007  167530          BICB    #7,@ML.REG+10
                        167524                  BISB    R1,@ML.REG+10
                        000004  167532          BISB    #4,@ML.REG+24
                        000001  135404          BIT     #1,REFRESH              ;                                                        3842
                                                                                ;                                                        3844
                        000200  167514          BISB    #200,@ML.REG+24
                        000001  135370  11$:     BIT     #1,ECCDIS               ;                                                        3846
                                        BEQ     12$
                        000002  167476          BISB    #2,@ML.REG+24
                        000022  167446  12$:     MOV     22(SP),@ML.REG+2        ; WRDCNT,*                                               3851
                        167442                  NEG     @ML.REG+2
                        000020  167436          MOV     20(SP),@ML.REG+4        ; BUFFER,*                                               3852
                        000016  167432          MOV     16(SP),@ML.REG+6        ; SECTOR,*                                               3853
                        167410                  CLR     I.AM.DONE               ;                                                        3857
                        000026                  MOV     26(SP),R2               ; COMMAND,*                                              3858
                                                MOV     R2,R1                   ; *,TEMP
                        000100                  ADD     #100,R1                 ; *,TEMP
                        167402                  MOV     R1,@ML.REG              ; TEMP,*
                                                CLR     R3                      ; READY.BIT                                              3859
                        000001  167362  13$:    BIT     #1,I.AM.DONE            ;                                                        3865
                                                BNE     17$                                                                              3867
                        177436                  JSR     PC,WAITER               ;
                        167360                  MOV     @ML.REG,R3              ; *,READY.BIT                                            3869
                        177577                  BIC     #177577,R3              ; *,READY.BIT                                            3870
                                                BEQ     13$                     ;
                        167336                  TST     I.AM.DONE                                                                        3872
                                                BNE     13$
                        000001  167330          BIT     #1,RETRYING             ;                                                        3876
                                                BNE     17$
                                                MOV     R4,-(SP)                ;                                                        3879
                        170504                  JSR     PC,SAYWHO
                        000061                  CMP     R2,#61                  ;                                                        3881
                                                BNE     14$
                        007256                  MOV     #WRD16,R1               ; *,RTN
                        000071          14$:    CMP     R2,#71                  ;                                                        3883
                                                BNE     15$
                        007264                  MOV     #WRD17,R1               ; *,RTN
```

```
                                        ;MLX4
                                        ;               COMMAND INITIATION AND TERMINATION                27-Mar-1982 19:24:42   TOPS
                                        ;                                                                 27-Mar-1982 19:23:44   PA:<
045044  020227  000051                15$:    CMP     R2,#51                          ;                                        3885
045050  001002                                BNE     16$
045052  012701  007714                        MOV     #PHR1,R1                        ; *,RTN
045056  016616  000020                16$:    MOV     20(SP),(SP)                     ; SECTOR,*                               3887
045062  012746  007246                        MOV     #WRD15,-(SP)
045066  010146                                MOV     R1,-(SP)                        ; RTN,*
045070  016646  000030                        MOV     30(SP),-(SP)                    ; WRDCNT,*
045074  012746  006376                        MOV     #FMT6,-(SP)
045100  012746  000005                        MOV     #5,-(SP)
045104  010600                                MOV     SP,R0                           ; SP,*
045106  104414                                TRAP    14
045110  012716  011002                        MOV     #MSG0,(SP)                      ;                                        3890
045114  012746  007072                        MOV     #SAY1,-(SP)
045120  012746  000002                        MOV     #2,-(SP)
045124  010600                                MOV     SP,R0                           ; SP,*
045126  104414                                TRAP    14
045130  062706  000020                        ADD     #20,SP                          ;
045134  010446                        17$:    MOV     R4,-(SP)                        ;                                        3878
045136  004767  174546                        JSR     PC,SYSERR                       ;                                        3900
045142  022626                                CMP     (SP)+,(SP)+                     ;
045144  000207                                RTS     PC                                                                       3783

                                        ; Routine Size:  169 words
                                        ; Maximum stack depth per invocation:  14 words
```

```
;MLX4
;                COUNTING BYTES TRANSFERED                    27-Mar-1982 19:24:42    TOPS-20 Bliss-16 V2(212)
;                                                             27-Mar-1982 19:23:44    PA:<NEALE>MLX4.BLI.5 (26)

                3902  %sbttl 'COUNTING BYTES TRANSFERED'
                3903  routine UP_WR_COUNT (WORD_COUNT) : novalue =
                3904      begin
                3905      WR_COUNT = .WR_COUNT + (.WORD_COUNT*2);
                3906
                3907      if .WR_COUNT geq 20000
                3908      then
                3909          begin
                3910          WR_THOUSANDS = .WR_THOUSANDS + 20;
                3911          WR_COUNT = .WR_COUNT - 20000;
                3912
                3913          if .WR_THOUSANDS geq 1000
                3914          then
                3915              begin
                3916              WR_THOUSANDS = .WR_THOUSANDS - 1000;
                3917              WR_MILLIONS = .WR_MILLIONS + 1;
                3918
                3919              if .WR_MILLIONS lss 0 then WR_MILLIONS = 0;
                3920
                3921              end;
                3922
                3923          end;
                3924
                3925      return;
                3926      end;


                                        .SBTTL  UP.WR.COUNT COUNTING BYTES TRANSFERED
045146                          UP.WR.COUNT:
045146  016600  000002                MOV     2(SP),R0                ; WORD.COUNT,*                  3905
045152  006300                        ASL     R0
045154  066700  167134                ADD     WR.COUNT,R0
045160  010067  167130                MOV     R0,WR.COUNT
045164  020027  047040                CMP     R0,#47040               ; WR.COUNT,*                    3907
045170  002422                        BLT     1$
045172  062767  000024  167116        ADD     #24,WR.THOUSANDS        ;                               3910
045200  162767  047040  167106        SUB     #47040,WR.COUNT         ;                               3911
045206  026727  167104  001750        CMP     WR.THOUSANDS,#1750      ;                               3913
045214  002410                        BLT     1$
045216  162767  001750  167072        SUB     #1750,WR.THOUSANDS      ;                               3916
045224  005267  167070                INC     WR.MILLIONS             ;                               3917
045230  100002                        BPL     1$                      ;                               3919
045232  005067  167062                CLR     WR.MILLIONS
045236  000207                1$:     RTS     PC                      ;                               3903

                                ; Routine Size:  29 words
                                ; Maximum stack depth per invocation:  0 words
                                ;MLX4
                                ;               COUNTING BYTES TRANSFERED                    27-Mar-1982 19:24:42    TOPS
                                ;                                                            27-Mar-1982 19:23:44    PA:<
```

```
;MLX4
;                    COUNTING BYTES TRANSFERED                    27-Mar-1982 19:24:42    TOPS-20 Bliss-16 V2(212)
;                                                                 27-Mar-1982 19:23:44    PA:<NEALE>MLX4.BLI.5 (27)

;                3927  routine UP_RD_COUNT (WORD_COUNT) : novalue =
;                3928      begin
;                3929      RD_COUNT = .RD_COUNT + (.WORD_COUNT*2);
;                3930
;                3931      if .RD_COUNT geq 20000
;                3932      then
;                3933          begin
;                3934          RD_THOUSANDS = .RD_THOUSANDS + 20;
;                3935          RD_COUNT = .RD_COUNT - 20000;
;                3936
;                3937          if .RD_THOUSANDS geq 1000
;                3938          then
;                3939              begin
;                3940              RD_THOUSANDS = .RD_THOUSANDS - 1000;
;                3941              RD_MILLIONS = .RD_MILLIONS + 1;
;                3942
;                3943              if .RD_MILLIONS lss 0 then RD_MILLIONS = 0;
;                3944
;                3945              end;
;                3946
;                3947          end;
;                3948
;                3949      return;
;                3950      end;

                                        .SBTTL  UP.RD.COUNT COUNTING BYTES TRANSFERED
045240                          UP.RD.COUNT:
045240  016600  000002                MOV     2(SP),R0                        ; WORD.COUNT,*          3929
045244  006300                        ASL     R0
045246  066700  167050                ADD     RD.COUNT,R0
045252  010067  167044                MOV     R0,RD.COUNT
045256  020027  047040                CMP     R0,#47040                       ; RD.COUNT,*            3931
045262  002422                        BLT     1$
045264  062767  000024  167032        ADD     #24,RD.THOUSANDS                ;                       3934
045272  162767  047040  167022        SUB     #47040,RD.COUNT                 ;                       3935
045300  026727  167020  001750        CMP     RD.THOUSANDS,#1750              ;                       3937
045306  002410                        BLT     1$
045310  162767  001750  167006        SUB     #1750,RD.THOUSANDS              ;                       3940
045316  005267  167004                INC     RD.MILLIONS                     ;                       3941
045322  100002                        BPL     1$                              ;                       3943
045324  005067  166776                CLR     RD.MILLIONS
045330  000207                  1$:   RTS     PC                              ;                       3927

                                ; Routine Size:  29 words
                                ; Maximum stack depth per invocation:  0 words
```

```
;MLX4
;                       COUNTING BYTES TRANSFERED                       27-Mar-1982 19:24:42    TOPS-20 Bliss-16 V2(212)
;                                                                       27-Mar-1982 19:23:44    PA:<NEALE>MLX4.BLI.5 (28)

;                3951   routine UP_WC_COUNT (WORD_COUNT) : novalue =
;                3952       begin
;                3953       WC_COUNT = .WC_COUNT + (.WORD_COUNT*2);
;                3954
;                3955       if .WC_COUNT geq 20000
;                3956       then
;                3957           begin
;                3958           WC_THOUSANDS = .WC_THOUSANDS + 20;
;                3959           WC_COUNT = .WC_COUNT - 20000;
;                3960
;                3961           if .WC_THOUSANDS geq 1000
;                3962           then
;                3963               begin
;                3964               WC_THOUSANDS = .WC_THOUSANDS - 1000;
;                3965               WC_MILLIONS = .WC_MILLIONS + 1;
;                3966
;                3967               if .WC_MILLIONS lss 0 then WC_MILLIONS = 0;
;                3968
;                3969               end;
;                3970
;                3971           end;
;                3972
;                3973       return;
;                3974       end;

                                        .SBTTL  UP.WC.COUNT COUNTING BYTES TRANSFERED
045332                          UP.WC.COUNT:
045332  016600  000002                  MOV     2(SP),R0                        ; WORD.COUNT,*          3953
045336  006300                          ASL     R0
045340  066700  166764                  ADD     WC.COUNT,R0
045344  010067  166760                  MOV     R0,WC.COUNT
045350  020027  047040                  CMP     R0,#47040                       ; WC.COUNT,*            3955
045354  002422                          BLT     1$
045356  062767  000024  166746          ADD     #24,WC.THOUSANDS                ;                       3958
045364  162767  047040  166736          SUB     #47040,WC.COUNT                 ;                       3959
045372  026727  166734  001750          CMP     WC.THOUSANDS,#1750              ;                       3961
045400  002410                          BLT     1$
045402  162767  001750  166722          SUB     #1750,WC.THOUSANDS              ;                       3964
045410  005267  166720                  INC     WC.MILLIONS                     ;                       3965
045414  100002                          BPL     1$                              ;                       3967
045416  005067  166712                  CLR     WC.MILLIONS
045422  000207                  1$:     RTS     PC                              ;                       3951

                                ; Routine Size:  29 words
                                ; Maximum stack depth per invocation:  0 words
```

```
;MLX4
;                ML11 TRANSFER COMMANDS                          27-Mar-1982 19:24:42    TOPS-20 Bliss-16 V2(212)
;                                                                27-Mar-1982 19:23:44    PA:<NEALE>MLX4.BLI.5 (29)

;          3975  %sbttl 'ML11 TRANSFER COMMANDS'
;          3976  routine write (LUN, WRDCNT, BUFFER, SECTOR) =
;          3977      begin
;          3978
;          3979  !++
;          3980  !  ROUTINE:     WRITE(LUN,WRDCNT,BUFFER,SECTOR)
;          3981  !
;          3982  !  PURPOSE:     TO TRANSFER INFORMATION FROM MAIN MEMORY TO THE ML11
;          3983  !               AND TO DECIDE ON THE ADVISABILITY OF A RETRY.
;          3984  !
;          3985  !  ARGUMENTS:   SEE 'START_IT' ROUTINE ABOVE FOR DETAILS
;          3986  !
;          3987  !  RESULTS:     UPON TERMINATION OF THE TRANSFER, ERROR DETECTION OCCURS
;          3988  !               AND THE CONTENTS OF 'VALUE' BECOMES IMPORTANT IN TERMS OF
;          3989  !               WHETHER THE TRANSFER WAS COMPLETELY SUCCESSFUL OR WHETHER A
;          3990  !               RETRY SHOULD OR SHOULD NOT BE PERMITTED.
;          3991  !
;          3992  !               THE VALUES RETURNED FOR THIS SUBROUTINE ARE THE SAME AS
;          3993  !               FOR THE SYSERR ROUTINE ABOVE.
;          3994  !--
;          3995
;          3996      local
;          3997          VALUE,
;          3998          COMMAND;
;          3999
;          4000      COMMAND = WR_CMD;
;          4001      VALUE = START_IT (.COMMAND, .LUN, .WRDCNT, .BUFFER, .SECTOR);
;          4002
;          4003      if ( not .RETRYING) then UP_WR_COUNT (.WRDCNT);
;          4004
;          4005      if .VALUE eql 0 then return 0;                !NO ERRORS AT ALL
;          4006
;          4007      if ((.ERROUT) and ( not .RETRYING)) then PRINTB (FMT6, .WRDCNT, WRD16, WRD15, .SECTOR);
;          4008
;          4009      !'BEGAN YYYY WORD WRITE AT SECTOR ZZZZZZ'
;          4010      return .VALUE;
;          4011      end;
;
;                                    .SBTTL  WRITE ML11 TRANSFER COMMANDS
;5765 045424 004167 137632  WRITE:  JSR     R1,$SAVE2                       ;                                3976
;5766 045430 012700 000061          MOV     #61,R0                          ; *,COMMAND                      4000
;5767 045434 010046                 MOV     R0,-(SP)                        ; COMMAND,*                      4001
;5768 045436 016646 000020          MOV     20(SP),-(SP)                    ; LUN,*
;5769 045442 016601 000020          MOV     20(SP),R1                       ; WRDCNT,*
;5770 045446 010146                 MOV     R1,-(SP)
;5771 045450 016646 000020          MOV     20(SP),-(SP)                    ; BUFFER,*
```

```
                                        ;MLX4
                                        ;                ML11 TRANSFER COMMANDS                    27-Mar-1982 19:24:42    TOPS
                                                                                                   27-Mar-1982 19:23:44    PA:<

045454  016646  000020                        MOV     20(SP),-(SP)        ; SECTOR,*
045460  004767  176740                        JSR     PC,START.IT
045464  010002                                MOV     R0,R2               ; *,VALUE
045466  032767  000001  166644                BIT     #1,RETRYING         ;                                  4003
045474  001004                                BNE     1$
045476  010146                                MOV     R1,-(SP)
045500  004767  177442                        JSR     PC,UP.WR.COUNT
045504  005726                                TST     (SP)+
045506  005702                        1$:     TST     R2                  ; VALUE                            4005
045510  001003                                BNE     2$
045512  062706  000012                        ADD     #12,SP
045516  000433                                BR      4$
045520  032767  000001  134532        2$:     BIT     #1,ERROUT           ;                                  4007
045526  001423                                BEQ     3$
045530  032767  000001  166602                BIT     #1,RETRYING
045536  001017                                BNE     3$
045540  016646  000022                        MOV     22(SP),-(SP)        ; SECTOR,*
045544  012746  007246                        MOV     #WRD15,-(SP)
045550  012746  007256                        MOV     #WRD16,-(SP)
045554  010146                                MOV     R1,-(SP)
045556  012746  006376                        MOV     #FMT6,-(SP)
045562  012746  000005                        MOV     #5,-(SP)
045566  010600                                MOV     SP,R0               ; SP,*
045570  104414                                TRAP    14
045572  062706  000014                        ADD     #14,SP
045576  062706  000012                3$:     ADD     #12,SP              ;                                  3976
045602  010200                                MOV     R2,R0               ; VALUE,*                          3977
045604  000207                                RTS     PC
045606  005000                        4$:     CLR     R0                  ;                                  3976
045610  000207                                RTS     PC

                                        ; Routine Size:  59 words
                                        ; Maximum stack depth per invocation:  14 words
```

```
5816 ;MLX4
5817 ;                  ML11 TRANSFER COMMANDS                          27-Mar-1982 19:24:42    TOPS-20 Bliss-16 V2(212)
5818 ;                                                                  27-Mar-1982 19:23:44    PA:<NEALE>MLX4.BLI.5 (30)
5819 ;          4012    routine read (LUN, WRDCNT, BUFFER, SECTOR) =
5820 ;          4013        begin
5821 ;          4014
5822 ;          4015    !++
5823 ;          4016    !   ROUTINE:    READ(LUN,WRDCNT,BUFFER,SECTOR)
5824 ;          4017    !
5825 ;          4018    !   PURPOSE:    TO TRANSFER INFORMATION FROM THE ML11 TO MAIN MEMORY
5826 ;          4019    !               USING A 'READ' COMMAND.
5827 ;          4020    !
5828 ;          4021    !   ARGUMENTS:  SEE 'START_IT' ROUTINE ABOVE FOR DETAILS
5829 ;          4022    !
5830 ;          4023    !   RESULTS:    UPON TERMINATION OF THE TRANSFER, ERROR DETECTION OCCURS
5831 ;          4024    !               AND THE CONTENTS OF 'VALUE' BECOMES IMPORTANT IN TERMS OF
5632 ;          4025    !               WHETHER THE TRANSFER WAS COMPLETELY SUCCESSFUL OR WHETHER A
5833 ;          4026    !               RETRY SHOULD OR SHOULD NOT BE PERMITTED.
5834 ;          4027    !
5835 ;          4028    !               THE VALUES RETURNED FOR THIS SUBROUTINE ARE THE SAME AS
5836 ;          4029    !               FOR THE SYSERR ROUTINE ABOVE.
5837 ;          4030    !--
5838 ;          4031
5839 ;          4032        local
5840 ;          4033            VALUE,
5841 ;          4034            COMMAND;
5842 ;          4035
5843 ;          4036        COMMAND = RD_CMD;
5844 ;          4037        VALUE = START_IT (.COMMAND, .LUN, .WRDCNT, .BUFFER, .SECTOR);
5845 ;          4038
5846 ;          4039        if ( not .RETRYING) then UP_RD_COUNT (.WRDCNT);
5847 ;          4040
5848 ;          4041        if .VALUE eql 0 then return 0;              !NO ERRORS AT ALL
5849 ;          4042
5850 ;          4043        if ((.ERROUT) and ( not .RETRYING)) then PRINTB (FMT6, .WRDCNT, WRD17, WRD15, .SECTOR);
5851 ;          4044
5852 ;          4045        !'BEGAN YYYY WORD READ AT SECTOR ZZZZZZ'
5853 ;          4046        return .VALUE;
5854 ;          4047        end;
5858
5859                                        .SBTTL  READ ML11 TRANSFER COMMANDS
5863 045612  004167  137444        READ:   JSR     R1,$SAVE2                       ;                                   4012
5864 045616  012700  000071                MOV     #71,R0                          ; *,COMMAND                         4036
5865 045622  010046                        MOV     R0,-(SP)                        ; COMMAND,*                         4037
5866 045624  016646  000020                MOV     20(SP),-(SP)                    ; LUN,*
5867 045630  016601  000020                MOV     20(SP),R1                       ; WRDCNT,*
5868 045634  010146                        MOV     R1,-(SP)
5869 045636  016646  000020                MOV     20(SP),-(SP)                    ; BUFFER,*
5870 045642  016646  000020                MOV     20(SP),-(SP)                    ; SECTOR,*
```

```
                                        ;MLX4
                                        ;               ML11 TRANSFER COMMANDS                  27-Mar-1982 19:24:42    TOPS
                                                                                                27-Mar-1982 19:23:44    PA:<

045646  004767  176552                        JSR     PC,START.IT
045652  010002                                MOV     R0,R2           ; *,VALUE
045654  032767  000001  166456                BIT     #1,RETRYING     ;                               4039
045662  001004                                BNE     1$
045664  010146                                MOV     R1,-(SP)
045666  004767  177346                        JSR     PC,UP.RD.COUNT
045672  005726                                TST     (SP)+
045674  005702                        1$:     TST     R2              ; VALUE                         4041
045676  001003                                BNE     2$
045700  062706  000012                        ADD     #12,SP
045704  000433                                BR      4$
045706  032767  000001  134344        2$:     BIT     #1,ERROUT       ;                               4043
045714  001423                                BEQ     3$
045716  032767  000001  166414                BIT     #1,RETRYING
045724  001017                                BNE     3$
045726  016646  000022                        MOV     22(SP),-(SP)    ; SECTOR,*
045732  012746  007246                        MOV     #WRD15,-(SP)
045736  012746  007264                        MOV     #WRD17,-(SP)
045742  010146                                MOV     R1,-(SP)
045744  012746  006376                        MOV     #FMT6,-(SP)
045750  012746  000005                        MOV     #5,-(SP)
045754  010600                                MOV     SP,R0           ; SP,*
045756  104414                                TRAP    14
045760  062706  000014                        ADD     #14,SP
045764  062706  000012                3$:     ADD     #12,SP          ;                               4012
045770  010200                                MOV     R2,R0           ; VALUE,*                       4013
045772  000207                                RTS     PC
045774  005000                        4$:     CLR     R0              ;                               4012
045776  000207                                RTS     PC

                                        ; Routine Size:  59 words
                                        ; Maximum stack depth per invocation:  14 words
```

```
5914 ;MLX4
5915 ;                  ML11 TRANSFER COMMANDS                          27-Mar-1982 19:24:42    TOPS-20 Bliss-16 V2(212)
5916 ;                                                                  27-Mar-1982 19:23:44    PA:<NEALE>MLX4.BLI.5 (31)
5917 ;          4048  routine CHECK (LUN, WRDCNT, BUFFER, SECTOR) =
5918 ;          4049      begin
5919 ;          4050
5920 ;          4051  !++
5921 ;          4052  !   ROUTINE:      CHECK(LUN,WRDCNT,BUFFER,SECTOR)
5922 ;          4053  !
5923 ;          4054  !   PURPOSE:      TO TRANSFER INFORMATION FROM THE ML11 TO MAIN MEMORY
5924 ;          4055  !                 USING A 'WRITE CHECK' COMMAND.
5925 ;          4056  !
5926 ;          4057  !   ARGUMENTS:    SEE 'START_IT' ROUTINE ABOVE FOR DETAILS
5927 ;          4058  !
5928 ;          4059  !   RESULTS:      UPON TERMINATION OF THE TRANSFER, ERROR DETECTION OCCURS
5929 ;          4060  !                 AND THE CONTENTS OF 'VALUE' BECOMES IMPORTANT IN TERMS OF
5930 ;          4061  !                 WHETHER THE TRANSFER WAS COMPLETELY SUCCESSFUL OR WHETHER A
5931 ;          4062  !                 RETRY SHOULD OR SHOULD NOT BE PERMITTED.
5932 ;          4063  !
5933 ;          4064  !                 THE VALUES RETURNED FOR THIS SUBROUTINE ARE THE SAME AS
5934 ;          4065  !                 FOR THE SYSERR ROUTINE ABOVE.
5935 ;          4066  !--
5936 ;          4067
5937 ;          4068      local
5938 ;          4069          VALUE,
5939 ;          4070          COMMAND;
5940 ;          4071
5941 ;          4072      COMMAND = WC_CMD;
5942 ;          4073      VALUE = START_IT (.COMMAND, .LUN, .WRDCNT, .BUFFER, .SECTOR);
5943 ;          4074
5944 ;          4075      if ( not .RETRYING) then UP_WC_COUNT (.WRDCNT);
5945 ;          4076
5946 ;          4077      if .VALUE eql 0 then return 0;                !NO ERRORS AT ALL
5947 ;          4078
5948 ;          4079      if ((.ERROUT) and ( not .RETRYING)) then PRINTB (FMT6, .WRDCNT, PHR1, WRD15, .SECTOR);
5949 ;          4080
5950 ;          4081      !'BEGAN YYYY WORD WRITE CHECK AT SECTOR ZZZZZZ'
5951 ;          4082      return .VALUE;
5952 ;          4083      end;
5956
5957                                      .SBTTL  CHECK ML11 TRANSFER COMMANDS
5961 046000  004167  137256      CHECK:  JSR     R1,$SAVE2                       ;                                   4048
5962 046004  012700  000051              MOV     #51,R0                          ; *,COMMAND                         4072
5963 046010  010046                      MOV     R0,-(SP)                        ; COMMAND,*                         4073
5964 046012  016646  000020              MOV     20(SP),-(SP)                    ; LUN,*
5965 046016  016601  000020              MOV     20(SP),R1                       ; WRDCNT,*
5966 046022  010146                      MOV     R1,-(SP)
5967 046024  016646  000020              MOV     20(SP),-(SP)                    ; BUFFER,*
5968 046030  016646  000020              MOV     20(SP),-(SP)                    ; SECTOR,*
```

```
                        ;MLX4                                                                    27-Mar-1982 19:24:42    TOPS
                        ;               ML11 TRANSFER COMMANDS                                   27-Mar-1982 19:23:44    PA:<
046034  004767  176364                JSR     PC,START.IT
046040  010002                        MOV     R0,R2
046042  032767  000001  166270        BIT     #1,RETRYING             ; *,VALUE
046050  001004                        BNE     1$                      ;                                                4075
046052  010146                        MOV     R1,-(SP)
046054  004767  177252                JSR     PC,UP.WC.COUNT
046060  005726                        TST     (SP)+
046062  005702                1$:     TST     R2                      ; VALUE                                          4077
046064  001003                        BNE     2$
046066  062706  000012                ADD     #12,SP
046072  000433                        BR      4$
046074  032767  000001  134156  2$:   BIT     #1,ERROUT               ;                                                4079
046102  001423                        BEQ     3$
046104  032767  000001  166226        BIT     #1,RETRYING
046112  001017                        BNE     3$
046114  016646  000022                MOV     22(SP),-(SP)            ; SECTOR,*
046120  012746  007246                MOV     #WRD15,-(SP)
046124  012746  007714                MOV     #PHR1,-(SP)
046130  010146                        MOV     R1,-(SP)
046132  012746  006376                MOV     #FMT6,-(SP)
046136  012746  000005                MOV     #5,-(SP)
046142  010600                        MOV     SP,R0                   ; SP,*
046144  104414                        TRAP    14
046146  062706  000014                ADD     #14,SP
046152  062706  000012          3$:   ADD     #12,SP                  ;                                                4048
046156  010200                        MOV     R2,R0                   ; VALUE,*                                        4049
046160  000207                        RTS     PC
046162  005000                4$:     CLR     R0                      ;                                                4048
046164  000207                        RTS     PC

                        ; Routine Size:  59 words
                        ; Maximum stack depth per invocation:  14 words
```

```
6012 ;MLX4
6013 ;                  ML11 TRANSFER COMMANDS                  27-Mar-1982 19:24:42    TOPS-20 Bliss-16 V2(212)
6014 ;                                                          27-Mar-1982 19:23:44    PA:<NEALE>MLX4.BLI.5 (32)
6015 ;          4084    routine CHOOSE =
6016 ;          4085        begin
6017 ;          4086
6018 ;          4087    !++
6019 ;          4088    !  ROUTINE:     CHOOSE
6020 ;          4089    !
6021 ;          4090    !  PURPOSE:     TO DECIDE WHETHER TO DO A 'WRITE CHECK' OR A 'READ'.
6022 ;          4091    !               THE 'READ' WILL BE SELECTED APPROXIMATELY 25% OF THE
6023 ;          4092    !               TIME.  THIS IS ACCOMPLISHED BY EXAMINING THE MOST AND
6024 ;          4093    !               THE LEAST SIGNIFICANT BITS OF A RANDOM NUMBER.  IF
6025 ;          4094    !               BOTH ARE SET, THEN 'READ' IS CHOSEN.
6026 ;          4095    !
6027 ;          4096    !  RESULTS:     THE VALUE RETURNED FOR THIS ROUTINE IS THE ADDRESS OF
6028 ;          4097    !               THE CHOSEN COMMAND (EITHER 'READ' OR 'CHECK').
6029 ;          4098    !--
6030 ;          4099
6031 ;          4100        local
6032 ;          4101            VALUE;
6033 ;          4102
6034 ;          4103        RN ();
6035 ;          4104
6036 ;          4105        if ((.RANDOM) and (.RANDOM lss 0))
6037 ;          4106        then
6038 ;          4107            VALUE = read                        !CHOOSE THE READ COMMAND
6039 ;          4108        else
6040 ;          4109            VALUE = CHECK;                      !CHOOSE THE WRITE CHECK COMMAND
6041 ;          4110
6042 ;          4111        return .VALUE;
6043 ;          4112        end;
6047
6048
                                .SBTTL  CHOOSE ML11 TRANSFER COMMANDS
6052 046166 004767 137176       CHOOSE: JSR     PC,RN
6053 046172 032767 000001 137264        BIT     #1,RANDOM           ;                       4103
6054 046200 001406                      BEQ     1$                  ;                       4105
6055 046202 005767 137256               TST     RANDOM
6056 046206 002003                      BGE     1$
6057 046210 012700 045612               MOV     #READ,R0            ; *,VALUE               4107
6058 046214 000207                      RTS     PC                  ;                       4105
6059 046216 012700 046000       1$:     MOV     #CHECK,R0           ; *,VALUE               4109
6060 046222 000207                      RTS     PC                  ;                       4084
6061
6062                            ; Routine Size:  15 words
6063                            ; Maximum stack depth per invocation:  0 words
6071
6072
```

```
6074 ;MLX4
6075 ;                  ML11 TRANSFER COMMANDS                  27-Mar-1982 19:24:42    TOPS-20 Bliss-16 V2(212)
6076 ;                                                          27-Mar-1982 19:23:44    PA:<NEALE>MLX4.BLI.5 (33)
6077 ;          4113  routine RETRY (TIMES, COMMAND, LUN, WRDCNT, BUFFER, SECTOR) =
6078 ;          4114      begin                                                !* 1 *
6079 ;          4115
6080 ;          4116  !++
6081 ;          4117  !  ROUTINE:     RETRY(TIMES,COMMAND,LUN,WRDCNT,BUFFER,SECTOR)
6082 ;          4118  !
6083 ;          4119  !  PURPOSE:     (1)  IF THE OPERATOR HAS ALLOWED ERROR PRINTOUTS, AND
6084 ;          4120  !                    IF THE RETRY IS NOT TO CATEGORIZE ERRORS, THEN SAY
6085 ;          4121  !                    THAT THE RETRY IS BEGINNING.
6086 ;          4122  !
6087 ;          4123  !                (2)  REISSUE THE WRITE, WRITE CHECK OR READ COMMAND
6088 ;          4124  !                    UNTIL THE COMMAND SUCCEEDS OR UNTIL ALL PERMITTED
6089 ;          4125  !                    RETRIES HAVE FAILED.
6090 ;          4126  !
6091 ;          4127  !                (3)  IF A 'BEGIN RETRY' MESSAGE WAS PRINTED, THEN A FINAL
6092 ;          4128  !                    MESSAGE ABOUT THE SUCCESS OR FAILURE OF THE RETRIES
6093 ;          4129  !                    SHOULD ALSO BE PRINTED.
6094 ;          4130  !
6095 ;          4131  !                (4)  INCREMENT THE RETRY COUNTER FOR EVERY RETRY DONE
6096 ;          4132  !                    WHICH WAS NOT FOR ERROR CLASSIFICATION.
6097 ;          4133  !
6098 ;          4134  !
6099 ;          4135  !  ARGUMENTS:   (1)  TIMES - THE NUMBER OF RETRIES PERMITTED BEFORE
6100 ;          4136  !                           IT IS CALLED A FAILURE.
6101 ;          4137  !
6102 ;          4138  !                (2)  OTHER ARGUMENTS ARE THE SAME AS FOR 'START_IT'
6103 ;          4139  !                    ROUTINE ABOVE.
6104 ;          4140  !
6105 ;          4141  !  RESULTS:     THE VALUES RETURNED FOR THIS SUBROUTINE ARE THE SAME AS
6106 ;          4142  !                FOR THE SYSERR ROUTINE ABOVE.
6107 ;          4143  !--
6108 ;          4144
6109 ;          4145      label
6110 ;          4146          LOOP;
6111 ;          4147
6112 ;          4148      local
6113 ;          4149          TEMP,
6114 ;          4150          VALUE,
6115 ;          4151          COUNT,
6116 ;          4152          TBOARD,
6117 ;          4153          TBANK;
6118 ;          4154
6119 ;          4155      !+
6120 ;          4156      !  THIS IS THE CODE FOR PURPOSE 1:
6121 ;          4157      !-
6122 ;          4158
6123 ;          4159      RETRYING = ACTIVE;
6124 ;          4160
6125 ;          4161      if ((.TIMES neq 1) and (.ERROUT))
6126 ;          4162      then
6127 ;          4163          begin                                        !* 2 *
6128 ;          4164
```

```
;MLX4
;                ML11 TRANSFER COMMANDS                    27-Mar-1982 19:24:42    TOPS-20 Bliss-16 V2(212)
;                                                          27-Mar-1982 19:23:44    PA:<NEALE>MLX4.BLI.5 (33)

;        4165            if .COMMAND eql write then PRINTB (SAY3, WRD2, WRD16, WRD18);
;        4166
;        4167            !BEGIN WRITE RETRY
;        4168
;        4169            if .COMMAND eql CHECK then PRINTB (SAY3, WRD2, PHR1, WRD18);
;        4170
;        4171            !BEGIN WRITE CHECK RETRY
;        4172
;        4173            if .COMMAND eql read then PRINTB (SAY3, WRD2, WRD17, WRD18);
;        4174
;        4175            !BEGIN READ RETRY
;        4176            end;                                        !* 2 *
;        4177
;        4178        !+
;        4179        ! THIS IS THE CODE FOR PURPOSE 2:
;        4180        !-
;        4181
;        4182    LOOP :
;        4183        begin                                           !* 3 *
;        4184
;        4185        incr KOUNT from 1 to .TIMES do
;        4186            begin                                       !* 4 *
;        4187
;        4188            if .COMMAND eql write then VALUE = write (.LUN, .WRDCNT, .BUFFER, .SECTOR);
;        4189
;        4190            if .COMMAND eql CHECK
;        4191            then
;        4192                begin
;        4193                write (.LUN, .WRDCNT, .BUFFER, .SECTOR);
;        4194                VALUE = CHECK (.LUN, .WRDCNT, .BUFFER, .SECTOR);
;        4195                end;
;        4196
;        4197            if .COMMAND eql read
;        4198            then
;        4199                begin
;        4200                write (.LUN, .WRDCNT, (.BUFFER - BUFSIZ*2), .SECTOR);
;        4201                VALUE = read (.LUN, .WRDCNT, .BUFFER, .SECTOR);
;        4202                end;
;        4203
;        4204            !+
;        4205            ! THIS IS THE CODE FOR PURPOSE 3:
;        4206            !-
;        4207
;        4208            if .VALUE eql 0
;        4209            then                                                !THE RETRY WAS SUCCESSFUL
;        4210                begin                                           !* 5 *
;        4211
;        4212                if ((.TIMES neq 1) and (.ERROUT))
;        4213                then
;        4214                    begin
;        4215                    PRINTB (SAY2, WRD18, WRD19);
;        4216                    PRINTB (CRLF);
```

```
0186 ;MLX4
6187 ;
6188
6189 ;          4217                    !RETRY SUCCEEDED
6190 ;          4218                    end;
6191 ;          4219
6192 ;          4220                COUNT = .KOUNT;
6193 ;          4221                leave LOOP;
6194 ;          4222                end;                                    !* 5 *
6195 ;          4223
6196 ;          4224            end;                                        !* 4 *
6197 ;          4225
6198 ;          4226        !+
6199 ;          4227        ! FALLS THROUGH HERE IF ALL RETRIES FAILED:
6200 ;          4228        !-
6201 ;          4229
6202 ;          4230        if ((.TIMES neq 1) and (.ERROUT))
6203 ;          4231        then
6204 ;          4232            begin
6205 ;          4233            PRINTB (SAY2, WRD18, WRD20);
6206 ;          4234    !
6207 ;          4235    ! VER CZMLBB COMMENTED OUT THIS PRINT CRLF
6208 ;          4236    !       PRINTB (CRLF);
6209 ;          4237            !RETRY FAILED
6210 ;          4238            COUNT = .TIMES;
6211 ;          4239            end;
6212 ;          4240
6213 ;          4241        end;                                            !* 3 *
6214 ;          4242
6215 ;          4243        !+
6216 ;          4244        ! THIS IS THE CODE FOR PURPOSE 4:
6217 ;          4245        !-
6218 ;          4246
6219 ;          4247        if .TIMES neq 1
6220 ;          4248        then
6221 ;          4249            begin
6222 ;          4250            TBOARD = .BOARD;
6223 ;          4251            TBANK = .BANK;
6224 ;          4252            DECODE (.SECTOR);
6225 ;          4253            TRIES [.LUN, .BOARD, 0, 16, 0] = .TRIES [.LUN, .BOARD, 0, 16, 0] + .COUNT;
6226 ;          4254            BOARD = .TBOARD;
6227 ;          4255            BANK = .TBANK;
6228 ;          4256            end;
6229 ;          4257
6230 ;          4258        RETRYING = INACTIVE;
6231 ;          4259        return .VALUE;
6232 ;          4260        end;                                                !* 1 *
6236
6237
                                        .SBTTL  RETRY ML11 TRANSFER COMMANDS
```

```
                        ;MLX4
                        ;               ML11 TRANSFER COMMANDS                          27-Mar-1982 19:24:42    TOPS
                                                                                        27-Mar-1982 19:23:44    PA:<

046224 004167 137104    RETRY:  JSR     R1,$SAVE5                       ;                                       4113
046230 012767 000001 166102     MOV     #1,RETRYING                     ;                                       4159
046236 016601 000030            MOV     30(SP),R1                       ; TIMES,*                               4161
046242 005046                   CLR     -(SP)
046244 020127 000001            CMP     R1,#1
046250 001472                   BEQ     3$
046252 005216                   INC     (SP)
046254 032767 000001 133776     BIT     #1,ERROUT
046262 001465                   BEQ     3$
046264 016602 000030            MOV     30(SP),R2                       ; COMMAND,*                             4165
046270 020227 045424            CMP     R2,#WRITE
046274 001016                   BNE     1$
046276 012746 007272            MOV     #WRD18,-(SP)
046302 012746 007256            MOV     #WRD16,-(SP)
046306 012746 007200            MOV     #WRD2,-(SP)
046312 012746 007112            MOV     #SAY3,-(SP)
046316 012746 000004            MOV     #4,-(SP)
046322 010600                   MOV     SP,R0                           ; SP,*
046324 104414                   TRAP    14
046326 062706 000012            ADD     #12,SP
046332 020227 046000    1$:     CMP     R2,#CHECK                       ;                                       4169
046336 001016                   BNE     2$
046340 012746 007272            MOV     #WRD18,-(SP)
046344 012746 007714            MOV     #PHR1,-(SP)
046350 012746 007200            MOV     #WRD2,-(SP)
046354 012746 007112            MOV     #SAY3,-(SP)
046360 012746 000004            MOV     #4,-(SP)
046364 010600                   MOV     SP,R0                           ; SP,*
046366 104414                   TRAP    14
046370 062706 000012            ADD     #12,SP
046374 020227 045612    2$:     CMP     R2,#READ                        ;                                       4173
046400 001016                   BNE     3$
046402 012746 007272            MOV     #WRD18,-(SP)
046406 012746 007264            MOV     #WRD17,-(SP)
046412 012746 007200            MOV     #WRD2,-(SP)
046416 012746 007112            MOV     #SAY3,-(SP)
046422 012746 000004            MOV     #4,-(SP)
046426 010600                   MOV     SP,R0                           ; SP,*
046430 104414                   TRAP    14
046432 062706 000012            ADD     #12,SP
046436 016602 000030    3$:     MOV     30(SP),R2                       ; COMMAND,*                             4188
046442 005004                   CLR     R4                              ; KOUNT                                 4185
046444 000543                   BR      9$
046446 020227 045424    4$:     CMP     R2,#WRITE                       ;                                       4188
046452 001015                   BNE     5$
046454 016646 000026            MOV     26(SP),-(SP)                    ; LUN,*
046460 016646 000026            MOV     26(SP),-(SP)                    ; WRDCNT,*
046464 016646 000026            MOV     26(SP),-(SP)                    ; BUFFER,*
046470 016646 000026            MOV     26(SP),-(SP)                    ; SECTOR,*
046474 004767 176724            JSR     PC,WRITE
046500 010003                   MOV     R0,R3                           ; *,VALUE
046502 062706 000010            ADD     #10,SP
```

SEQ 0166

```
                                                                                27-Mar-1982 19:24:42    TOPS
                                                                                27-Mar-1982 19:23:44    PA:<
                        ;MLX4
                        ;                ML11 TRANSFER COMMANDS

6298
6299
6300
6301 046506 020227 046000   5$:   CMP   R2,#CHECK                ;                                    4190
6302 046512 001027            BNE   6$
6303 046514 016646 000026           MOV   26(SP),-(SP)           ; LUN,*                              4193
6304 046520 016646 000026           MOV   26(SP),-(SP)           ; WRDCNT,*
6305 046524 016646 000026           MOV   26(SP),-(SP)           ; BUFFER,*
6306 046530 016646 000026           MOV   26(SP),-(SP)           ; SECTOR,*
6307 046534 004767 176664           JSR   PC,WRITE
6308 046540 016616 000036           MOV   36(SP),(SP)            ; LUN,*                              4194
6309 046544 016646 000034           MOV   34(SP),-(SP)           ; WRDCNT,*
6310 046550 016646 000034           MOV   34(SP),-(SP)           ; BUFFER,*
6311 046554 016646 000034           MOV   34(SP),-(SP)           ; SECTOR,*
6312 046560 004767 177214           JSR   PC,CHECK
6313 046564 010003                  MOV   R0,R3                  ; *,VALUE
6314 046566 062706 000016           ADD   #16,SP                 ;
6315 046572 020227 045612   6$:   CMP   R2,#READ                ;                                    4192
6316 046576 001031            BNE   7$                                                               4197
6317 046600 016646 000026           MOV   26(SP),-(SP)           ; LUN,*                              4200
6318 046604 016646 000026           MOV   26(SP),-(SP)           ; WRDCNT,*
6319 046610 016646 000026           MOV   26(SP),-(SP)           ; BUFFER,*
6320 046614 162716 010000           SUB   #10000,(SP)
6321 046620 016646 000026           MOV   26(SP),-(SP)           ; SECTOR,*
6322 046624 004767 177574           JSR   PC,WRITE
6323 046630 016616 000036           MOV   36(SP),(SP)            ; LUN,*                              4201
6324 046634 016646 000034           MOV   34(SP),-(SP)           ; WRDCNT,*
6325 046640 016646 000034           MOV   34(SP),-(SP)           ; BUFFER,*
6326 046644 016646 000034           MOV   34(SP),-(SP)           ; SECTOR,*
6327 046650 004767 176736           JSR   PC,READ
6328 046654 010003                  MOV   R0,R3                  ; *,VALUE
6329 046656 062706 000016           ADD   #16,SP                 ;
6330 046662 005703          7$:   TST   R3                     ; VALUE                              4199
6331 046664 001033            BNE   9$                                                               4208
6332 046666 032716 000001           BIT   #1,(SP)                ;                                    4212
6333 046672 001426            BEQ   8$
6334 046674 032767 000001 133356    BIT   #1,ERROUT
6335 046702 001422            BEQ   8$
6336 046704 012746 007300           MOV   #WRD19,-(SP)           ;                                    4215
6337 046710 012746 007272           MOV   #WRD18,-(SP)
6338 046714 012746 007100           MOV   #SAY2,-(SP)
6339 046720 012746 000003           MOV   #3,-(SP)
6340 046724 010600                  MOV   SP,R0                  ; SP,*
6341 046726 104414                  TRAP  14
6342 046730 012716 007066           MOV   #CRLF,(SP)             ;                                    4216
6343 046734 012746 000001           MOV   #1,-(SP)
6344 046740 010600                  MOV   SP,R0                  ; SP,*
6345 046742 104414                  TRAP  14
6346 046744 062706 000012           ADD   #12,SP                 ;
6347 046750 010405          8$:   MOV   R4,R5                  ; KOUNT,COUNT                        4214
6348 046752 000427            BR    10$                                                              4220
6349 046754 005204          9$:   INC   R4                     ; KOUNT                              4221
6350 046756 020401            CMP   R4,R1                  ; KOUNT,*                            4185
6351 046760 003632            BLE   4$
6352 046762 032716 000001           BIT   #1,(SP)                ;                                    4230
```

```
                                        ;MLX4
                                        ;               ML11 TRANSFER COMMANDS                  27-Mar-1982 19:24:42    TOPS
                                                                                                27-Mar-1982 19:23:44    PA:<

046766  001454                                BEQ     11$
046770  032767  000001  133262                BIT     #1,ERROUT
046776  001415                                BEQ     10$
047000  012746  007312                        MOV     #WRD20,-(SP)            ;                                       4233
047004  012746  007272                        MOV     #WRD18,-(SP)
047010  012746  007100                        MOV     #SAY2,-(SP)
047014  012746  000003                        MOV     #3,-(SP)
047020  010600                                MOV     SP,R0                   ; SP,*
047022  104414                                TRAP    14
047024  010105                                MOV     R1,R5                   ; *,COUNT                               4238
047026  062706  000010                        ADD     #10,SP                  ;                                       4232
047032  032716  000001          10$:          BIT     #1,(SP)                 ;                                       4247
047036  001430                                BEQ     11$
047040  016701  165276                        MOV     BOARD,R1                ; *,TBOARD                              4250
047044  016702  165274                        MOV     BANK,R2                 ; *,TBANK                               4251
047050  016646  000020                        MOV     20(SP),-(SP)            ; SECTOR,*                              4252
047054  004767  170116                        JSR     PC,DECODE
047060  016600  000030                        MOV     30(SP),R0               ; LUN,*                                 4253
047064  006300                                ASL     R0
047066  006300                                ASL     R0
047070  006300                                ASL     R0
047072  006300                                ASL     R0
047074  066700  165242                        ADD     BOARD,R0
047100  006300                                ASL     R0
047102  060560  033714                        ADD     R5,TRIES(R0)            ; COUNT,*
047106  010167  165230                        MOV     R1,BOARD                ; TBOARD,*                              4254
047112  010267  165226                        MOV     R2,BANK                 ; TBANK,*                               4255
047116  005726                                TST     (SP)+                   ;                                       4249
047120  005067  165214          11$:          CLR     RETRYING                ;                                       4258
047124  010300                                MOV     R3,R0                   ; VALUE,*                               4114
047126  005726                                TST     (SP)+                   ;                                       4113
047130  000207                                RTS     PC

                                        ; Routine Size:  227 words
                                        ; Maximum stack depth per invocation:  14 words
```

```
 6399 ;MLX4
 6400 ;                  TO SET UP BUFFER POINTERS BEFORE A TRANSFER     27-Mar-1982 19:24:42    TOPS-20 Bliss-16 V2(212)
 6401                                                                    27-Mar-1982 19:23:44    PA:<NEALE>MLX4.BLI.5 (34)
 6402 ;          4261  %sbttl 'TO SET UP BUFFER POINTERS BEFORE A TRANSFER'
 6403 ;          4262  routine SET_PTRS (WRDCNT) : novalue =
 6404 ;          4263      begin
 6405 ;          4264
 6406 ;          4265  !++
 6407 ;          4266  !   ROUTINE:      SET_PTRS(WRDCNT)
 6408 ;          4267  !
 6409 ;          4268  !   PURPOSE:      TO MAKE SURE THAT THE BUFFER POINTERS ARE SUITABLY PLACED
 6410 ;          4269  !                 BEFORE A TRANSFER, SO THAT THEY WILL SUPPORT THE CHOSEN WORD
 6411 ;          4270  !                 COUNT WITHIN THE BUFFER SPACE.
 6412 ;          4271  !
 6413 ;          4272  !   ARGUMENT:     WRDCNT = THE NUMBER OF WORDS IN A TRANSFER.
 6414 ;          4273  !
 6415 ;          4274  !   RESULTS:      'WPTR' AND 'RPTR' ARE SET BACK TO THE START OF THE BUFFERS
 6416 ;          4275  !                 ONLY IF THE WORD COUNT WON'T FIT.  OTHERWISE, THEY ARE LEFT
 6417 ;          4276  !                 UNCHANGED.
 6418 ;          4277  !
 6419 ;          4278  !                 NOTE:  THE WRITE BUFFER IS LOCATED BEFORE THE READ BUFFER.
 6420 ;          4279  !--
 6421 ;          4280
 6422 ;          4281      if (.WPTR + .WRDCNT*2) geqa END_WBUFF then WPTR = WBUFF;
 6423 ;          4282
 6424 ;          4283      RPTR = .WPTR + (BUFSIZ*2);
 6425 ;          4284      return;
 6426 ;          4285      end;
 6430
 6431                                  .SBTTL  SET.PTRS TO SET UP BUFFER POINTERS BEFORE A TRANSFER
 6435 047132                      SET.PTRS:
 6436 047132  016600  000002              MOV     2(SP),R0                        ; WRDCNT,*      4281
 6437 047136  006300                      ASL     R0
 6438 047140  066700  163524              ADD     WPTR,R0
 6439 047144  020027  022670              CMP     R0,#END.WBUFF
 6440 047150  103403                      BLO     1$
 6441 047152  012767  012670  163510      MOV     #WBUFF,WPTR
 6442 047160  016700  163504      1$:     MOV     WPTR,R0                         ;               4283
 6443 047164  062700  010000              ADD     #10000,R0
 6444 047170  010067  163476              MOV     R0,RPTR
 6445 047174  000207                      RTS     PC                              ;               4262
 6446
 6447                             ; Routine Size:  18 words
 6448                             ; Maximum stack depth per invocation:  0 words
```

27-Mar-1982 19:24:42 TOPS-20 Bliss-16 V2(212)
27-Mar-1982 19:23:44 PA:<NEALE>MLX4.BLI.5 (35)

```
6454 ;MLX4
6455 ;                  CHOOSING A WORD COUNT
6456
6457 ;          4286   %sbttl 'CHOOSING A WORD COUNT'
6458 ;          4287   routine GET_WRDCNT (SECTOR, LAST) =
6459 ;          4288       begin
6460 ;          4289
6461 ;          4290       local
6462 ;          4291           TEMP,
6463 ;          4292           WRDCNT;
6464 ;          4293
6465 ;          4294       TEMP = .LAST - .SECTOR + 1;                !NUMBER OF SECTORS LEFT TO TEST
6466 ;          4295
6467 ;          4296   !
6468 ;          4297   ! version czmlbb changed gtr to gtru
6469 ;          4298   !
6470 ;          4299       if (BUFSIZ/256) gtru .TEMP                 !LESS THAN A FULL BUFFER LEFT?
6471 ;          4300       then
6472 ;          4301           WRDCNT = .TEMP*256                     !YES -- USE AS LARGE A COUNT AS FITS
6473 ;          4302       else
6474 ;          4303           WRDCNT = BUFSIZ;                       !NO  -- USE THE ENTIRE BUFFER SIZE
6475 ;          4304
6476 ;          4305       return .WRDCNT;
6477 ;          4306       end;
6481
6482                                .SBTTL  GET.WRDCNT CHOOSING A WORD COUNT
6486 047176                  GET.WRDCNT:
6487 047176  016600  000002          MOV     2(SP),R0                   ; LAST,*                    4294
6488 047202  166600  000004          SUB     4(SP),R0                   ; SECTOR,*
6489 047206  005200                  INC     R0
6490 047210  020027  000010          CMP     R0,#10                     ; TEMP,*                    4299
6491 047214  103003                  BHIS    1$
6492 047216  000300                  SWAB    R0                         ;                           4301
6493 047220  105000                  CLRB    R0
6494 047222  000207                  RTS     PC                         ;                           4299
6495 047224  012700  004000    1$:   MOV     #4000,R0                   ; *,WRDCNT                  4303
6496 047230  000207                  RTS     PC                         ;                           4287
6497
6498                           ; Routine Size:  14 words
6499                           ; Maximum stack depth per invocation:  0 words
6504
6505
```

```
;MLX4
;                COMMAND INTEGRITY ROUTINE                          27-Mar-1982 19:24:42    TOPS-20 Bliss-16 V2(212)
;                                                                   27-Mar-1982 19:23:44    PA:<NEALE>MLX4.BLI.5 (36)
;
;        4307  %sbttl 'COMMAND INTEGRITY ROUTINE'
;        4308  routine INTEGRITY : novalue =
;        4309      begin                                            !* 1 *  START OF ROUTINE
;        4310
;        4311  !++
;        4312  !  ROUTINE:     INTEGRITY
;        4313  !
;        4314  !  PURPOSE:     TO MAKE SURE THAT THE BASIC ML11 TRANSFER COMMANDS
;        4315  !               WHICH WILL BE USED BY THE EXERCISER (WRITE,READ,
;        4316  !               WRITE CHECK) WORK PROPERLY.
;        4317  !
;        4318  !  THE CODE FOR 'INTEGRITY' IN BRIEF:
;        4319  !
;        4320  !  BEGIN 1 (START OF ROUTINE)
;        4321  !  SAY ROUTINE IS RUNNING
;        4322  !  INCR COMPLEMENT FLAG FROM 0 TO 1
;        4323  !  : BEGIN 2 (START OF COMPLEMENT FLAG SELECTION LOOP)
;        4324  !  : GENERATE THE PATTERN
;        4325  !  : INCR LOGICAL UNIT NUMBER FROM 0 TO LAST
;        4326  !  : : BEGIN 3 (START OF LOGICAL UNIT SELECTION LOOP)
;        4327  !  : : TESTLOOP:
;        4328  !  : : : BEGIN 4 (START OF LOOP THAT COMPLETELY TESTS 1 UNIT)
;        4329  !  : : : IF UNIT IS ACTIVE
;        4330  !  : : : THEN
;        4331  !  : : : : BEGIN 5 (START OF TEST FOR AN ACTIVE UNIT)
;        4332  !  : : : : SECTOR = LOWEST
;        4333  !  : : : : GET WRDCNT
;        4334  !  : : : : WRITE
;        4335  !  : : : : LOOK FOR ERRORS (IF DROP UNIT, LEAVE TESTLOOP)
;        4336  !  : : : : READ
;        4337  !  : : : : LOOK FOR ERRORS (IF DROP UNIT, LEAVE TESTLOOP)
;        4338  !  : : : : WRITE CHECK
;        4339  !  : : : : LOOK FOR ERRORS (IF DROP UNIT, LEAVE TESTLOOP)
;        4340  !  : : : : END 5 (END OF TEST FOR AN ACTIVE UNIT)
;        4341  !  : : : END 4 (END OF LOOP THAT COMPLETELY TESTS 1 UNIT)
;        4342  !  : : END 3 (END OF LOGICAL UNIT SELECTION LOOP)
;        4343  !  : END 2 (END OF COMPLEMENT FLAG SELECTION LOOP)
;        4344  !  RETURN
;        4345  !  END 1 (END OF ROUTINE)
;        4346  !--
;        4347
;        4348      label
;        4349          LOOP;
;        4350
;        4351      local
;        4352          VALUE,
;        4353          WRDCNT,
;        4354          OLDSEC,
;        4355          OLDCHN,
;        4356          SECTOR,
;        4357          DBL_VALUE;
;        4358
```

```
4359      PRINTB (SAY2, WRD34, RTN0);
4360      !'RUNNING COMMAND INTEGRITY ROUTINE'
4361
4362      incr COMP_FLAG from 0 to 1 do
4363          begin                                   !* 2 *  START OF COMPLEMENT FLAG SELECTION LOOP
4364          GEN3 (.COMP_FLAG);
4365
4366          incr LUN from 0 to (.L$UNIT - 1) do
4367              begin                               !* 3 *  START OF LOGICAL UNIT SELECTION LOOP
4368      LOOP :
4369              begin                               !* 4 *  START OF LOOP THAT COMPLETELY TESTS 1 UNIT
4370
4371              if .DRIVE_STATUS [.LUN] eql ACTIVE
4372              then
4373                  begin                           !* 5 *  START OF TEST FOR AN ACTIVE UNIT
4374                  L$LUN = .LUN;
4375                  SECTOR = LOWEST;
4376                  WRDCNT = GET_WRDCNT (.SECTOR, HIGHEST);
4377                  VALUE = write (.LUN, .WRDCNT, WBUFF, LOWEST);
4378
4379                  !+
4380                  !  SEE HOW SUCCESSFUL THE WRITE WAS:
4381                  !-
4382
4383                  selectone .VALUE of             !SEE 'SYSERR' FOR DEFINITION
4384                      set                         !OF ERROR # CONTAINED IN 'VALUE'
4385
4386                      [1] :
4387                          begin                   !* 5A * RETRY ALLOWED
4388
4389                      if RETRY (SIX, write, .LUN, .WRDCNT, WBUFF, LOWEST) neq 0
4390                      then                        !THE RETRY FAILED -- SYSTEM FATAL ERROR
4391                          begin
4392                          WHY_DROPT [.LUN] = CODE_4;
4393                          ERRDF (1, MSG1, 0); !**** INTEGRITY ROUTINE ERROR 01 ****
4394                          DODU (.LUN);
4395                          leave LOOP;             !JUMP JUST BEYOND END OF BLOCK * 4 *
4396                          end;
4397
4398                      end;                        !* 5A *
4399
4400                  [2] :
4401                      begin                       !* 5B * FATAL CONTROLLER ERROR -- NO RETRY ALLOWED
4402                      WHY_DROPT [.LUN] = CODE_5;
4403                      ERRDF (2, MSG1, 0);         !**** INTEGRITY ROUTINE ERROR 02 ****
4404                      DODU (.LUN);
4405                      leave LOOP;                 !JUMP JUST BEYOND END OF BLOCK * 4 *
4406                      end;                        !* 5B *
4407
4408                  [3] :
4409                      begin                       !* 5C * FATAL DRIVE ERROR -- NO RETRY ALLOWED
4410                      ERRDF (3, MSG1, 0);         !**** INTEGRITY ROUTINE ERROR 03 ****
```

```
6622 ;  4411                    WHY_DROPT [.LUN] = CODE_6;
6623 ;  4412                    DODO (.LUN);
6624 ;  4413                    leave LOOP;             !JUMP JUST BEYOND END OF BLOCK * 4 *
6625 ;  4414                    end;                    !* 5C *
6626 ;  4415                tes;
6627 ;  4416
6628 ;  4417            VALUE = read (.LUN, .WRDCNT, RBUFF, LOWEST);
6629 ;  4418
6630 ;  4419            !+
6631 ;  4420            !  SEE HOW SUCCESSFUL THE READ WAS:
6632 ;  4421            !-
6633 ;  4422
6634 ;  4423            selectone .VALUE of             !SEE 'SYSERR' FOR DEFINITION
6635 ;  4424                set                         !OF ERROR # CONTAINED IN 'VALUE'
6636 ;  4425
6637 ;  4426                [0] :
6638 ;  4427
6639 ;  4428                    if (DBL_VALUE = DOUBLE_CHECK (WBUFF, RBUFF, .WRDCNT)) neq 0
6640 ;  4429                    then
6641 ;  4430                        begin
6642 ;  4431                        SAYWHO (.LUN);
6643 ;  4432                        PRINTB (SAY1, MSG5);
6644 ;  4433                        !'ECC LOGIC FAILED TO DETECT DATA ERROR'
6645 ;  4434                        PRINTB (FMT12A, ..DBL_VALUE, .DBL_VALUE);
6646 ;  4435                        !'GOOD DATA: XXXXXX AT LOCATION YYYYYY'
6647 ;  4436                        DBL_VALUE = .DBL_VALUE + BUFSIZ*2;
6648 ;  4437                        PRINTB (FMT12B, ..DBL_VALUE, .DBL_VALUE);
6649 ;  4438                        !'BAD DATA:  PPPPPP AT LOCATION QQQQQQ'
6650 ;  4439                        WHY_DROPT [.LUN] = CODE_8;
6651 ;  4440                        ERRDF (4, MSG1, 0); !**** INTEGRITY ROUTINE ERROR 04 ****
6652 ;  4441                        DODU (.LUN);
6653 ;  4442                        leave LOOP;         !JUMP JUST BEYOND END OF BLOCK * 4 *
6654 ;  4443                        end;
6655 ;  4444
6656 ;  4445                [1] :
6657 ;  4446                    begin                   !* 5D * RETRY ALLOWED
6658 ;  4447
6659 ;  4448                    if RETRY (SIX, read, .LUN, .WRDCNT, RBUFF, LOWEST) neq 0
6660 ;  4449                    then                    !THE RETRY FAILED -- SYSTEM FATAL ERROR
6661 ;  4450                        begin
6662 ;  4451                        WHY_DROPT [.LUN] = CODE_4;
6663 ;  4452                        ERRDF (5, MSG1, 0); !**** INTEGRITY ROUTINE ERROR 05 ****
6664 ;  4453                        DODU (.LUN);
6665 ;  4454                        leave LOOP;         !JUMP JUST BEYOND END OF BLOCK * 4 *
6666 ;  4455                        end;
6667 ;  4456
6668 ;  4457                    end;                    !* 5D *
6669 ;  4458
6670 ;  4459                [2] :
6671 ;  4460                    begin                   !* 5E * FATAL CONTROLLER ERROR -- NO RETRY ALLOWED
6672 ;  4461                    WHY_DROPT [.LUN] = CODE_5;
6673 ;  4462                    ERRDF (6, MSG1, 0);     !**** INTEGRITY ROUTINE ERROR 06 ****
```

```
;MLX4
;                    COMMAND INTEGRITY ROUTINE                    27-Mar-1982 19:24:42    TOPS-20 Bliss-16 V2(212)
;                                                                 27-Mar-1982 19:23:44    PA:<NEALE>MLX4.BLI.5 (36)

;          4463                         DODU (.LUN);
;          4464                         leave LOOP;                !JUMP JUST BEYOND END OF BLOCK * 4 *
;          4465                         end;                       !* 5E *
;          4466
;          4467                     [3] :
;          4468                         begin                      !* 5F * FATAL DRIVE ERROR -- NO RETRY ALLOWED
;          4469                         ERRDF (7, MSG1, 0);        !**** INTEGRITY ROUTINE ERROR 07 ****
;          4470                         WHY_DROPT [.LUN] = CODE_6;
;          4471                         DODU (.LUN);
;          4472                         leave LOOP;                !JUMP JUST BEYOND END OF BLOCK * 4 *
;          4473                         end;                       !* 5F *
;          4474
;          4475                     [4] :
;          4476                         begin                      !* 5G * UNRECOVERABLE DATA ERROR
;          4477                         ISOLATE ();
;          4478                         ERRDF (8, MSG2, 0);        !**** INTEGRITY ROUTINE ERROR 08 ****
;          4479                         WHY_DROPT [.LUN] = CODE_7;
;          4480                         DODU (.LUN);
;          4481                         leave LOOP;                !JUMP JUST BEYOND END OF BLOCK * 4 *
;          4482                         end;                       !* 5G *
;          4483
;          4484                     [5] :
;          4485                         begin                      !* 5H * RECOVERABLE DATA ERROR
;          4486                         ISOLATE ();
;          4487
;          4488                         if .ERROUT then PRINTB (FMT10B, .CHAN);
;          4489
;          4490                         !'    BIT 00'
;          4491                         OLDSEC = .MLEL;
;          4492                         OLDCHN = .CHAN;
;          4493
;          4494                         if RETRY (ONE, read, .LUN, .WRDCNT, RBUFF, LOWEST) eql 5
;          4495                         then
;          4496
;          4497                             if ((.MLEL eql .OLDSEC) and (.CHAN eql .OLDCHN))
;          4498                             then
;          4499                                 begin
;          4500
;          4501                                 if .ERROUT then ERRHRD (9, MSG4, 0);
;          4502
;          4503                                                    !**** INTEGRITY ROUTINE ERROR 09 ****
;          4504                                 UP_HARD_COUNT (.LUN, .BOARD);
;          4505                                 end
;          4506                             else
;          4507                                 begin
;          4508
;          4509                                 if .ERROUT then ERRSOFT (10, MSG3, 0);
;          4510
;          4511                                                    !**** INTEGRITY ROUTINE ERROR 10 ****
;          4512                                 UP_SOFT_COUNT (.LUN, .BOARD);
;          4513                                 end
;          4514
```

```
6731 ;MLX4
6732 ;
6733
6734 ;          4515                        else
6735 ;          4516                            begin
6736 ;          4517
6737 ;          4518                            if .ERROUT then ERRSOFT (11, MSG3, 0);
6738 ;          4519
6739 ;          4520                                                    !**** INTEGRITY ROUTINE ERROR 11 ****
6740 ;          4521                            UP_SOFT_COUNT (.LUN, .BOARD);
6741 ;          4522                            end;
6742 ;          4523
6743 ;          4524                        end;                        !* 5H *
6744 ;          4525                tes;
6745 ;          4526
6746 ;          4527            VALUE = CHECK (.LUN, .WRDCNT, WBUFF, LOWEST);
6747 ;          4528
6748 ;          4529            !+
6749 ;          4530            !  SEE HOW SUCCESSFUL THE WRITE CHECK WAS:
6750 ;          4531            !-
6751 ;          4532
6752 ;          4533            selectone .VALUE of             !SEE 'SYSERR' FOR DEFINITION
6753 ;          4534                set                         !OF ERROR # CONTAINED IN 'VALUE'
6754 ;          4535
6755 ;          4536                [1] :
6756 ;          4537                    begin                   !* 5I * RETRY ALLOWED
6757 ;          4538
6758 ;          4539                    if RETRY (SIX, CHECK, .LUN, .WRDCNT, WBUFF, LOWEST) neq 0
6759 ;          4540                    then                    !THE RETRY FAILED -- SYSTEM FATAL ERROR
6760 ;          4541                        begin
6761 ;          4542                        WHY_DROPT [.LUN] = CODE_4;
6762 ;          4543                        ERRDF (12, MSG1, 0);        !**** INTEGRITY ROUTINE ERROR 12 ****
6763 ;          4544                        DODU (.LUN);
6764 ;          4545                        leave LOOP;             !JUMP JUST BEYOND END OF BLOCK * 4 *
6765 ;          4546                        end;
6766 ;          4547
6767 ;          4548                    end;                        !* 5I *
6768 ;          4549
6769 ;          4550                [2] :
6770 ;          4551                    begin                       !* 5J * FATAL CONTROLLER ERROR -- NO RETRY ALLOWED
6771 ;          4552                    WHY_DROPT [.LUN] = CODE_5;
6772 ;          4553                    ERRDF (13, MSG1, 0);        !**** INTEGRITY ROUTINE ERROR 13 ****
6773 ;          4554                    DODU (.LUN);
6774 ;          4555                    leave LOOP;                 !JUMP JUST BEYOND END OF BLOCK * 4 *
6775 ;          4556                    end;                        !* 5J *
6776 ;          4557
6777 ;          4558                [3] :
6778 ;          4559                    begin                       !* 5K * FATAL DRIVE ERROR -- NO RETRY ALLOWED
6779 ;          4560                    ERRDF (14, MSG1, 0);        !**** INTEGRITY ROUTINE ERROR 14 ****
6780 ;          4561                    WHY_DROPT [.LUN] = CODE_6;
6781 ;          4562                    DODU (.LUN);
6782 ;          4563                    leave LOOP;                 !JUMP JUST BEYOND END OF BLOCK * 4 *
6783 ;          4564                    end;                        !* 5K *
6784 ;          4565
6785 ;          4566                [4] :
```

```
;MLX4
;                    COMMAND INTEGRITY ROUTINE                    27-Mar-1982 19:24:42    TOPS-20 Bliss-16 V2(212)
;                                                                 27-Mar-1982 19:23:44    PA:<NEALE>MLX4.BLI.5 (36)

;          4567                          begin                    !* 5L * UNRECOVERABLE DATA ERROR
;          4568                          ISOLATE ();
;          4569                          ERRDF (15, MSG2, 0);     !**** INTEGRITY ROUTINE ERROR 15 ****
;          4570                          WHY_DROPT [.LUN] = CODE_7;
;          4571                          DODU (.LUN);
;          4572                          leave LOOP;              !JUMP JUST BEYOND END OF BLOCK * 4 *
;          4573                          end;                     !* 5L *
;          4574
;          4575                      [5] :
;          4576                          begin                    !* 5M * RECOVERABLE DATA ERROR
;          4577                          ISOLATE ();
;          4578
;          4579                          if .ERROUT then PRINTB (FMT10B, .CHAN);
;          4580
;          4581                          !'    BIT 00'
;          4582                          OLDSEC = .MLEL;
;          4583                          OLDCHN = .CHAN;
;          4584
;          4585                          if RETRY (ONE, CHECK, .LUN, .WRDCNT, WBUFF, LOWEST) eql 5
;          4586                          then
;          4587
;          4588                              if ((.MLEL eql .OLDSEC) and (.CHAN eql .OLDCHN))
;          4589                              then
;          4590                                  begin
;          4591
;          4592                                  if .ERROUT then ERRHRD (16, MSG4, 0);
;          4593
;          4594                                                   !**** INTEGRITY ROUTINE ERROR 16 ****
;          4595                                  UP_HARD_COUNT (.LUN, .BOARD);
;          4596                                  end
;          4597                              else
;          4598                                  begin
;          4599
;          4600                                  if .ERROUT then ERRSOFT (17, MSG3, 0);
;          4601
;          4602                                                   !**** INTEGRITY ROUTINE ERROR 17 ****
;          4603                                  UP_SOFT_COUNT (.LUN, .BOARD);
;          4604                                  end
;          4605
;          4606                          else
;          4607                              begin
;          4608
;          4609                              if .ERROUT then ERRSOFT (18, MSG3, 0);
;          4610
;          4611                                                   !**** INTEGRITY ROUTINE ERROR 18 ****
;          4612                              UP_SOFT_COUNT (.LUN, .BOARD);
;          4613                              end;
;          4614
;          4615                          end;                     !* 5M *
;          4616                  tes;
;          4617
;          4618              end;                                 !* 5 *  END OF TEST FOR AN ACTIVE DRIVE
```

```
;MLX4
;                    COMMAND INTEGRITY ROUTINE                    27-Mar-1982 19:24:42    TOPS-20 Bliss-16 V2(212)
;                                                                 27-Mar-1982 19:23:44    PA:<NEALE>MLX4.BLI.5 (36)

;          4619
;          4620              end;                                 !* 4 *  END OF LOOP THAT COMPLETELY TESTS 1 UNIT
;          4621              end;                                 !* 3 *  END OF LOGICAL UNIT SELECTION LOOP
;          4622
;          4623          end;                                     !* 2 *  END OF COMPLEMENT FLAG SELECTION LOOP
;          4624
;          4625      return;
;          4626      end;                                         !* 1 *  END OF ROUTINE

                                    .SBTTL  INTEGRITY COMMAND INTEGRITY ROUTINE
047232                          INTEGRITY:
047232  004167  136076                 JSR     R1,$SAVE5                       ;                       4308
047236  162706  000012                 SUB     #12,SP
047242  012746  007574                 MOV     #RTN0,-(SP)                     ;                       4359
047246  012746  007344                 MOV     #WRD34,-(SP)
047252  012746  007100                 MOV     #SAY2,-(SP)
047256  012746  000003                 MOV     #3,-(SP)
047262  010600                        MOV     SP,R0                           ; SP,*
047264  104414                        TRAP    14
047266  005001                        CLR     R1                              ; COMP.FLAG             4362
047270  010146                 1$:    MOV     R1,-(SP)                        ; COMP.FLAG,*           4364
047272  004767  172170                 JSR     PC,GEN3
047276  016766  132510  000020        MOV     L$UNIT,20(SP)                   ;                       4366
047304  005004                        CLR     R4                              ; LUN
047306  000167  002110                 JMP     38$
047312  010400                 2$:    MOV     R4,R0                           ; LUN,*                 4371
047314  006200                        ASR     R0
047316  006200                        ASR     R0
047320  006200                        ASR     R0
047322  062700  034442                 ADD     #DRIVE.STATUS,R0
047326  010046                        MOV     R0,-(SP)
047330  010446                        MOV     R4,-(SP)                        ; LUN,*
047332  042716  177770                 BIC     #177770,(SP)
047336  012746  000001                 MOV     #1,-(SP)
047342  005046                        CLR     -(SP)
047344  004767  135006                 JSR     PC,BL$GT2
047350  062706  000010                 ADD     #10,SP
047354  005300                        DEC     R0
047356  001117                        BNE     6$
047360  010467  132510                 MOV     R4,L$LUN                        ; LUN,*                 4374
047364  010400                        MOV     R4,R0                           ; LUN,*                 4375
047366  006300                        ASL     R0
047370  012702  034460                 MOV     #LOW.SECT,R2
047374  060002                        ADD     R0,R2
047376  011266  000022                 MOV     (R2),22(SP)                     ; *,SECTOR
047402  016646  000022                 MOV     22(SP),-(SP)                    ; SECTOR,*              4376
```

```
                                        ;MLX4                                                                   27-Mar-1982 19:24:42    TOPS
                                        ;               COMMAND INTEGRITY ROUTINE                               27-Mar-1982 19:23:44    PA:<
047406  016046  034500                        MOV     TOP.SECT(R0),-(SP)
047412  004767  177560                        JSR     PC,GET.WRDCNT
047416  010003                                MOV     R0,R3                           ; *,WRDCNT
047420  010416                                MOV     R4,(SP)                         ; LUN,*                 4377
047422  010346                                MOV     R3,-(SP)                        ; WRDCNT,*
047424  012746  012670                        MOV     #WBUFF,-(SP)
047430  011246                                MOV     (R2),-(SP)
047432  004767  175766                        JSR     PC,WRITE
047436  010005                                MOV     R0,R5                           ; *,VALUE
047440  020527  000001                        CMP     R5,#1                           ; VALUE,*               4383
047444  001031                                BNE     3$
047446  012746  000006                        MOV     #6,-(SP)                        ;                       4389
047452  012746  045424                        MOV     #WRITE,-(SP)
047456  010446                                MOV     R4,-(SP)                        ; LUN,*
047460  010346                                MOV     R3,-(SP)                        ; WRDCNT,*
047462  012746  012670                        MOV     #WBUFF,-(SP)
047466  011246                                MOV     (R2),-(SP)
047470  004767  176530                        JSR     PC,RETRY
047474  062706  000014                        ADD     #14,SP
047500  005700                                TST     R0
047502  001446                                BEQ     7$
047504  112764  000004  034446                MOVB    #4,WHY.DROPT(R4)                ; *,*(LUN)              4392
047512  104455                                TRAP    55                              ;                       4393
047514  000001                                .WORD   1
047516  011070                                .WORD   MSG1
047520  000000                                .WORD   0
047522  010400                                MOV     R4,R0                           ; LUN,*                 4394
047524  104451                                TRAP    51
047526  000431                                BR      5$                              ;                       4395
047530  020527  000002            3$:         CMP     R5,#2                           ; VALUE,*               4383
047534  001012                                BNE     4$
047536  112764  000005  034446                MOVB    #5,WHY.DROPT(R4)                ; *,*(LUN)              4402
047544  104455                                TRAP    55                              ;                       4403
047546  000002                                .WORD   2
047550  011070                                .WORD   MSG1
047552  000000                                .WORD   0
047554  010400                                MOV     R4,R0                           ; LUN,*                 4404
047556  104451                                TRAP    51
047560  000414                                BR      5$                              ;                       4405
047562  020527  000003            4$:         CMP     R5,#3                           ; VALUE,*               4383
047566  001014                                BNE     7$
047570  104455                                TRAP    55                              ;                       4410
047572  000003                                .WORD   3
047574  011070                                .WORD   MSG1
047576  000000                                .WORD   0
047600  112764  000006  034446                MOVB    #6,WHY.DROPT(R4)                ; *,*(LUN)              4411
047606  010400                                MOV     R4,R0                           ; LUN,*                 4412
047610  104451                                TRAP    51
047612  062706  000012            5$:         ADD     #12,SP                          ;                       4413
047616  000502                    6$:         BR      8$
047620  010416                    7$:         MOV     R4,(SP)                         ; LUN,*                 4417
047622  010346                                MOV     R3,-(SP)                        ; WRDCNT,*
```

```
                        ;MLX4
                        ;                COMMAND INTEGRITY ROUTINE                    27-Mar-1982 19:24:42    TOPS
                        ;                                                             27-Mar-1982 19:23:44    PA:<
 6955
 6956
 6957
 6958 047624 012746 022670          MOV     #RBUFF,-(SP)
 6959 047630 011246                 MOV     (R2),-(SP)
 6960 047632 004767 175754          JSR     PC,READ
 6961 047636 010005                 MOV     R0,R5                   ; *,VALUE
 6962 047640 001072                 BNE     9$                      ;
 6963 047642 012746 012670          MOV     #WBUFF,-(SP)            ;                                        4423
 6964 047646 012746 022670          MOV     #RBUFF,-(SP)                                                     4428
 6965 047652 010346                 MOV     R3,-(SP)                ; WRDCNT,*
 6966 047654 004767 174446          JSR     PC,DOUBLE.CHECK
 6967 047660 062706 000006          ADD     #6,SP
 6968 047664 010066 000032          MOV     R0,32(SP)               ; *,DBL.VALUE
 6969 047670 001477                 BEQ     10$
 6970 047672 010446                 MOV     R4,-(SP)                ; LUN,*                                  4431
 6971 047674 004767 165624          JSR     PC,SAYWHO
 6972 047700 012716 011216          MOV     #MSG5,(SP)              ;                                        4432
 6973 047704 012746 007072          MOV     #SAY1,-(SP)
 6974 047710 012746 000002          MOV     #2,-(SP)
 6975 047714 010600                 MOV     SP,R0                   ; SP,*
 6976 047716 104414                 TRAP    14
 6977 047720 016616 000040          MOV     40(SP),(SP)             ; DBL.VALUE,*                            4434
 6978 047724 017646 000040          MOV     @40(SP),-(SP)           ; DBL.VALUE,*
 6979 047730 012746 006710          MOV     #FMT12A,-(SP)
 6980 047734 012746 000003          MOV     #3,-(SP)
 6981 047740 010600                 MOV     SP,R0                   ; SP,*
 6982 047742 104414                 TRAP    14
 6983 047744 062766 010000 000046   ADD     #10000,46(SP)           ; *,DBL.VALUE                            4436
 6984 047752 016616 000046          MOV     46(SP),(SP)             ; DBL.VALUE,*                            4437
 6985 047756 017646 000046          MOV     @46(SP),-(SP)           ; DBL.VALUE,*
 6986 047762 012746 006756          MOV     #FMT12B,-(SP)
 6987 047766 012746 000003          MOV     #3,-(SP)
 6988 047772 010600                 MOV     SP,R0                   ; SP,*
 6989 047774 104414                 TRAP    14
 6990 047776 112764 000010 034446   MOVB    #10,WHY.DROPT(R4)       ; *,*(LUN)                               4439
 6991 050004 104455                 TRAP    55                      ;                                        4440
 6992 050006 000004                 .WORD   4
 6993 050010 011070                 .WORD   MSG1
 6994 050012 000000                 .WORD   0
 6995 050014 010400                 MOV     R4,R0                   ; LUN,*                                  4441
 6996 050016 104451                 TRAP    51
 6997 050020 062706 000042          ADD     #42,SP                  ;                                        4442
 6998 050024 000510                 BR      16$                                                              4423
 6999 050026 020527 000001  9$:     CMP     R5,#1                   ; VALUE,*                                4448
 7000 050032 001033                 BNE     12$
 7001 050034 012746 000006          MOV     #6,-(SP)                ;
 7002 050040 012746 045612          MOV     #READ,-(SP)
 7003 050044 010446                 MOV     R4,-(SP)                ; LUN,*
 7004 050046 010346                 MOV     R3,-(SP)                ; WRDCNT,*
 7005 050050 012746 022670          MOV     #RBUFF,-(SP)
 7006 050054 011246                 MOV     (R2),-(SP)
 7007 050056 004767 176142          JSR     PC,RETRY
 7008 050062 062706 000014          ADD     #14,SP
 7009 050066 005700                 TST     R0
```

```
                                        ;MLX4
                                        ;                COMMAND INTEGRITY ROUTINE                    27-Mar-1982 19:24:42   TOPS
                                        ;                                                             27-Mar-1982 19:23:44   PA:<
7011
7012
7013
7014 050070  001002                10$:  BNE     11$
7015 050072  000167  000520              JMP     24$
7016 050076  112764  000004  034446  11$:  MOVB    #4,WHY.DROPT(R4)      ; *,*(LUN)                    4451
7017 050104  104455                       TRAP    55                    ;                             4452
7018 050106  000005                       .WORD   5
7019 050110  011070                       .WORD   MSG1
7020 050112  000000                       .WORD   0
7021 050114  010400                       MOV     R4,R0                 ; LUN,*                       4453
7022 050116  104451                       TRAP    51
7023 050120  000450                       BR      15$                   ;                             4454
7024 050122  020527  000002        12$:   CMP     R5,#2                 ; VALUE,*                     4423
7025 050126  001012                       BNE     13$
7026 050130  112764  000005  034446       MOVB    #5,WHY.DROPT(R4)      ; *,*(LUN)                    4461
7027 050136  104455                       TRAP    55                    ;                             4462
7028 050140  000006                       .WORD   6
7029 050142  011070                       .WORD   MSG1
7030 050144  000000                       .WORD   0
7031 050146  010400                       MOV     R4,R0                 ; LUN,*                       4463
7032 050150  104451                       TRAP    51
7033 050152  000433                       BR      15$                   ;                             4464
7034 050154  020527  000003        13$:   CMP     R5,#3                 ; VALUE,*                     4423
7035 050160  001012                       BNE     14$
7036 050162  104455                       TRAP    55                    ;                             4469
7037 050164  000007                       .WORD   7
7038 050166  011070                       .WORD   MSG1
7039 050170  000000                       .WORD   0
7040 050172  112764  000006  034446       MOVB    #6,WHY.DROPT(R4)      ; *,*(LUN)                    4470
7041 050200  010400                       MOV     R4,R0                 ; LUN,*                       4471
7042 050202  104451                       TRAP    51
7043 050204  000416                       BR      15$                   ;                             4472
7044 050206  020527  000004        14$:   CMP     R5,#4                 ; VALUE,*                     4423
7045 050212  001017                       BNE     17$
7046 050214  004767  167112               JSR     PC,ISOLATE            ;                             4477
7047 050220  104455                       TRAP    55                    ;                             4478
7048 050222  000010                       .WORD   10
7049 050224  011120                       .WORD   MSG2
7050 050226  000000                       .WORD   0
7051 050230  112764  000007  034446       MOVB    #7,WHY.DROPT(R4)      ; *,*(LUN)                    4479
7052 050236  010400                       MOV     R4,R0                 ; LUN,*                       4480
7053 050240  104451                       TRAP    51
7054 050242  062706  000020        15$:   ADD     #20,SP                ;                             4481
7055 050246  000167  001146        16$:   JMP     37$
7056 050252  020527  000005        17$:   CMP     R5,#5                 ; VALUE,*                     4423
7057 050256  001157                       BNE     24$
7058 050260  004767  167046               JSR     PC,ISOLATE            ;                             4486
7059 050264  032767  000001  131766       BIT     #1,ERROUT             ;                             4488
7060 050272  001423                       BEQ     18$
7061 050274  017700  164110               MOV     @#L.REG+42,R0
7062 050300  006200                       ASR     R0
7063 050302  006200                       ASR     R0
7064 050304  006200                       ASR     R0
7065 050306  006200                       ASR     R0
```

```
                                        ;MLX4
                                        ;                COMMAND INTEGRITY ROUTINE                    27-Mar-1982 19:24:42    TOPS
                                                                                                     27-Mar-1982 19:23:44    PA:<
 7070 050310  006200                            ASR     R0
 7071 050312  006200                            ASR     R0
 7072 050314  042700  177700                    BIC     #177700,R0
 7073 050320  010046                            MOV     R0,-(SP)
 7074 050322  012746  006640                    MOV     #FMT10B,-(SP)
 7075 050326  012746  000002                    MOV     #2,-(SP)
 7076 050332  010600                            MOV     SP,R0                   ; SP,*
 7077 050334  104414                            TRAP    14
 7078 050336  062706  000006                    ADD     #6,SP
 7079 050342  017766  164044  000036  18$:      MOV     @ML.REG+44,36(SP)       ; *,OLDSEC                   4491
 7080 050350  017700  164034                    MOV     @ML.REG+42,R0           ;                            4492
 7081 050354  006200                            ASR     R0
 7082 050356  006200                            ASR     R0
 7083 050360  006200                            ASR     R0
 7084 050362  006200                            ASR     R0
 7085 050364  006200                            ASR     R0
 7086 050366  006200                            ASR     R0
 7087 050370  042700  177700                    BIC     #177700,R0
 7088 050374  010066  000034                    MOV     R0,34(SP)               ; *,OLDCHN
 7089 050400  012746  000001                    MOV     #1,-( P)                ;                            4494
 7090 050404  012746  045612                    MOV     #READ,-(SP)
 7091 050410  010446                            MOV     R4,-(SP)                ; LUN,*
 7092 050412  010346                            MOV     R3,-(SP)                ; WRDCNT,*
 7093 050414  012746  022670                    MOV     #RBUFF,-(SP)
 7094 050420  011246                            MOV     (R2),-(SP)
 7095 050422  004767  175576                    JSR     PC,RETRY
 7096 050426  062706  000014                    ADD     #14,SP
 7097 050432  020027  000005                    CMP     R0,#5
 7098 050436  001051                            BNE     21$
 7099 050440  027766  163746  000036            CMP     @ML.REG+44,36(SP)       ; *,OLDSEC                   4497
 7100 050446  001034                            BNE     20$
 7101 050450  016646  000034                    MOV     34(SP),-(SP)            ; OLDCHN,*
 7102 050454  017700  163730                    MOV     @ML.REG+42,R0
 7103 050460  006200                            ASR     R0
 7104 050462  006200                            ASR     R0
 7105 050464  006200                            ASR     R0
 7106 050466  006200                            ASR     R0
 7107 050470  006200                            ASR     R0
 7108 050472  006200                            ASR     R0
 7109 050474  042700  177700                    BIC     #177700,R0
 7110 050500  020026                            CMP     R0,(SP)+
 7111 050502  001016                            BNE     20$
 7112 050504  032767  000001  131546            BIT     #1,ERROUT               ;                            4501
 7113 050512  001404                            BEQ     19$
 7114 050514  104456                            TRAP    56
 7115 050516  000011                            .WORD   11
 7116 050520  011202                            .WORD   MSG4
 7117 050522  000000                            .WORD   0
 7118 050524  010446                    19$:     MOV     R4,-(SP)                ; LUN,*                      4504
 7119 050526  016746  163610                    MOV     BOARD,-(SP)
 7120 050532  004767  166664                    JSR     PC,UP.HARD.COUNT
 7121 050536  000426                            BR      23$                     ;                            4497
```

```
                                        ;MLX4
                                        ;               COMMAND INTEGRITY ROUTINE                    27-Mar-1982 19:24:42   TOPS
                                                                                                     27-Mar-1982 19:23:44   PA:<
 7126 050540 032767 000001 131512 20$:  BIT     #1,ERROUT                       ;                                            4509
 7127 050546 001415                     BEQ     22$
 7128 050550 104457                     TRAP    57
 7129 050552 000012                     .WORD   12
 7130 050554 011166                     .WORD   MSG3
 7131 050556 000000                     .WORD   0
 7132 050560 000410                     BR      22$                             ;                                            4512
 7133 050562 032767 000001 131470 21$:  BIT     #1,ERROUT                       ;                                            4518
 7134 050570 001404                     BEQ     22$
 7135 050572 104457                     TRAP    57
 7136 050574 000013                     .WORD   13
 7137 050576 011166                     .WORD   MSG3
 7138 050600 000000                     .WORD   0
 7139 050602 010446                 22$:MOV     R4,-(SP)                        ; LUN,*                                      4521
 7140 050604 016746 163532              MOV     BOARD,-(SP)
 7141 050610 004767 167270              JSR     PC,UP.SOFT.COUNT
 7142 050614 022626                 23$:CMP     (SP)+,(SP)+                     ;                                            4485
 7143 050616 010416                 24$:MOV     R4,(SP)                         ; LUN,*                                      4527
 7144 050620 010346                     MOV     R3,-(SP)                        ; WRDCNT,*
 7145 050622 012746 012670              MOV     #WBUFF,-(SP)
 7146 050626 011246                     MOV     (R2),-(SP)
 7147 050630 004767 175144              JSR     PC,CHECK
 7148 050634 010005                     MOV     R0,R5                           ; *,VALUE
 7149 050636 020527 000001              CMP     R5,#1                           ; VALUE,*                                    4533
 7150 050642 001031                     BNE     25$
 7151 050644 012746 000006              MOV     #6,-(SP)                        ;                                            4539
 7152 050650 012746 046000              MOV     #CHECK,-(SP)
 7153 050654 010446                     MOV     R4,-(SP)                        ; LUN,*
 7154 050656 010346                     MOV     R3,-(SP)                        ; WRDCNT,*
 7155 050660 012746 012670              MOV     #WBUFF,-(SP)
 7156 050664 011246                     MOV     (R2),-(SP)
 7157 050666 004767 175332              JSR     PC,RETRY
 7158 050672 062706 000014              ADD     #14,SP
 7159 050676 005700                     TST     R0
 7160 050700 001462                     BEQ     28$
 7161 050702 112764 000004 034446       MOVB    #4,WHY.DROPT(R4)                ; *,*(LUN)                                   4542
 7162 050710 104455                     TRAP    55                              ;                                            4543
 7163 050712 000014                     .WORD   14
 7164 050714 011070                     .WORD   MSG1
 7165 050716 000000                     .WORD   0
 7166 050720 010400                     MOV     R4,R0                           ; LUN,*                                      4544
 7167 050722 104451                     TRAP    51
 7168 050724 000450                     BR      28$                             ;                                            4545
 7169 050726 020527 000002          25$:CMP     R5,#2                           ; VALUE,*                                    4533
 7170 050732 001012                     BNE     26$
 7171 050734 112764 000005 034446       MOVB    #5,WHY.DROPT(R4)                ; *,*(LUN)                                   4552
 7172 050742 104455                     TRAP    55                              ;                                            4553
 7173 050744 000015                     .WORD   15
 7174 050746 011070                     .WORD   MSG1
 7175 050750 000000                     .WORD   0
 7176 050752 010400                     MOV     R4,R0                           ; LUN,*                                      4554
 7177 050754 104451                     TRAP    51
```

```
                                        ;MLX4
                                        ;                COMMAND INTEGRITY ROUTINE                    27-Mar-1982 19:24:42    TOPS
                                                                                                     27-Mar-1982 19:23:44    PA:<
                                                BR      28$                             ;                                        4555
                                26$:    CMP     R5,#3                           ; VALUE,*                                4533
                                        BNE     27$
                                        TRAP    55                              ;                                        4560
                                        .WORD   16
                                        .WORD   MSG1
                                        .WORD   0
                                        MOVB    #6,WHY.DROPT(R4)                ; *,*(LUN)                               4561
                                        MOV     R4,R0                           ; LUN,*                                  4562
                                        TRAP    51
                                        BR      28$                             ;                                        4563
                                27$:    CMP     R5,#4                           ; VALUE,*                                4533
                                        BNE     29$
                                        JSR     PC,ISOLATE                      ;                                        4568
                                        TRAP    55                              ;                                        4569
                                        .WORD   17
                                        .WORD   MSG2
                                        .WORD   0
                                        MOVB    #7,WHY.DROPT(R4)                ; *,*(LUN)                               4570
                                        MOV     R4,R0                           ; LUN,*                                  4571
                                        TRAP    51
                                28$:    BR      36$                             ;                                        4572
                                29$:    CMP     R5,#5                           ; VALUE,*                                4533
                                        BNE     36$
                                        JSR     PC,ISOLATE                      ;                                        4577
                                        BIT     #1,ERROUT                       ;                                        4579
                                        BEQ     30$
                                        MOV     @ML.REG+42,R0
                                        ASR     R0
                                        ASR     R0
                                        ASR     R0
                                        ASR     R0
                                        ASR     R0
                                        BIC     #177700,R0
                                        MOV     R0,-(SP)
                                        MOV     #FMT10B,-(SP)
                                        MOV     #2,-(SP)
                                        MOV     SP,R0                           ; SP,*
                                        TRAP    14
                                        ADD     #6,SP
                                30$:    MOV     @ML.REG+44,44(SP)               ; *,OLDSEC                               4582
                                        MOV     @ML.REG+42,R0                   ;                                        4583
                                        ASR     R0
                                        ASR     R0
                                        ASR     R0
                                        ASR     R0
                                        ASR     R0
                                        ASR     R0
                                        BIC     #177700,R0
                                        MOV     R0,42(SP)                       ; *,OLDCHN
                                        MOV     #1,-(SP)                        ;                                        4585
```

SEQ 0183

```
                                        ;MLX4
                                        ;                COMMAND INTEGRITY ROUTINE                    27-Mar-1982 19:24:42   TOPS
                                                                                                     27-Mar-1982 19:23:44   PA:<
 7238 051202  012746  046000                        MOV     #CHECK,-(SP)
 7239 051206  010446                                MOV     R4,-(SP)              ; LUN,*
 7240 051210  010346                                MOV     R3,-(SP)              ; WRDCNT,*
 7241 051212  012746  012670                        MOV     #WBUFF,-(SP)
 7242 051216  011246                                MOV     (R2),-(SP)
 7243 051220  004767  175000                        JSR     PC,RETRY
 7244 051224  062706  000014                        ADD     #14,SP
 7245 051230  020027  000005                        CMP     R0,#5
 7246 051234  001051                                BNE     33$
 7247 051236  027766  163150  000044                CMP     @ML.REG+44,44(SP)     ; *,OLDSEC                                  4588
 7248 051244  001034                                BNE     32$
 7249 051246  016600  000042                        MOV     42(SP),R0             ; OLDCHN,*
 7250 051252  017702  163132                        MOV     @ML.REG+42,R2
 7251 051256  006202                                ASR     R2
 7252 051260  006202                                ASR     R2
 7253 051262  006202                                ASR     R2
 7254 051264  006202                                ASR     R2
 7255 051266  006202                                ASR     R2
 7256 051270  006202                                ASR     R2
 7257 051272  042702  177700                        BIC     #177700,R2
 7258 051276  020200                                CMP     R2,R0
 7259 051300  001016                                BNE     32$
 7260 051302  032767  000001  130750                BIT     #1,ERROUT             ;                                           4592
 7261 051310  001404                                BEQ     31$
 7262 051312  104456                                TRAP    56
 7263 051314  000020                                .WORD   20
 7264 051316  011202                                .WORD   MSG4
 7265 051320  000000                                .WORD   0
 7266 051322  010446                    31$:         MOV     R4,-(SP)              ; LUN,*
 7267 051324  016746  163012                        MOV     BOARD,-(SP)                                                         4595
 7268 051330  004767  166066                        JSR     PC,UP.HARD.COUNT
 7269 051334  000426                                BR      35$
 7270 051336  032767  000001  130714    32$:        BIT     #1,ERROUT             ;                                           4588
 7271 051344  001415                                BEQ     34$                   ;                                           4600
 7272 051346  104457                                TRAP    57
 7273 051350  000021                                .WORD   21
 7274 051352  011166                                .WORD   MSG3
 7275 051354  000000                                .WORD   0
 7276 051356  000410                                BR      34$
 7277 051360  032767  000001  130672    33$:        BIT     #1,ERROUT             ;                                           4603
 7278 051366  001404                                BEQ     34$                   ;                                           4609
 7279 051370  104457                                TRAP    57
 7280 051372  000022                                .WORD   22
 7281 051374  011166                                .WORD   MSG3
 7282 051376  000000                                .WORD   0
 7283 051400  010446                    34$:         MOV     R4,-(SP)              ; LUN,*                                     4612
 7284 051402  016746  162734                        MOV     BOARD,-(SP)
 7285 051406  004767  166472                        JSR     PC,UP.SOFT.COUNT
 7286 051412  022626                    35$:        CMP     (SP)+,(SP)+           ;                                           4576
 7287 051414  062706  000026            36$:        ADD     #26,SP                ;                                           4373
 7288 051420  005204                    37$:        INC     R4                    ; LUN                                       4366
 7289 051422  020466  000020            38$:        CMP     R4,20(SP)             ; LUN,*
```

SEQ 0184

```
                                        ;MLX4
                                        ;               COMMAND INTEGRITY ROUTINE                    27-Mar-1982 19:24:42    TOPS
                                                                                                     27-Mar-1982 19:23:44    PA:<
 7294 051426  002002                            BGE     39$
 7295 051430  000167  175656                    JMP     2$
 7296 051434  005726                    39$:    TST     (SP)+                   ;
 7297 051436  005201                            INC     R1                      ; COMP.FLAG                                  4363
 7298 051440  020127  000001                    CMP     R1,#1                   ; COMP.FLAG,*                                4362
 7299 051444  003002                            BGT     40$
 7300 051446  000167  175616                    JMP     1$
 7301 051452  062706  000022            40$:    ADD     #22,SP                  ;                                            4308
 7302 051456  000207                            RTS     PC
 7303
 7304                                   ; Routine Size:  587 words
 7305                                   ; Maximum stack depth per invocation:  33 words
 7310
 7311
```

```
7313 ;MLX4
7314 ;                  DEFINITION OF OPTION 1                          27-Mar-1982 19:24:42    TOPS-20 Bliss-16 V2(212)
7315 ;                                                                  27-Mar-1982 19:23:44    PA:<NEALE>MLX4.BLI.5 (37)
7316 ;          4627  %sbttl 'DEFINITION OF OPTION 1'
7317 ;          4628  routine OPT1 : novalue =
7318 ;          4629      begin                                         !* 1 *  START OF ROUTINE
7319 ;          4630
7320 ;          4631  !++
7321 ;          4632  !  ROUTINE:     OPT1
7322 ;          4633  !
7323 ;          4634  !  PURPOSE:     TO CHECK ADDRESSES USING DATA = SECTOR NUMBER.
7324 ;          4635  !               TRANSFERS ARE 4K WORDS IN LENGTH, AND ALL SECTORS
7325 ;          4636  !               ARE TESTED.
7326 ;          4637  !
7327 ;          4638  !  THE CODE FOR 'OPT1' IN BRIEF:
7328 ;          4639  !
7329 ;          4640  !  BEGIN 1 (START OF ROUTINE)
7330 ;          4641  !  SAY ROUTINE IS RUNNING
7331 ;          4642  !  INCR COMPLEMENT FLAG FROM 0 TO 1
7332 ;          4643  !  : BEGIN 2 (START OF COMPLEMENT FLAG SELECTION LOOP)
7333 ;          4644  !  : INCR LOGICAL UNIT FROM 0 TO LAST
7334 ;          4645  !  : : BEGIN 3 (START OF LOGICAL UNIT SELECTION LOOP)
7335 ;          4646  !  : : TESTLOOP:
7336 ;          4647  !  : : : BEGIN 4 (START OF LOOP THAT COMPLETELY TESTS ONE UNIT)
7337 ;          4648  !  : : : IF UNIT IS ACTIVE
7338 ;          4649  !  : : : THEN
7339 ;          4650  !  : : : : BEGIN 5 (START OF TEST FOR AN ACTIVE UNIT)
7340 ;          4651  !  : : : : INITIALIZE WRITE AND READ BUFFER POINTERS
7341 ;          4652  !  : : : : SECTOR = LOWEST
7342 ;          4653  !  : : : : WHILE SECTOR LEQ HIGHEST DO
7343 ;          4654  !  : : : : : BEGIN 6 (START OF SECTOR SELECTION LOOP)
7344 ;          4655  !  : : : : : GET WRDCNT
7345 ;          4656  !  : : : : : GENERATE THE PATTERN
7346 ;          4657  !  : : : : : SET UP BUFFER POINTERS BEFORE TRANSFER
7347 ;          4658  !  : : : : : WRITE
7348 ;          4659  !  : : : : : LOOK FOR ERRORS (IF DROP UNIT, LEAVE TESTLOOP)
7349 ;          4660  !  : : : : : CHOOSE WHETHER TO WRITE CHECK OR READ
7350 ;          4661  !  : : : : : DO THE WRITE CHECK OR READ
7351 ;          4662  !  : : : : : LOOK FOR ERRORS (IF DROP UNIT, LEAVE TESTLOOP)
7352 ;          4663  !  : : : : : CHANGE BUFFER POINTERS AFTER TRANSFER
7353 ;          4664  !  : : : : : UPDATE SECTOR NUMBER BY NUMBER OF SECTORS IN PREVIOUS TRANSFER
7354 ;          4665  !  : : : : : END 6 (END OF SECTOR SELECTION LOOP)
7355 ;          4666  !  : : : : END 5 (END OF TEST FOR AN ACTIVE UNIT)
7356 ;          4667  !  : : : END 4 (END OF TESTLOOP)
7357 ;          4668  !  : : END 3 (END OF LOGICAL UNIT SELECTION LOOP)
7358 ;          4669  !  : END 2 (END OF COMPLEMENT FLAG SELECTION LOOP)
7359 ;          4670  !  RETURN
7360 ;          4671  !  END 1 (END OF ROUTINE)
7361 ;          4672  !--
7362 ;          4673
7363 ;          4674      label
7364 ;          4675          LOOP;
7365 ;          4676
7366 ;          4677      local
7367 ;          4678          VALUE,
```

```
7369 ;MLX4
7370 ;                  DEFINITION OF OPTION 1
7371
7372 ;          4679           WRDCNT,
7373 ;          4680           COMMAND,
7374 ;          4681           PTR,
7375 ;          4682           OLDSEC,
7376 ;          4683           OLDCHN,
7377 ;          4684           SECTOR,
7378 ;          4685           DBL_VALUE;
7379 ;          4686
7380 ;          4687       PRINTB (SAY2, WRD34, RTN1);
7381 ;          4688       !'RUNNING OPT1'
7382 ;          4689
7383 ;          4690       incr COMP_FLAG from 0 to 1 do
7384 ;          4691           begin                                  !* 2 *  START OF COMPLEMENT FLAG SELECTION LOOP
7385 ;          4692
7386 ;          4693           incr LUN from 0 to (.L$UNIT - 1) do
7387 ;          4694               begin                              !* 3 *  START OF LOGICAL UNIT SELECTION LOOP
7388 ;          4695   LOOP :
7389 ;          4696               begin                              !* 4 *  START OF THE LOOP THAT COMPLETELY TESTS ONE UNIT
7390 ;          4697
7391 ;          4698               if .DRIVE_STATUS [.LUN] eql ACTIVE
7392 ;          4699               then
7393 ;          4700                   begin                          !* 5 *  START OF TEST FOR AN ACTIVE UNIT
7394 ;          4701                   L$LUN = .LUN;
7395 ;          4702                   WPTR = WBUFF;
7396 ;          4703                   RPTR = RBUFF;
7397 ;          4704                   SECTOR = LOWEST;
7398 ;          4705
7399 ;          4706                   while .SECTOR lequ HIGHEST do
7400 ;          4707                       begin                      !* 6 *  START OF SECTOR SELECTION LOOP
7401 ;          4708                       GEN1 (.SECTOR, .COMP_FLAG);
7402 ;          4709                       WRDCNT = GET_WRDCNT (.SECTOR, HIGHEST);
7403 ;          4710                       SET_PTRS (.WRDCNT);
7404 ;          4711                       VALUE = write (.LUN, .WRDCNT, .WPTR, .SECTOR);
7405 ;          4712
7406 ;          4713                       !+
7407 ;          4714                       !  SEE HOW SUCCESSFUL THE WRITE WAS:
7408 ;          4715                       !-
7409 ;          4716
7410 ;          4717                       selectone .VALUE of        !SEE 'SYSERR' FOR DEFINITION
7411 ;          4718                           set                    !OF ERROR # CONTAINED IN 'VALUE'
7412 ;          4719
7413 ;          4720                           [1] :
7414 ;          4721                               begin              !* 6A * RETRY ALLOWED
7415 ;          4722
7416 ;          4723                               if RETRY (SIX, write, .LUN, .WRDCNT, .WPTR, .SECTOR) neq 0
7417 ;          4724                               then               !THE RETRY FAILED -- SYSTEM FATAL ERROR
7418 ;          4725                                   begin
7419 ;          4726                                   WHY_DROPT [.LUN] = CODE_4;
7420 ;          4727                                   ERRDF (101, MSG1, 0);   !**** OPTION 1 ERROR 01 ****
7421 ;          4728                                   DODU (.LUN);
7422 ;          4729                                   leave LOOP;     !JUMP JUST BEYOND END OF BLOCK * 4 *
7423 ;          4730                                   end;
```

```
;          4731                                    end;                !* 6A *
;          4732
;          4733                                [2] :
;          4734                                    begin               !* 6B * FATAL CONTROLLER ERROR -- NO RETRY ALLOWED
;          4735                                    WHY_DROPT [.LUN] = CODE_5;
;          4736                                    ERRDF (102, MSG1, 0);       !**** OPTION 1 ERROR 02 ****
;          4737                                    DODU (.LUN);
;          4738                                    leave LOOP;         !JUMP JUST BEYOND END OF BLOCK * 4 *
;          4739                                    end;                !* 6B *
;          4740
;          4741                                [3] :
;          4742                                    begin               !* 6C * FATAL DRIVE ERROR -- NO RETRY ALLOWED
;          4743                                    ERRDF (103, MSG1, 0);       !**** OPTION 1 ERROR 03 ****
;          4744                                    WHY_DROPT [.LUN] = CODE_6;
;          4745                                    DODU (.LUN);
;          4746                                    leave LOOP;         !JUMP JUST BEYOND END OF BLOCK * 4 *
;          4747                                    end;                !* 6C *
;          4748                                tes;
;          4749
;          4750                            COMMAND = CHOOSE ();
;          4751
;          4752                            if .COMMAND eql read
;          4753                            then
;          4754                                begin
;          4755                                PTR = .RPTR;
;          4756                                VALUE = read (.LUN, .WRDCNT, .RPTR, .SECTOR);
;          4757                                end
;          4758                            else
;          4759                                begin
;          4760                                PTR = .WPTR;
;          4761                                VALUE = CHECK (.LUN, .WRDCNT, .WPTR, .SECTOR);
;          4762                                end;
;          4763
;          4764                            !+
;          4765                            !  SEE HOW SUCCESSFUL THE OPERATION WAS:
;          4766                            !-
;          4767
;          4768                            selectone .VALUE of         !SEE 'SYSERR' FOR DEFINITION
;          4769                                set                     !OF ERROR # CONTAINED IN 'VALUE'
;          4770
;          4771                                [0] :
;          4772
;          4773                                    if .COMMAND eql read
;          4774                                    then
;          4775                                        begin
;          4776
;          4777                                        if (DBL_VALUE = DOUBLE_CHECK (.WPTR, .RPTR, .WRDCNT)) neq 0
;          4778                                        then
;          4779                                            begin
;          4780                                            SAYWHO (.LUN);
;          4781                                            PRINTB (SAY1, MSG5);        !'ECC LOGIC FAILED TO DETECT DATA ERROR'
;          4782
```

```
;MLX4
;                    DEFINITION OF OPTION 1                          27-Mar-1982 19:24:42    DECPS-20 Bliss-16 V2(212)
;                                                                    27-Mar-1982 19:23.44    PS:<NEALE>MLX4.BLI.5 (37)
;
;          4783                                  PRINTB (FMT12A, ..DBL_VALUE, .DBL_VALUE);
;          4784                                                  !'GOOD DATA: XXXXXX AT LOCATION YYYYYY'
;          4785                                  DBL_VALUE = .DBL_VALUE + BUFSIZ*2;
;          4786                                  PRINTB (FMT12B, ..DBL_VALUE, .DBL_VALUE);
;          4787                                                  !'BAD DATA:  PPPPPP AT LOCATION QQQQQQ'
;          4788                                  WHY_DROPT [.LUN] = CODE_8;
;          4789                                  ERRDF (104, MSG1, 0);       !**** OPTION 1 ERROR 04 ****
;          4790                                  DODU (.LUN);
;          4791                                  leave LOOP; !JUMP JUST BEYOND END OF BLOCK * 4 *
;          4792                                  end;
;          4793
;          4794                              end;
;          4795
;          4796                          [1] :
;          4797                              begin                   !* 6D * RETRY ALLOWED
;          4798
;          4799                              if RETRY (SIX, .COMMAND, .LUN, .WRDCNT, .PTR, .SECTOR) neq 0
;          4800                              then                    !THE RETRY FAILED -- SYSTEM FATAL ERROR
;          4801                                  begin
;          4802                                  WHY_DROPT [.LUN] = CODE_4;
;          4803                                  ERRDF (105, MSG1, 0);   !**** OPTION 1 ERROR 05 ****
;          4804                                  DODU (.LUN);
;          4805                                  leave LOOP;     !JUMP JUST BEYOND END OF BLOCK * 4 *
;          4806                                  end;
;          4807
;          4808                              end;                    !* 6D *
;          4809
;          4810                          [2] :
;          4811                              begin                   !* 6E * FATAL CONTROLLER ERROR -- NO RETRY ALLOWED
;          4812                              WHY_DROPT [.LUN] = CODE_5;
;          4813                              ERRDF (106, MSG1, 0);       !**** OPTION 1 ERROR 06 ****
;          4814                              DODU (.LUN);
;          4815                              leave LOOP;             !JUMP JUST BEYOND END OF BLOCK * 4 *
;          4816                              end;                    !* 6E *
;          4817
;          4818                          [3] :
;          4819                              begin                   !* 6F * FATAL DRIVE ERROR -- NO RETRY ALLOWED
;          4820                              WHY_DROPT [.LUN] = CODE_6;
;          4821                              ERRDF (107, MSG1, 0);       !**** OPTION 1 ERROR 07 ****
;          4822                              DODU (.LUN);
;          4823                              leave LOOP;             !JUMP JUST BEYOND END OF BLOCK * 4 *
;          4824                              end;                    !* 6F *
;          4825
;          4826                          [4] :
;          4827                              begin                   !* 6G * UNRECOVERABLE DATA ERROR
;          4828                              ISOLATE ();
;          4829                              ERRDF (108, MSG2, 0);       !**** OPTION 1 ERROR 08 ****
;          4830                              WHY_DROPT [.LUN] = CODE_7;
;          4831                              DODU (.LUN);
;          4832                              leave LOOP;             !JUMP JUST BEYOND END OF BLOCK * 4 *
;          4833                              end;                    !* 6G *
;          4834
```

```
;MLX4
;                DEFINITION OF OPTION 1                          27-Mar-1982 19:24:42    TOPS-20 Bliss-16 V2(212)
;                                                                27-Mar-1982 19:23:44    PA:<NEALE>MLX4.BLI.5 (37)
;
;       4835                                    [5] :
;       4836                                        begin                   !* 6H * RECOVERABLE DATA ERROR
;       4837                                        ISOLATE ();
;       4838
;       4839                                        if .ERROUT then PRINTB (FMT10B, .CHAN);
;       4840
;       4841                                        !'    BIT 00'
;       4842                                        OLDSEC = .MLEL;
;       4843                                        OLDCHN = .CHAN;
;       4844
;       4845                                        if RETRY (ONE, .COMMAND, .LUN, .WRDCNT, .PTR, .SECTOR) eql 5
;       4846                                        then
;       4847
;       4848                                            if ((.MLEL eql .OLDSEC) and (.CHAN eql .OLDCHN))
;       4849                                            then
;       4850                                                begin
;       4851
;       4852                                                if .ERROUT then ERRHRD (109, MSG4, 0);      !**** OPTION 1 ERROR 09 ****
;       4853
;       4854                                                UP_HARD_COUNT (.LUN, .BOARD);
;       4855                                                end
;       4856                                            else
;       4857                                                begin
;       4858
;       4859                                                if .ERROUT then ERRSOFT (110, MSG3, 0);     !**** OPTION 1 ERROR 10 ****
;       4860
;       4861                                                UP_SOFT_COUNT (.LUN, .BOARD);
;       4862                                                end
;       4863
;       4864                                        else
;       4865                                            begin
;       4866
;       4867                                            if .ERROUT then ERRSOFT (111, MSG3, 0); !**** OPTION 1 ERROR 11 ****
;       4868
;       4869                                            UP_SOFT_COUNT (.LUN, .BOARD);
;       4870                                            end;
;       4871
;       4872                                        end;                    !* 6H *
;       4873                                    tes;
;       4874
;       4875                                WPTR = .WPTR + (.WRDCNT*2);
;       4876                                SECTOR = .SECTOR + (.WRDCNT/256);
;       4877                                end;                            !* 6 *  END OF SECTOR SELECTION LOOP
;       4878
;       4879                            end;                                !* 5 *  END OF TEST FOR AN ACTIVE UNIT
;       4880
;       4881                    !+
;       4882                    ! Test to see if this uut's address space is
;       4883                    ! to be read for soft errors.  This test is
;       4884                    ! is intended for DMT purposes.
;       4885                    !-
;       4886
```

```
;MLX4
;                    DEFINITION OF OPTION 1                          27-Mar-1982 19:24:42    TOPS-20 Bliss-16 V2(212)
;                                                                    27-Mar-1982 19:23:44    PA:<NEALE>MLX4.BLI.5 (37)

;          4887              if .EFN$21                              !Is the background pattern to be read
;          4888              then
;          4889                  begin
;          4890
;          4891          !
;          4892          ! version czmlbb changed incr to incru
;          4893          !
;          4894                  incru SECTOR from LOWEST to HIGHEST do
;          4895
;          4896                      if read (.LUN, 256, RBUFF, .SECTOR) eql 5
;          4897                      then
;          4898                          begin
;          4899                          ISOLATE ();             !Find the failing bank and board no.
;          4900
;          4901                          if .ERROUT then PRINTB (FMT10B, .CHAN);
;          4902
;          4903                          ! Print where the error is
;          4904                          !
;          4905                          ! Save the contents of the ML error location
;          4906                          ! register so we can compare it to the new
;          4907                          ! contents of this register after the retry.
;          4908                          ! This is done to classify the error.
;          4909                          !
;          4910                          OLDSEC = .MLEL;
;          4911                          OLDCHN = .CHAN;
;          4912                          !
;          4913                          ! Do a classify retry call.  If the same error
;          4914                          ! occures then classify it as a hard error.  If
;          4915                          ! a different error occured or the error went away
;          4916                          ! then classify it as a soft error.
;          4917                          !
;          4918
;          4919                          if RETRY (ONE, .COMMAND, .LUN, 256, .PTR, .OLDSEC) eql 5
;          4920                          then
;          4921                          !
;          4922                          ! The same error occured so see if it is at the same
;          4923                          ! sector and channel number, if so then classify
;          4924                          ! it as a hard error else classiy it as a soft
;          4925                          ! error.
;          4926                          !
;          4927
;          4928                              if ((.MLEL eql .OLDSEC) and (.CHAN eql .OLDCHN))
;          4929                              then
;          4930                                  begin           !Same error occured 'hard'
;          4931
;          4932                                  if .ERROUT      !Print error if enabled
;          4933                                  then
;          4934                                      begin
;        P 4935                                      ERRHRD (112,        !Error number
;        P 4936                                          MSG4,   !Error message
;          4937                                          0);     !Additional message routine
;          4938                                      end;
```

```
;MLX4
;                    DEFINITION OF OPTION 1                                  27-Mar-1982 19:24:42    TOPS-20 Bliss-16 V2(212)
;                                                                            27-Mar-1982 19:23:44    PA:<NEALE>MLX4.BLI.5 (37)

;          4939
;          4940                                        UP_HARD_COUNT (.LUN, .BOARD);
;          4941                                        end
;          4942                                    else
;          4943                                        begin               !Not the same error 'soft'
;          4944
;          4945                                        if .ERROUT          !Print error if enabled
;          4946                                        then
;          4947                                            begin
;        P 4948                                            ERRSOFT (113,       !Error number
;        P 4949                                                MSG3,   !Error message
;          4950                                                0);     !Additional message routine
;          4951                                            end;
;          4952
;          4953                                        UP_SOFT_COUNT (.LUN, .BOARD);
;          4954                                        end
;          4955
;          4956                                else                            !Not the same error 'soft'
;          4957                                    begin
;          4958
;          4959                                        if .ERROUT          !Print error if enabled
;          4960                                        then
;          4961                                            begin
;        P 4962                                            ERRSOFT (114,   !Error number
;        P 4963                                                MSG3,       !Error message
;          4964                                                0);         !Additional message routine
;          4965                                            end;
;          4966
;          4967                                        UP_SOFT_COUNT (.LUN, .BOARD);
;          4968                                        end;
;          4969
;          4970                                end;
;          4971
;          4972                        end;
;          4973
;          4974                    end;                                    !* 4 *  END OF LOOP THAT COMPLETELY TESTS ONE UNIT
;          4975                    end;                                    !* 3 *  END OF LOGICAL UNIT SELECTION LOOP
;          4976
;          4977                end;                                        !* 2 *  END OF COMPLEMENT FLAG SELECTION LOOP
;          4978
;          4979            return;
;          4980            end;                                            !* 1 *  END OF ROUTINE
;
;
;                                    .SBTTL  OPT1 DEFINITION OF OPTION 1
;          OPT1:                     JSR     R1,$SAVE5                     ;                                                  4628
;                                    SUB     #20,SP
```

SEQ 0192

```
                                        ;MLX4
                                        ;                DEFINITION OF OPTION 1                                  27-Mar-1982 19:24:42    TOPS
                                                                                                                 27-Mar-1982 19:23:44    PA:<
 7705
 7706
 7707
 7708 051470  012746  007626                    MOV     #RTN1,-(SP)             ;                                                4687
 7709 051474  012746  007344                    MOV     #WRD34,-(SP)
 7710 051500  012746  007100                    MOV     #SAY2,-(SP)
 7711 051504  012746  000003                    MOV     #3,-(SP)
 7712 051510  010600                            MOV     SP,R0                   ; SP,*
 7713 051512  104414                            TRAP    14
 7714 051514  005066  000026                    CLR     26(SP)                  ; COMP.FLAG                              4690
 7715 051520  016766  130266  000012  1$:      MOV     L$UNIT,12(SP)           ;                                        4693
 7716 051526  005066  000010                    CLR     10(SP)                  ; LUN
 7717 051532  000167  002254                    JMP     41$
 7718 051536  016600  000010            2$:     MOV     10(SP),R0               ; LUN,*                                  4698
 7719 051542  006200                            ASR     R0
 7720 051544  006200                            ASR     R0
 7721 051546  006200                            ASR     R0
 7722 051550  062700  034442                    ADD     #DRIVE.STATUS,R0
 7723 051554  010046                            MOV     R0,-(SP)
 7724 051556  016646  000012                    MOV     12(SP),-(SP)            ; LUN,*
 7725 051562  042716  177770                    BIC     #177770,(SP)
 7726 051566  012746  000001                    MOV     #1,-(SP)
 7727 051572  005046                            CLR     -(SP)
 7728 051574  004767  132556                    JSR     PC,BL$GT2
 7729 051600  062706  000010                    ADD     #10,SP
 7730 051604  005300                            DEC     R0
 7731 051606  001402                            BEQ     4$
 7732 051610  000167  001516            3$:     JMP     29$
 7733 051614  016667  000010  130252  4$:      MOV     10(SP),L$LUN            ; LUN,*                                  4701
 7734 051622  012767  012670  161040            MOV     #WBUFF,WPTR             ;                                        4702
 7735 051630  012767  022670  161034            MOV     #RBUFF,RPTR             ;                                        4703
 7736 051636  016605  000010                    MOV     10(SP),R5               ; LUN,*                                  4704
 7737 051642  006305                            ASL     R5
 7738 051644  016504  034460                    MOV     LOW.SECT(R5),R4         ; *,SECTOR
 7739 051650  020465  034500            5$:     CMP     R4,TOP.SECT(R5)         ; SECTOR,*                               4706
 7740 051654  101355                            BHI     3$
 7741 051656  010446                            MOV     R4,-(SP)                ; SECTOR,*                               4708
 7742 051660  016646  000030                    MOV     30(SP),-(SP)            ; COMP.FLAG,*
 7743 051664  004767  167042                    JSR     PC,GEN1
 7744 051670  010416                            MOV     R4,(SP)                 ; SECTOR,*                               4709
 7745 051672  016546  034500                    MOV     TOP.SECT(R5),-(SP)
 7746 051676  004767  175274                    JSR     PC,GET.WRDCNT
 7747 051702  010002                            MOV     R0,R2                   ; *,WRDCNT
 7748 051706  010216                            MOV     R2,(SP)                 ; WRDCNT,*                               4710
 7749 051706  004767  175220                    JSR     PC,SET.PTRS
 7750 051712  016616  000016                    MOV     16(SP),(SP)             ; LUN,*                                  4711
 7751 051716  010246                            MOV     R2,-(SP)                ; WRDCNT,*
 7752 051720  016746  160744                    MOV     WPTR,-(SP)
 7753 051724  010446                            MOV     R4,-(SP)                ; SECTOR,*
 7754 051726  004767  173472                    JSR     PC,WRITE
 7755 051732  010003                            MOV     R0,R3                   ; *,VALUE
 7756 051734  020327  000001                    CMP     R3,#1                   ; VALUE,*                                4717
 7757 051740  001035                            BNE     6$
 7758 051742  012746  000006                    MOV     #6,-(SP)                ;                                        4723
 7759 051746  012746  045424                    MOV     #WRITE,-(SP)
```

SEQ 0193

```
                                        ;MLX4
                                        ;
                                        ;               DEFINITION OF OPTION 1                                  27-Mar-1982 19:24:42    TOPS
                                        ;                                                                       27-Mar-1982 19:23:44    PA:<
7764 051752  016646  000030                    MOV     30(SP),-(SP)            ; LUN,*
7765 051756  010246                            MOV     R2,-(SP)                ; WRDCNT,*
7766 051760  016746  160704                    MOV     WPTR,-(SP)
7767 051764  010446                            MOV     R4,-(SP)                ; SECTOR,*
7768 051766  004767  174232                    JSR     PC,RETRY
7769 051772  062706  000014                    ADD     #14,SP
7770 051776  005700                            TST     R0
7771 052000  001456                            BEQ     9$
7772 052002  016601  000024                    MOV     24(SP),R1
7773 052006  112761  000004  034446            MOVB    #4,WHY.DROPT(R1)        ; LUN,*                         4726
7774 052014  104455                            TRAP    55                      ;                               4727
7775 052016  000145                            .WORD   145
7776 052020  011070                            .WORD   MSG1
7777 052022  000000                            .WORD   0
7778 052024  016600  000024                    MOV     24(SP),R0               ; LUN,*                         4728
7779 052030  104451                            TRAP    51
7780 052032  000436                            BR      8$                      ;                               4729
7781 052034  020327  000002            6$:     CMP     R3,#2                   ; VALUE,*                       4717
7782 052040  001015                            BNE     7$
7783 052042  016601  000024                    MOV     24(SP),R1
7784 052046  112761  000005  034446            MOVB    #5,WHY.DROPT(R1)        ; LUN,*                         4736
7785 052054  104455                            TRAP    55                      ;                               4737
7786 052056  000146                            .WORD   146
7787 052060  011070                            .WORD   MSG1
7788 052062  000000                            .WORD   0
7789 052064  016600  000024                    MOV     24(SP),R0               ; LUN,*                         4738
7790 052070  104451                            TRAP    51
7791 052072  000416                            BR      8$                      ;                               4739
7792 052074  020327  000003            7$:     CMP     R3,#3                   ; VALUE,*                       4717
7793 052100  001016                            BNE     9$
7794 052102  104455                            TRAP    55                      ;                               4744
7795 052104  000147                            .WORD   147
7796 052106  011070                            .WORD   MSG1
7797 052110  000000                            .WORD   0
7798 052112  016601  000024                    MOV     24(SP),R1
7799 052116  112761  000006  034446            MOVB    #6,WHY.DROPT(R1)        ; LUN,*                         4745
7800 052124  010100                            MOV     R1,R0                   ; LUN,*                         4746
7801 052126  104451                            TRAP    51
7802 052130  062706  000014            8$:     ADD     #14,SP                  ;                               4747
7803 052134  000543                            BR      14$
7804 052136  004767  174024            9$:     JSR     PC,CHOOSE               ;                               4751
7805 052142  010066  000034                    MOV     R0,34(SP)               ; *,COMMAND
7806 052146  005001                            CLR     R1                      ;                               4753
7807 052150  020027  045612                    CMP     R0,#READ                ; COMMAND,*
7808 052154  001015                            BNE     10$
7809 052156  005201                            INC     R1
7810 052160  016766  160506  000036            MOV     RPTR,36(SP)             ; *,PTR                         4756
7811 052166  016646  000024                    MOV     24(SP),-(SP)            ; LUN,*                         4757
7812 052172  010246                            MOV     R2,-(SP)                ; WRDCNT,*
7813 052174  016746  160472                    MOV     RPTR,-(SP)
7814 052200  010446                            MOV     R4,-(SP)                ; SECTOR,*
7815 052202  004767  173404                    JSR     PC,READ
```

```
                                        ;MLX4
                                        ;               DEFINITION OF OPTION 1                                  27-Mar-1982 19:24:42    TOPS
                                                                                                                27-Mar-1982 19:23:44    PA:<
 7820 052206  000413                            BR      11$
 7821 052210  016766  160454  000036  10$:     MOV     WPTR,36(SP)             ; *,PTR                                         4761
 7822 052216  016646  000024                    MOV     24(SP),-(SP)            ; LUN,*                                         4762
 7823 052222  010246                            MOV     R2,-(SP)                ; WRDCNT,*
 7824 052224  016746  160440                    MOV     WPTR,-(SP)
 7825 052230  010446                            MOV     R4,-(SP)                ; SECTOR,*
 7826 052232  004767  173542                    JSR     PC,CHECK
 7827 052236  010003                    11$:    MOV     R0,R3                   ; *,VALUE
 7828 052240  001102                            BNE     15$                     ;                                               4769
 7829 052242  006001                            ROR     R1                      ;                                               4774
 7830 052244  103402                            BLO     13$
 7831 052246  000167  001016            12$:    JMP     28$
 7832 052252  016746  160412            13$:    MOV     WPTR,-(SP)              ;                                               4778
 7833 052256  016746  160410                    MOV     RPTR,-(SP)
 7834 052262  010246                            MOV     R2,-(SP)                ; WRDCNT,*
 7835 052264  004767  172036                    JSR     PC,DOUBLE.CHECK
 7836 052270  062706  000006                    ADD     #6,SP
 7837 052274  010066  000042                    MOV     R0,42(SP)               ; *,DBL.VALUE
 7838 052300  001762                            BEQ     12$
 7839 052302  016646  000034                    MOV     34(SP),-(SP)            ; LUN,*                                         4781
 7840 052306  004767  163212                    JSR     PC,SAYWHO
 7841 052312  012716  011216                    MOV     #MSG5,(SP)              ;                                               4782
 7842 052316  012746  007072                    MOV     #SAY1,-(SP)
 7843 052322  012746  000002                    MOV     #2,-(SP)
 7844 052326  010600                            MOV     SP,R0                   ; SP,*
 7845 052330  104414                            TRAP    14
 7846 052332  016616  000050                    MOV     50(SP),(SP)             ; DBL.VALUE,*                                   4783
 7847 052336  017646  000050                    MOV     @50(SP),-(SP)           ; DBL.VALUE,*
 7848 052342  012746  006710                    MOV     #FMT12A,-(SP)
 7849 052346  012746  000003                    MOV     #3,-(SP)
 7850 052352  010600                            MOV     SP,R0                   ; SP,*
 7851 052354  104414                            TRAP    14
 7852 052356  062766  010000  000056            ADD     #10000,56(SP)           ; *,DBL.VALUE                                   4785
 7853 052364  016616  000056                    MOV     56(SP),(SP)             ; DBL.VALUE,*                                   4786
 7854 052370  017646  000056                    MOV     @56(SP),-(SP)           ; DBL.VALUE,*
 7855 052374  012746  006756                    MOV     #FMT12B,-(SP)
 7856 052400  012746  000003                    MOV     #3,-(SP)
 7857 052404  010600                            MOV     SP,R0                   ; SP,*
 7858 052406  104414                            TRAP    14
 7859 052410  016601  000056                    MOV     56(SP),R1               ; LUN,*                                         4788
 7860 052414  112761  000010  034446            MOVB    #10,WHY.DROPT(R1)
 7861 052422  104455                            TRAP    55                      ;                                               4789
 7862 052424  000150                            .WORD   150
 7863 052426  011070                            .WORD   MSG1
 7864 052430  000000                            .WORD   0
 7865 052432  016600  000056                    MOV     56(SP),R0               ; LUN,*                                         4790
 7866 052436  104451                            TRAP    51
 7867 052440  062706  000046                    ADD     #46,SP                  ;                                               4791
 7868 052444  000522                    14$:    BR      20$
 7869 052446  020327  000001            15$:    CMP     R3,#1                   ; VALUE,*                                       4769
 7870 052452  001035                            BNE     16$
 7871 052454  012746  000006                    MOV     #6,-(SP)                ;                                               4799
```

```
                                        ;MLX4
                                        ;               DEFINITION OF OPTION 1                                  27-Mar-1982 19:24:42    TOPS
                                                                                                                27-Mar-1982 19:23:44    PA:<
052466  016646  000046                        MOV     46(SP),-(SP)            ; COMMAND,*
052464  016646  000040                        MOV     40(SP),-(SP)            ; LUN,*
052470  010246                                MOV     R2,-(SP)                ; WRDCNT,*
052472  016646  000056                        MOV     56(SP),-(SP)            ; PTR,*
052476  010446                                MOV     R4,-(SP)                ; SECTOR,*
052500  004767  173520                        JSR     PC,RETRY
052504  062706  000014                        ADD     #14,SP
052510  005700                                TST     R0
052512  001655                                BEQ     12$
052514  016601  000034                        MOV     34(SP),R1               ; LUN,*                                 4802
052520  112761  000004  034446                MOVB    #4,WHY.DROPT(R1)
052526  104455                                TRAP    55                      ;                                       4803
052530  000151                                .WORD   151
052532  011070                                .WORD   MSG1
052534  000000                                .WORD   0
052536  016600  000034                        MOV     34(SP),R0               ; LUN,*                                 4804
052542  104451                                TRAP    51
052544  000460                                BR      19$                     ;                                       4805
052546  020327  000002                16$:    CMP     R3,#2                   ; VALUE,*                               4769
052552  001015                                BNE     17$
052554  016601  000034                        MOV     34(SP),R1               ; LUN,*                                 4812
052560  112761  000005  034446                MOVB    #5,WHY.DROPT(R1)
052566  104455                                TRAP    55                      ;                                       4813
052570  000152                                .WORD   152
052572  011070                                .WORD   MSG1
052574  000000                                .WORD   0
052576  016600  000034                        MOV     34(SP),R0               ; LUN,*                                 4814
052602  104451                                TRAP    51
052604  000440                                BR      19$                     ;                                       4815
052606  020327  000003                17$:    CMP     R3,#3                   ; VALUE,*                               4769
052612  001015                                BNE     18$
052614  016601  000034                        MOV     34(SP),R1               ; LUN,*                                 4820
052620  112761  000006  034446                MOVB    #6,WHY.DROPT(R1)
052626  104455                                TRAP    55                      ;                                       4821
052630  000153                                .WORD   153
052632  011070                                .WORD   MSG1
052634  000000                                .WORD   0
052636  016600  000034                        MOV     34(SP),R0               ; LUN,*                                 4822
052642  104451                                TRAP    51
052644  000420                                BR      19$                     ;                                       4823
052646  020327  000004                18$:    CMP     R3,#4                   ; VALUE,*                               4769
052652  001021                                BNE     21$
052654  004767  164452                        JSR     PC,ISOLATE              ;                                       4828
052660  104455                                TRAP    55                      ;                                       4829
052662  000154                                .WORD   154
052664  011120                                .WORD   MSG2
052666  000000                                .WORD   0
052670  016601  000034                        MOV     34(SP),R1               ; LUN,*                                 4830
052674  112761  000007  034446                MOVB    #7,WHY.DROPT(R1)
052702  010100                                MOV     R1,R0                   ; LUN,*                                 4831
052704  104451                                TRAP    51
052706  062706  000024                19$:    ADD     #24,SP                  ;                                       4832
```

```
                                        :MLX4
                                        :                       DEFINITION OF OPTION 1                                  27-Mar-1982 19:24:42    TOPS
                                                                                                                        27-Mar-1982 19:23:44    PA:<
7932 052712  000167  001070            20$:    JMP     40$
7933 052716  020327  000005            21$:    CMP     R3,#5                   ; VALUE,*                                       4769
7934 052722  001162                            BNE     28$
7935 052724  004767  164402                    JSR     PC,ISOLATE              ;                                       4837
7936 052730  032767  000001  127322            BIT     #1,ERROUT               ;                                       4839
7937 052736  001423                            BEQ     22$
7938 052740  017701  161444                    MOV     @ML.REG+42,R1
7939 052744  006201                            ASR     R1
7940 052746  006201                            ASR     R1
7941 052750  006201                            ASR     R1
7942 052752  006201                            ASR     R1
7943 052754  006201                            ASR     R1
7944 052756  006201                            ASR     R1
7945 052760  042701  177700                    BIC     #177700,R1
7946 052764  010146                            MOV     R1,-(SP)
7947 052766  012746  006640                    MOV     #FMT10B,-(SP)
7948 052772  012746  000002                    MOV     #2,-(SP)
7949 052776  010600                            MOV     SP,R0                   ; SP,*
7950 053000  104414                            TRAP    14
7951 053002  062706  000006                    ADD     #6,SP
7952 053006  017766  161400  000050  22$:     MOV     @ML.REG+44,50(SP)       ; *,OLDSEC                              4842
7953 053014  017701  161370                    MOV     @ML.REG+42,R1           ;                                       4843
7954 053020  006201                            ASR     R1
7955 053022  006201                            ASR     R1
7956 053024  006201                            ASR     R1
7957 053026  006201                            ASR     R1
7958 053030  006201                            ASR     R1
7959 053032  006201                            ASR     R1
7960 053034  042701  177700                    BIC     #177700,R1
7961 053040  010166  000040                    MOV     R1,40(SP)               ; *,OLDCHN
7962 053044  012746  000001                    MOV     #1,-(SP)                ;                                       4845
7963 053050  016646  000046                    MOV     46(SP),-(SP)            ; COMMAND,*
7964 053054  016646  000040                    MOV     40(SP),-(SP)            ; LUN,*
7965 053060  010246                            MOV     R2,-(SP)                ; WRDCNT,*
7966 053062  016646  000056                    MOV     56(SP),-(SP)            ; PTR,*
7967 053066  010446                            MOV     R4,-(SP)                ; SECTOR,*
7968 053070  004767  173130                    JSR     PC,RETRY
7969 053074  062706  000014                    ADD     #14,SP
7970 053100  020027  000005                    CMP     R0,#5
7971 053104  001052                            BNE     25$
7972 053106  027766  161300  000050            CMP     @ML.REG+44,50(SP)       ; *,OLDSEC                              4848
7973 053114  001035                            BNE     24$
7974 053116  016600  000040                    MOV     40(SP),R0               ; OLDCHN,*
7975 053122  017701  161262                    MOV     @ML.REG+42,R1
7976 053126  006201                            ASR     R1
7977 053130  006201                            ASR     R1
7978 053132  006201                            ASR     R1
7979 053134  006201                            ASR     R1
7980 053136  006201                            ASR     R1
7981 053140  006201                            ASR     R1
7982 053142  042701  177700                    BIC     #177700,R1
7983 053146  020100                            CMP     R1,R0
```

```
                                        ;MLX4
                                        ;                DEFINITION OF OPTION 1                                  27-Mar-1982 19:24:42   TOPS
                                                                                                                 27-Mar-1982 19:23:44   PA:<
 7988 053150  001017                            BNE     24$
 7989 053152  032767  000001  127100           BIT     #1,ERROUT               ;                                                4852
 7990 053160  001404                            BEQ     23$
 7991 053162  104456                            TRAP    56
 7992 053164  000155                            .WORD   155
 7993 053166  011202                            .WORD   MSG4
 7994 053170  000000                            .WORD   0
 7995 053172  016646  000034           23$:     MOV     34(SP),-(SP)            ; LUN,*                                  4834
 7996 053176  016746  161140                    MOV     BOARD,-(SP)
 7997 053202  004767  164214                    JSR     PC,UP.HARD.COUNT
 7998 053206  000427                            BR      27$
 7999 053210  032767  000001  127042  24$:     BIT     #1,ERROUT               ;                                        4848
 8000 053216  001415                            BEQ     26$                     ;                                        4859
 8001 053220  104457                            TRAP    57
 8002 053222  000156                            .WORD   156
 8003 053224  011166                            .WORD   MSG3
 8004 053226  000000                            .WORD   0
 8005 053230  000410                            BR      26$
 8006 053232  032767  000001  127020  25$:     BIT     #1,ERROUT               ;                                        4861
 8007 053240  001404                            BEQ     26$                     ;                                        4867
 8008 053242  104457                            TRAP    57
 8009 053244  000157                            .WORD   157
 8010 053246  011166                            .WORD   MSG3
 8011 053250  000000                            .WORD   0
 8012 053252  016646  000034           26$:     MOV     34(SP),-(SP)            ; LUN,*                                  4869
 8013 053256  016746  161060                    MOV     BOARD,-(SP)
 8014 053262  004767  164616                    JSR     PC,UP.SOFT.COUNT
 8015 053266  022626                    27$:    CMP     (SP)+,(SP)+             ;                                        4836
 8016 053270  010200                    28$:    MOV     R2,R0                   ; WRDCNT,*                               4875
 8017 053272  006300                            ASL     R0
 8018 053274  066700  157370                    ADD     WPTR,R0
 8019 053300  010067  157364                    MOV     R0,WPTR
 8020 053304  010216                            MOV     R2,(SP)                 ; WRDCNT,*                               4876
 8021 053306  012746  000400                    MOV     #400,-(SP)
 8022 053312  004767  131652                    JSR     PC,BL$DIV
 8023 053316  060400                            ADD     R4,R0                   ; SECTOR,*
 8024 053320  010004                            MOV     R0,R4                   ; *,SECTOR
 8025 053322  062706  000026                    ADD     #26,SP                  ;                                        4707
 8026 053326  000167  176316                    JMP     5$                      ;                                        4706
 8027 053332  032767  000001  126722  29$:     BIT     #1,EFN$21               ;                                        4887
 8028 053340  001002                            BNE     30$
 8029 053342  000167  000440                    JMP     40$
 8030 053346  016600  000010           30$:     MOV     10(SP),R0               ; LUN,*                                  4894
 8031 053352  006300                            ASL     R0
 8032 053354  016001  034500                    MOV     TOP.SECT(R0),R1
 8033 053360  016005  034460                    MOV     LOW.SECT(R0),R5         ; *,SECTOR
 8034 053364  000167  000406                    JMP     39$
 8035 053370  016646  000010           31$:     MOV     10(SP),-(SP)            ; LUN,*                                  4896
 8036 053374  012746  000400                    MOV     #400,-(SP)
 8037 053400  012746  022670                    MOV     #RBUFF,-(SP)
 8038 053404  010546                            MOV     R5,-(SP)                ; SECTOR,*
 8039 053406  004767  172200                    JSR     PC,READ
```

```
                                        ;MLX4
                                        ;
                                        ;       DEFINITION OF OPTION 1                                  27-Mar-1982 19:24:42    TOPS
                                                                                                        27-Mar-1982 19:23:44    PA:<
053412  062706  000010                        ADD     #10,SP
053416  020027  000005                        CMP     R0,#5
053422  001164                              BNE     38$
053424  004767  163702                      JSR     PC,ISOLATE
053430  032767  000001  126622              BIT     #1,ERROUT                       ;                       4899
053436  001423                              BEQ     32$                             ;                       4901
053440  017700  160744                      MOV     @ML.REG+42,R0
053444  006200                              ASR     R0
053446  006200                              ASR     R0
053450  006200                              ASR     R0
053452  006200                              ASR     R0
053454  006200                              ASR     R0
053456  006200                              ASR     R0
053460  042700  177700                      BIC     #177700,R0
053464  010046                              MOV     R0,-(SP)
053466  012746  006640                      MOV     #FMT10B,-(SP)
053472  012746  000002                      MOV     #2,-(SP)
053476  010600                              MOV     SP,R0                           ; SP,*
053500  104414                              TRAP    14
053502  062706  000006                      ADD     #6,SP
053506  017766  160700  000024  32$:        MOV     @ML.REG+44,24(SP)               ; *,OLDSEC
053514  017700  160670                      MOV     @ML.REG+42,R0                   ;                       4910
053520  006200                              ASR     R0                              ;                       4911
053522  006200                              ASR     R0
053524  006200                              ASR     R0
053526  006200                              ASR     R0
053530  006200                              ASR     R0
053532  006200                              ASR     R0
053534  042700  177700                      BIC     #177700,R0
053540  010066  000014                      MOV     R0,14(SP)                       ; *,OLDCHN
053544  012746  000001                      MOV     #1,-(SP)                        ;                       4919
053550  016646  000022                      MOV     22(SP),-(SP)                    ; COMMAND,*
053554  016646  000014                      MOV     14(SP),-(SP)                    ; LUN,*
053560  012746  000400                      MOV     #400,-(SP)
053564  016646  000032                      MOV     32(SP),-(SP)                    ; PTR,*
053570  016646  000036                      MOV     36(SP),-(SP)                    ; OLDSEC,*
053574  004767  172424                      JSR     PC,RETRY
053600  062706  000014                      ADD     #14,SP
053604  020027  000005                      CMP     R0,#5
053610  001052                              BNE     35$
053612  027766  160574  000024              CMP     @ML.REG+44,24(SP)               ; *,OLDSEC              4928
053620  001035                              BNE     34$
053622  016646  000014                      MOV     14(SP),-(SP)                    ; OLDCHN,*
053626  017700  160556                      MOV     @ML.REG+42,R0
053632  006200                              ASR     R0
053634  006200                              ASR     R0
053636  006200                              ASR     R0
053640  006200                              ASR     R0
053642  006200                              ASR     R0
053644  006200                              ASR     R0
053646  042700  177700                      BIC     #177700,R0
053652  020026                              CMP     R0,(SP)+
```

SEQ 0199

```
                                        ;MLX4
                                        ;               DEFINITION OF OPTION 1                  27-Mar-1982 19:24:42    TOPS
                                                                                                27-Mar-1982 19:23:44    PA:<
 8097
 8098
 8099
 8100 053654  001017                            BNE     34$
 8101 053656  032767  000001  126374            BIT     #1,ERROUT               ;                                       4932
 8102 053664  001404                            BEQ     33$
 8103 053666  104456                            TRAP    56                      ;                                       4937
 8104 053670  000160                            .WORD   160
 8105 053672  011202                            .WORD   MSG4
 8106 053674  000000                            .WORD   0
 8107 053676  016646  000010          33$:      MOV     10(SP),-(SP)            ; LUN,*                                 4940
 8108 053702  016746  160434                    MOV     BOARD,-(SP)
 8109 053706  004767  163510                    JSR     PC,UP.HARD.COUNT
 8110 053712  000427                            BR      37$
 8111 053714  032767  000001  126336  34$:      BIT     #1,ERROUT               ;                                       4928
 8112 053722  001415                            BEQ     36$                     ;                                       4945
 8113 053724  104457                            TRAP    57
 8114 053726  000161                            .WORD   161                     ;                                       4950
 8115 053730  011166                            .WORD   MSG3
 8116 053732  000000                            .WORD   0
 8117 053734  000410                            BR      36$
 8118 053736  032767  000001  126314  35$:      BIT     #1,ERROUT               ;                                       4953
 8119 053744  001404                            BEQ     36$                     ;                                       4959
 8120 053746  104457                            TRAP    57
 8121 053750  000162                            .WORD   162                     ;                                       4964
 8122 053752  011166                            .WORD   MSG3
 8123 053754  000000                            .WORD   0
 8124 053756  016646  000010          36$:      MOV     10(SP),-(SP)            ; LUN,*                                 4967
 8125 053762  016746  160354                    MOV     BOARD,-(SP)
 8126 053766  004767  164112                    JSR     PC,UP.SOFT.COUNT
 8127 053772  022626                  37$:      CMP     (SP)+,(SP)+             ;                                       4898
 8128 053774  005205                  38$:      INC     R5                      ; SECTOR                                4894
 8129 053776  020501                  39$:      CMP     R5,R1                   ; SECTOR,*
 8130 054000  101002                            BHI     40$
 8131 054002  000167  177362                    JMP     31$
 8132 054006  005266  000010          40$:      INC     10(SP)                  ; LUN
 8133 054012  026666  000010  000012  41$:      CMP     10(SP),12(SP)           ; LUN,*                                 4693
 8134 054020  002002                            BGE     42$
 8135 054022  000167  175510                    JMP     2$
 8136 054026  005266  000026          42$:      INC     26(SP)                  ; COMP.FLAG                             4690
 8137 054032  026627  000026  000001          CMP     26(SP),#1               ; COMP.FLAG,*
 8138 054040  003002                            BGT     43$
 8139 054042  000167  175452                    JMP     1$
 8140 054046  062706  000030          43$:      ADD     #30,SP                  ;                                       4628
 8141 054052  000207                            RTS     PC
 8142
 8143                                   ; Routine Size:  638 words
 8144                                   ; Maximum stack depth per invocation:  37 words
 8149
 8150
```

```
3152 ;MLX4
8153 ;                DEFINITION OF OPTION 2                    27-Mar-1982 19:24:42    TOPS-20 Bliss-16 V2(212)
8154 ;                                                          27-Mar-1982 19:23:44    PA:<NEALE>MLX4.BLI.5 (38)
8155 ;          4981  %sbttl 'DEFINITION OF OPTION 2'
8156 ;          4982  routine OPT2 : novalue =
8157 ;          4983      begin
8158 ;          4984                                            !* 1 *  START OF ROUTINE
8159 ;          4985  !++
8160 ;          4986  !   ROUTINE:     OPT2
8161 ;          4987  !
8162 ;          4988  !   PURPOSE:     TO CHECK ON DATA RELIABILITY USING THE PATTERNS FROM
8163 ;          4989  !                THE PATTERN TABLE.
8164 ;          4990  !
8165 ;          4991  !   THE CODE FOR 'OPT2' IN BRIEF:
8166 ;          4992  !
8167 ;          4993  !   BEGIN 1 (START OF ROUTINE)
8168 ;          4994  !   SAY THE ROUTINE IS RUNNING
8169 ;          4995  !   CHOOSE A MAXIMUM PATTERN NUMBER
8170 ;          4996  !   INCR COUNT FROM 1 TO (2*MAX)
8171 ;          4997  !   : BEGIN 2 (START OF PATTERN SELECTION LOOP)
8172 ;          4998  !   : GENERATE THE PATTERN
8173 ;          4999  !   : INCR LUN FROM 0 TO LAST
8174 ;          5000  !   : : BEGIN 3 (START OF LOGICAL UNIT SELECTION LOOP)
8175 ;          5001  !   : : TESTLOOP:
8176 ;          5002  !   : : : BEGIN 4 (START OF LOOP THAT COMPLETELY TESTS 1 UNIT)
8177 ;          5003  !   : : : IF UNIT IS ACTIVE
8178 ;          5004  !   : : : THEN
8179 ;          5005  !   : : : : BEGIN 5 (START OF TEST FOR AN ACTIVE UNIT)
8180 ;          5006  !   : : : : INITIALIZE THE WRITE AND READ BUFFER POINTERS
8181 ;          5007  !   : : : : SECTOR = LOWEST
8182 ;          5008  !   : : : : WHILE SECTOR LEQ HIGHEST DO
8183 ;          5009  !   : : : : : BEGIN 6 (START OF SECTOR SELECTION LOOP)
8184 ;          5010  !   : : : : : GET_WRDCNT
8185 ;          5011  !   : : : : : SET_UP BUFFER POINTERS BEFORE TRANSFER
8186 ;          5012  !   : : : : : WRITE
8187 ;          5013  !   : : : : : LOOK FOR ERRORS (IF DROP UNIT, LEAVE TESTLOOP)
8188 ;          5014  !   : : : : : CHOOSE WHETHER TO WRITE CHECK OR READ
8189 ;          5015  !   : : : : : DO THE WRITE CHECK OR READ
8190 ;          5016  !   : : : : : LOOK FOR ERRORS (IF DROP UNIT, LEAVE TESTLOOP)
8191 ;          5017  !   : : : : : CHANGE BUFFER POINTERS AFTER TRANSFER
8192 ;          5018  !   : : : : : UPDATE SECTOR NUMBER BY NUMBER OF SECTORS IN PREVIOUS TRANSFER
8193 ;          5019  !   : : : : : END 6 (END OF SECTOR SELECTION LOOP)
8194 ;          5020  !   : : : : END 5 (END OF TEST FOR AN ACTIVE UNIT)
8195 ;          5021  !   : : : END 4 (END OF TESTLOOP)
8196 ;          5022  !   : : END 3 (END OF LOGICAL UNIT SELECTION LOOP)
8197 ;          5023  !   : IF NOT THE QUICK VERIFY PASS THEN 'LOOP READ' (DESCRIBED BELOW)
8198 ;          5024  !   : END 2 (END OF PATTERN SELECTION LOOP)
8199 ;          5025  !   RETURN
8200 ;          5026  !   END 1 (END OF ROUTINE)
8201 ;          5027  !--
8202 ;          5028
8203 ;          5029      label
8204 ;          5030          LOOP,
8205 ;          5031          LOOP2;
8206 ;          5032
```

```
;MLX4
;                DEFINITION OF OPTION 2                          27-Mar-1982 19:24:42    TOPS-20 Bliss-16 V2(212)
;                                                                27-Mar-1982 19:23:44    PA:<NEALE>MLX4.BLI.5 (38)
;
;          5033      local
;          5034          WRDCNT,
;          5035          VALUE,
;          5036          TEMP,
;          5037          MAXPAT,
;          5038          OLDSEC,
;          5039          OLDCHN,
;          5040          SECTOR,
;          5041          PTR,
;          5042          COMMAND,
;          5043          DBL_VALUE;
;          5044
;          5045      PRINTB (SAY2, WRD34, RTN2);
;          5046      !'RUNNING OPT2'
;          5047      PATTERN = 0;
;          5048
;          5049      if .QUICK neq 0 then MAXPAT = 1 else MAXPAT = NUM_PATS;
;          5050
;          5051      incr COUNT from 1 to (.MAXPAT*2) do
;          5052          begin                                   !* 2 *  START OF PATTERN SELECTION LOOP
;          5053          SELPAT ();
;          5054
;          5055          if .PATTERN gtr 0
;          5056          then
;          5057              PRINTB (FMT1A, PHR9, .PATTERN)
;          5058              !'PATTERN NUMBER   XX'
;          5059          else
;          5060              begin
;          5061              TEMP = -(.PATTERN);
;          5062              PRINTB (FMT1B, PHR9, .TEMP);
;          5063              !'       PATTERN NUMBER - XX'
;          5064              end;
;          5065
;          5066          GEN2 (.PATTERN);
;          5067
;          5068          incr LUN from 0 to (.L$UNIT - 1) do
;          5069              begin                               !* 3 *  START OF LOGICAL UNIT SELECTION LOOP
;          5070  LOOP :
;          5071              begin                               !* 4 *  START OF THE LOOP THAT COMPLETELY TESTS 1 UNIT
;          5072
;          5073              if .DRIVE_STATUS [.LUN] eql ACTIVE
;          5074              then
;          5075                  begin                           !* 5 *  START OF TEST FOR AN ACTIVE UNIT
;          5076                  L$LUN = .LUN;
;          5077                  WPTR = WBUFF;
;          5078                  RPTR = RBUFF;
;          5079                  SECTOR = LOWEST;
;          5080
;          5081                  while .SECTOR lequ HIGHEST do
;          5082                      begin                       !* 6 *  START OF SECTOR SELECTION LOOP
;          5083                      WRDCNT = GET_WRDCNT (.SECTOR, HIGHEST);
;          5084                      SET_PTRS (.WRDCNT);
```

```
;MLX4
;                DEFINITION OF OPTION 2
;
;          5085                  VALUE = write (.LUN, .WRDCNT, .WPTR, .SECTOR);
;          5086
;          5087                  !+
;          5088                  !  SEE HOW SUCCESSFUL THE WRITE WAS:
;          5089                  !-
;          5090
;          5091                  selectone .VALUE of           !SEE 'SYSERR' FOR DEFINITION
;          5092                      set                       !OF ERROR # CONTAINED IN 'VALUE'
;          5093
;          5094                      [1] :
;          5095                          begin                 !* 6A * RETRY ALLOWED
;          5096
;          5097                          if RETRY (SIX, write, .LUN, .WRDCNT, .WPTR, .SECTOR) neq 0
;          5098                          then                  !THE RETRY FAILED -- SYSTEM FATAL ERROR
;          5099                              begin
;          5100                              WHY_DROPT [.LUN] = CODE_4;
;          5101                              ERRDF (201, MSG1, 0);   !**** OPTION 2 ERROR 01 ****
;          5102                              DODU (.LUN);
;          5103                              leave LOOP;       !JUMP JUST BEYOND END OF BLOCK * 4 *
;          5104                              end;
;          5105
;          5106                          end;                  !* 6A *
;          5107
;          5108                      [2] :
;          5109                          begin                 !* 6B * FATAL CONTROLLER ERROR -- NO RETRY ALLOWED
;          5110                          WHY_DROPT [.LUN] = CODE_5;
;          5111                          ERRDF (202, MSG1, 0);       !**** OPTION 2 ERROR 02 ****
;          5112                          DODU (.LUN);
;          5113                          leave LOOP;           !JUMP JUST BEYOND END OF BLOCK * 4 *
;          5114                          end;                  !* 6B *
;          5115
;          5116                      [3] :
;          5117                          begin                 !* 6C * FATAL DRIVE ERROR -- NO RETRY ALLOWED
;          5118                          ERRDF (203, MSG1, 0);       !**** OPTION 2 ERROR 03 ****
;          5119                          WHY_DROPT [.LUN] = CODE_6;
;          5120                          DODO (.LUN);
;          5121                          leave LOOP;           !JUMP JUST BEYOND END OF BLOCK * 4 *
;          5122                          end;                  !* 6C *
;          5123                      tes;
;          5124
;          5125                  COMMAND = CHOOSE ();
;          5126
;          5127                  if .COMMAND eql read
;          5128                  then
;          5129                      begin
;          5130                      PTR = .RPTR;
;          5131                      VALUE = read (.LUN, .WRDCNT, .RPTR, .SECTOR);
;          5132                      end
;          5133                  else
;          5134                      begin
;          5135                      PTR = .WPTR;
;          5136                      VALUE = CHECK (.LUN, .WRDCNT, .WPTR, .SECTOR);
```

```
8320 ;MLX4
8321 ;                DEFINITION OF OPTION 2
8322 ;
8323 ;          5137                        end;
8324 ;          5138
8325 ;          5139                    !+
8326 ;          5140                    !  SEE HOW SUCCESSFUL THE OPERATION WAS:
8327 ;          5141                    !-
8328 ;          5142
8329 ;          5143                    selectone .VALUE of          !SEE 'SYSERR' FOR DEFINITION
8330 ;          5144                        set                      !OF ERROR # CONTAINED IN 'VALUE'
8331 ;          5145
8332 ;          5146                        [0] :
8333 ;          5147
8334 ;          5148                            if .COMMAND eql read
8335 ;          5149                            then
8336 ;          5150                                begin
8337 ;          5151
8338 ;          5152                                if (DBL_VALUE = DOUBLE_CHECK (.WPTR, .RPTR, .WRDCNT)) neq 0
8339 ;          5153                                then
8340 ;          5154                                    begin
8341 ;          5155                                    SAYWHO (.LUN);
8342 ;          5156                                    PRINTB (SAY1, MSG5);        !'ECC LOGIC FAILED TO DETECT DATA ERROR'
8343 ;          5157                                    PRINTB (FMT12A, ..DBL_VALUE, .DBL_VALUE);
8344 ;          5158                                                !'GOOD DATA: XXXXXX AT LOCATION YYYYYY'
8345 ;          5159                                    DBL_VALUE = .DBL_VALUE + BUFSIZ*2;
8346 ;          5160                                    PRINTB (FMT12B, ..DBL_VALUE, .DBL_VALUE);
8347 ;          5161                                                !'BAD DATA:  PPPPPP AT LOCATION QQQQQQ'
8348 ;          5162                                    WHY_DROPT [.LUN] = CODE_8;
8349 ;          5163                                    ERRDF (204, MSG1, 0);       !**** OPTION 2 ERROR 04 ****
8350 ;          5164                                    DODU (.LUN);
8351 ;          5165                                    leave LOOP; !JUMP JUST BEYOND END OF BLOCK * 4 *
8352 ;          5166                                    end;
8353 ;          5167
8354 ;          5168                                end;
8355 ;          5169
8356 ;          5170                        [1] :
8357 ;          5171                            begin                !* 6D * RETRY ALLOWED
8358 ;          5172
8359 ;          5173                            if RETRY (SIX, .COMMAND, .LUN, .WRDCNT, .PTR, .SECTOR) neq 0
8360 ;          5174                            then                 !THE RETRY FAILED -- SYSTEM FATAL ERROR
8361 ;          5175                                begin
8362 ;          5176                                WHY_DROPT [.LUN] = CODE_4;
8363 ;          5177                                ERRDF (205, MSG1, 0);   !**** OPTION 2 ERROR 05 ****
8364 ;          5178                                DODU (.LUN);
8365 ;          5179                                leave LOOP;     !JUMP JUST BEYOND END OF BLOCK * 4 *
8366 ;          5180                                end;
8367 ;          5181
8368 ;          5182                            end;                 !* 6D *
8369 ;          5183
8370 ;          5184                        [2] :
8371 ;          5185                            begin                !* 6E * FATAL CONTROLLER ERROR -- NO RETRY ALLOWED
8372 ;          5186                            WHY_DROPT [.LUN] = CODE_5;
8373 ;          5187                            ERRDF (206, MSG1, 0);       !**** OPTION 2 ERROR 06 ****
8374 ;          5188                            DODU (.LUN);
```

```
8376 ;MLX4
8377 ;                  DEFINITION OF OPTION 2
8378
8379 ;          5189                                    leave LOOP;             !JUMP JUST BEYOND END OF BLOCK * 4 *
8380 ;          5190                                    end;                    !* 6E *
8381 ;          5191
8382 ;          5192                                [3] :
8383 ;          5193                                    begin                   !* 6F * FATAL DRIVE ERROR -- NO RETRY ALLOWED
8384 ;          5194                                    ERRDF (207, MSG1, 0);       !**** OPTION 2 ERROR 07 ****
8385 ;          5195                                    WHY_DROPT [.LUN] = CODE_6;
8386 ;          5196                                    DODO (.LUN);
8387 ;          5197                                    leave LOOP;             !JUMP JUST BEYOND END OF BLOCK * 4 *
8388 ;          5198                                    end;                    !* 6F *
8389 ;          5199
8390 ;          5200                                [4] :
8391 ;          5201                                    begin                   !* 6G * UNRECOVERABLE DATA ERROR
8392 ;          5202                                    ISOLATE ();
8393 ;          5203                                    ERRDF (208, MSG2, 0);       !**** OPTION 2 ERROR 08 ****
8394 ;          5204                                    WHY_DROPT [.LUN] = CODE_7;
8395 ;          5205                                    DODO (.LUN);
8396 ;          5206                                    leave LOOP;             !JUMP JUST BEYOND END OF BLOCK * 4 *
8397 ;          5207                                    end;                    !* 6G *
8398 ;          5208
8399 ;          5209                                [5] :
8400 ;          5210                                    begin                   !* 6H * RECOVERABLE DATA ERROR
8401 ;          5211                                    ISOLATE ();
8402 ;          5212
8403 ;          5213                                    if .ERROUT then PRINTB (FMT10B, .CHAN);
8404 ;          5214
8405 ;          5215                                    !'    BIT QQ'
8406 ;          5216                                    OLDSEC = .MLEL;
8407 ;          5217                                    OLDCHN = .CHAN;
8408 ;          5218
8409 ;          5219                                    if RETRY (ONE, .COMMAND, .LUN, .WRDCNT, .PTR, .SECTOR) eql 5
8410 ;          5220                                    then
8411 ;          5221
8412 ;          5222                                        if ((.MLEL eql .OLDSEC) and (.CHAN eql .OLDCHN))
8413 ;          5223                                        then
8414 ;          5224                                            begin
8415 ;          5225
8416 ;          5226                                            if .ERROUT then ERRHRD (209, MSG4, 0);      !**** OPTION 2 ERROR 09 ****
8417 ;          5227
8418 ;          5228                                            UP_HARD_COUNT (.LUN, .BOARD);
8419 ;          5229                                            end
8420 ;          5230                                        else
8421 ;          5231                                            begin
8422 ;          5232
8423 ;          5233                                            if .ERROUT then ERRSOFT (210, MSG3, 0);     !**** OPTION 2 ERROR 10 ****
8424 ;          5234
8425 ;          5235                                            UP_SOFT_COUNT (.LUN, .BOARD);
8426 ;          5236                                            end
8427 ;          5237
8428 ;          5238                                    else
8429 ;          5239                                        begin
8430 ;          5240
```

SEQ 0205


```
8432 ;MLX4
8433 ;
8434
8435 ;          5241                                    if .ERROUT then ERRSOFT (211, MSG3, 0); !**** OPTION 2 ERROR 11 ****
8436 ;          5242
8437 ;          5243                                    UP_SOFT_COUNT (.LUN, .BOARD);
8438 ;          5244                                    end;
8439 ;          5245
8440 ;          5246                                end;                    !* 6H *
8441 ;          5247                            tes;
8442 ;          5248
8443 ;          5249                            WPTR = .WPTR + (.WRDCNT*2);
8444 ;          5250                            SECTOR = .SECTOR + (.WRDCNT/256);
8445 ;          5251                            end;                        !* 6 *  END OF SECTOR SELECTION LOOP
8446 ;          5252
8447 ;          5253                        end;                            !* 5 *  END OF TEST FOR AN ACTIVE UNIT
8448 ;          5254
8449 ;          5255                    end;                                !* 4 *  END OF TESTLOOP
8450 ;          5256                    end;                                !* 3 *  END OF LOGICAL UNIT SELECTION LOOP
8451 ;          5257
8452 ;          5258                if .QUICK eql 0
8453 ;          5259                then
8454 ;          5260                    begin                               !* 11 *  START OF LOOP READING SECTION
8455 ;          5261
8456 ;          5262                    !++
8457 ;          5263                    !  THIS IS THE 'LOOP READ' SECTION WHICH WAS MENTIONED IN
8458 ;          5264                    !  THE DOCUMENTATION AT THE BEGINNING OF THIS ROUTINE.  IT
8459 ;          5265                    !  IS NOT EXECUTED DURING THE QUICK VERIFY PASS, BUT IT IS
8460 ;          5266                    !  FOR EVERY OTHER PASS THROUGH OPT2.
8461 ;          5267                    !
8462 ;          5268                    !  THE CODE IN BRIEF:
8463 ;          5269                    !
8464 ;          5270                    !  BEGIN 11 (START OF LOOP READING SECTION)
8465 ;          5271                    !  INCR LUN FROM 0 TO LAST
8466 ;          5272                    !  : BEGIN 12 (START OF LOGICAL UNIT SELECTION LOOP)
8467 ;          5273                    !  : TESTLOOP2:
8468 ;          5274                    !  : : BEGIN 13 (START OF LOOP THAT COMPLETELY TESTS 1 UNIT)
8469 ;          5275                    !  : : IF UNIT IS ACTIVE
8470 ;          5276                    !  : : THEN
8471 ;          5277                    !  : : : BEGIN 14 (START OF TEST FOR AN ACTIVE UNIT)
8472 ;          5278                    !  : : : INITIALIZE THE WRITE AND READ BUFFER POINTERS
8473 ;          5279                    !  : : : SECTOR = LOWEST
8474 ;          5280                    !  : : : WHILE SECTOR LEQ HIGHEST DO
8475 ;          5281                    !  : : : : BEGIN 15 (START OF SECTOR SELECTION LOOP)
8476 ;          5282                    !  : : : : GET_WRDCNT
8477 ;          5283                    !  : : : : SET_UP BUFFER POINTERS BEFORE TRANSFER
8478 ;          5284                    !  : : : : CHOOSE WHETHER TO WRITE CHECK OR READ
8479 ;          5285                    !  : : : : INCR KOUNT FROM 1 TO TIMES
8480 ;          5286                    !  : : : : : BEGIN 16 (START OF COUNTING LOOP FOR LOOP READING)
8481 ;          5287                    !  : : : : : DO THE WRITE CHECK OR READ
8482 ;          5288                    !  : : : : : LOOK FOR ERRORS (IF DROP UNIT, LEAVE TESTLOOP2)
8483 ;          5289                    !  : : : : : END 16 (END OF COUNTING LOOP FOR LOOP READING)
8484 ;          5290                    !  : : : : CHANGE BUFFER POINTERS AFTER TRANSFER
8485 ;          5291                    !  : : : : UPDATE SECTOR NUMBER BY # SECTORS IN PREVIOUS TRANSFER
8486 ;          5292                    !  : : : : END 15 (END OF SECTOR SELECTION LOOP)
```

```
;MLX4
;                DEFINITION OF OPTION 2                          27-Mar-1982 19:24:42    TOPS-20 Bliss-16 V2(212)
;                                                                27-Mar-1982 19:23:44    PA:<NEALE>MLX4.BLI.5 (38)

;          5293                !  : : : END 14 (END OF TEST FOR AN ACTIVE UNIT)
;          5294                !  : : END 13 (END OF TESTLOOP2)
;          5295                !  : END 12 (END OF LOGICAL UNIT SELECTION LOOP)
;          5296                !  END 11 (END OF LOOP READING SECTION)
;          5297                !--
;          5298
;          5299                incr LUN from 0 to (.L$UNIT - 1) do
;          5300                    begin                               !* 12 *  START OF LOGICAL UNIT SELECTION LOOP
;          5301  LOOP2 :
;          5302                    begin                               !* 13 *  START OF THE 2ND LOOP THAT COMPLETELY TESTS 1 UNIT
;          5303
;          5304                    if .DRIVE_STATUS [.LUN] eql ACTIVE
;          5305                    then
;          5306                        begin                           !* 14 *  START OF LOOP THAT TESTS AN ACTIVE UNIT
;          5307                        L$LUN = .LUN;
;          5308                        WPTR = WBUFF;
;          5309                        RPTR = RBUFF;
;          5310                        SECTOR = LOWEST;
;          5311
;          5312                        while .SECTOR lequ HIGHEST do
;          5313                            begin                       !* 15 *  START OF SECTOR SELECTION LOOP
;          5314                            WRDCNT = GET_WRDCNT (.SECTOR, HIGHEST);
;          5315                            SET_PTRS (.WRDCNT);
;          5316                            COMMAND = CHOOSE ();
;          5317
;          5318                            incr KOUNT from 1 to TIMES_TO_LOOP do
;          5319                                begin                   !* 16 *  START OF COUNTING LOOP FOR LOOP READING
;          5320
;          5321                            if .COMMAND eql read
;          5322                            then
;          5323                                begin
;          5324                                PTR = .RPTR;
;          5325                                VALUE = read (.LUN, .WRDCNT, .RPTR, .SECTOR);
;          5326                                end
;          5327                            else
;          5328                                begin
;          5329                                PTR = .WPTR;
;          5330                                VALUE = CHECK (.LUN, .WRDCNT, .WPTR, .SECTOR);
;          5331                                end;
;          5332
;          5333                            !+
;          5334                            !  SEE HOW SUCCESSFUL THE OPERATION WAS:
;          5335                            !-
;          5336
;          5337                            selectone .VALUE of             !SEE 'SYSERR' FOR DEFINITION
;          5338                                set                 !OF ERROR # CONTAINED IN 'VALUE'
;          5339
;          5340                                [0] :
;          5341
;          5342                                    if .COMMAND eql read
;          5343                                    then
;          5344                                        begin
```

SEQ 0207

```
;MLX4
;                    DEFINITION OF OPTION 2

;          5345
;          5346
;          5347
;          5348
;          5349
;          5350
;          5351
;          5352
;          5353
;          5354
;          5355
;          5356
;          5357
;          5358
;          5359
;          5360
;          5361
;          5362
;          5363
;          5364
;          5365
;          5366
;          5367
;          5368
;          5369
;          5370
;          5371
;          5372
;          5373
;          5374
;          5375
;          5376
;          5377
;          5378
;          5379
;          5380
;          5381
;          5382
;          5383
;          5384
;          5385
;          5386
;          5387
;          5388
;          5389
;          5390
;          5391
;          5392
;          5393
;          5394
;          5395
;          5396
```



```
        if (DBL_VALUE = DOUBLE_CHECK (.WPTR, .RPTR, .WRDCNT)) neq 0
        then
            begin
            SAYWHO (.LUN);
            PRINTB (SAY1, MSG5);
                !'ECC LOGIC FAILED TO DETECT DATA ERROR'
            PRINTB (FMT12A, ..DBL_VALUE, .DBL_VALUE);
                !'GOOD DATA: XXXXXX AT LOCATION YYYYYY'
            DBL_VALUE = .DBL_VALUE + BUFSIZ*2;
            PRINTB (FMT12B, ..DBL_VALUE, .DBL_VALUE);
                !'BAD DATA:  PPPPPP AT LOCATION QQQQQQ'
            WHY_DROPT [.LUN] = CODE_8;
            ERRDF (212, MSG1, 0);          !**** OPTION 2 ERROR 12 ****
            DODU (.LUN);
            leave LOOP2;          !JUMP JUST BEYOND END OF BLOCK * 4 *
            end;

        end;

    [1] :
        begin       !* 16A *        RETRY ALLOWED

        if RETRY (SIX, .COMMAND, .LUN, .WRDCNT, .PTR, .SECTOR) neq 0
        then        !THE RETRY FAILED -- SYSTEM FATAL ERROR
            begin
            WHY_DROPT [.LUN] = CODE_4;
            ERRDF (213, MSG1, 0);    !**** OPTION 2 ERROR 13 ****
            DODU (.LUN);
            leave LOOP2;     !JUMP JUST BEYOND END OF BLOCK * 13 *
            end;

        end;        !* 16A *

    [2] :
        begin       !* 16B *        FATAL CONTROLLER ERROR -- NO RETRY ALLOWED
        WHY_DROPT [.LUN] = CODE_5;
        ERRDF (214, MSG1, 0);         !**** OPTION 2 ERROR 14 ****
        DODU (.LUN);
        leave LOOP2;          !JUMP JUST BEYOND END OF BLOCK * 13 *
        end;        !* 16B *

    [3] :
        begin       !* 16C *        FATAL DRIVE ERROR -- NO RETRY ALLOWED
        ERRDF (215, MSG1, 0);         !**** OPTION 2 ERROR 15 ****
        WHY_DROPT [.LUN] = CODE_6;
        DODU (.LUN);
        leave LOOP2;          !JUMP JUST BEYOND END OF BLOCK * 13 *
        end;        !* 16C *

    [4] :
        begin       !* 16D *        UNRECOVERABLE DATA ERROR
```

```
8600 ;MLX4
8601 ;                  DEFINITION OF OPTION 2
8602
8603 ;          5397                                    ISOLATE ();
8604 ;          5398                                    ERRDF (216, MSG2, 0);       !**** OPTION 2 ERROR 16 ****
8605 ;          5399                                    WHY_DROPT [.LUN] = CODE_7;
8606 ;          5400                                    DODU (.LUN);
8607 ;          5401                                    leave LOOP2;            !JUMP JUST BEYOND END OF BLOCK * 13 *
8608 ;          5402                                    end;            !* 16D *
8609 ;          5403
8610 ;          5404                                [5] :
8611 ;          5405                                    begin           !* 16E *        RECOVERABLE DATA ERROR
8612 ;          5406                                    ISOLATE ();
8613 ;          5407
8614 ;          5408                                    if .ERROUT then PRINTB (FMT10B, .CHAN);     !'    BIT QQ'
8615 ;          5409
8616 ;          5410                                    OLDSEC = .MLEL;
8617 ;          5411                                    OLDCHN = .CHAN;
8618 ;          5412
8619 ;          5413                                    if RETRY (ONE, .COMMAND, .LUN, .WRDCNT, .PTR, .SECTOR) eql 5
8620 ;          5414                                    then
8621 ;          5415
8622 ;          5416                                        if ((.MLEL eql .OLDSEC) and (.CHAN eql .OLDCHN))
8623 ;          5417                                        then
8624 ;          5418                                            begin
8625 ;          5419
8626 ;          5420                                            if .ERROUT then ERRHRD (217, MSG4, 0);
8627 ;          5421
8628 ;          5422                                                !**** OPTION 2 ERROR 17 ****
8629 ;          5423                                            UP_HARD_COUNT (.LUN, .BOARD);
8630 ;          5424                                            end
8631 ;          5425                                        else
8632 ;          5426                                            begin
8633 ;          5427
8634 ;          5428                                            if .ERROUT then ERRSOFT (218, MSG3, 0);
8635 ;          5429
8636 ;          5430                                                !**** OPTION 2 ERROR 18 ****
8637 ;          5431                                            UP_SOFT_COUNT (.LUN, .BOARD);
8638 ;          5432                                            end
8639 ;          5433
8640 ;          5434                                    else
8641 ;          5435                                        begin
8642 ;          5436
8643 ;          5437                                        if .ERROUT then ERRSOFT (219, MSG3, 0); !**** OPTION 2 ERROR 19 ****
8644 ;          5438
8645 ;          5439                                        UP_SOFT_COUNT (.LUN, .BOARD);
8646 ;          5440                                        end;
8647 ;          5441
8648 ;          5442                                    end;            !* 16E *
8649 ;          5443                                tes;
8650 ;          5444
8651 ;          5445                                end;                !* 16 *  END OF COUNTING LOOP FOR LOOP READING
8652 ;          5446
8653 ;          5447                            WPTR = .WPTR + (.WRDCNT*2);
8654 ;          5448                            SECTOR = .SECTOR + (.WRDCNT/256);
```

SEQ 0209


```
8656 ;MLX4
8657 ;                  DEFINITION OF OPTION 2
8658
8659 ;          5449                        end;                        !* 15 *  END OF SECTOR SELECTION LOOP
8660 ;          5450
8661 ;          5451                    end;                            !* 14 *  END OF  TEST FOR AN ACTIVE UNIT
8662 ;          5452
8663 ;          5453                !+
8664 ;          5454                ! Test to see if this uut's address space is
8665 ;          5455                ! to be read for soft errors.  This test is
8666 ;          5456                  is intended for DMT purposes.
8667 ;          5457                :-
8668 ;          5458
8669 ;          5459                if .EFN$21                          !Is the background pattern to be read
8670 ;          5460                then
8671 ;          5461                    begin
8672 ;          5462
8673 ;          5463    !
8674 ;          5464    ! version czmlbb changed incr to incru
8675 ;          5465    !
8676 ;          5466                        incru SECTOR from LOWEST to HIGHEST do
8677 ;          5467
8678 ;          5468                            if read (.LUN, 256, RBUFF, .SECTOR) eql 5
8679 ;          5469                            then
8680 ;          5470                                begin
8681 ;          5471                                ISOLATE ();         !Find the failing bank and board no.
8682 ;          5472
8683 ;          5473                                if .ERROUT then PRINTB (FMT10B, .CHAN);
8684 ;          5474
8685 ;          5475                                ! Print where the error is
8686 ;          5476                                !
8687 ;          5477                                ! Save the contents of the ML error location
8688 ;          5478                                ! register so we can compare it to the new
8689 ;          5479                                ! contents of this register after the retry.
8690 ;          5480                                ! This is done to classify the error.
8691 ;          5481                                !
8692 ;          5482                                OLDSEC = .MLEL;
8693 ;          5483                                OLDCHN = .CHAN;
8694 ;          5484                                !
8695 ;          5485                                ! Do a classify retry call.  If the same error
8696 ;          5486                                ! occures then classify it as a hard error.  If
8697 ;          5487                                ! a different error occured or the error went away
8698 ;          5488                                ! then classify it as a soft error.
8699 ;          5489                                !
8700 ;          5490
8701 ;          5491                                if RETRY (ONE, .COMMAND, .LUN, 256, .PTR, .OLDSEC) eql 5
8702 ;          5492                                then
8703 ;          5493                                !
8704 ;          5494                                ! The same error occured so see if it is at the same
8705 ;          5495                                ! sector and channel number, if so then classify
8706 ;          5496                                ! it as a hard error else classiy it as a soft
8707 ;          5497                                ! error.
8708 ;          5498                                !
8709 ;          5499
8710 ;          5500                                    if ((.MLEL eql .OLDSEC) and (.CHAN eql .OLDCHN))
```

SEQ 0210


```
8712 ;MLX4
8713 ;                  DEFINITION OF OPTION 2
8714
8715 ;          5501                                        then
8716 ;          5502                                            begin       !Same error occured 'hard'
8717 ;          5503
8718 ;          5504                                            if .ERROUT  !Print error if enabled
8719 ;          5505                                            then
8720 ;          5506                                                begin
8721 ;        P 5507                                                ERRHRD (220,     !Error number
8722 ;        P 5508                                                    MSG4,        !Error message
8723 ;          5509                                                    0); !Additional message routine
8724 ;          5510                                                end;
8725 ;          5511
8726 ;          5512                                            UP_HARD_COUNT (.LUN, .BOARD);
8727 ;          5513                                            end
8728 ;          5514                                        else
8729 ;          5515                                            begin       !Not the same error 'soft'
8730 ;          5516
8731 ;          5517                                            if .ERROUT  !Print error if enabled
8732 ;          5518                                            then
8733 ;          5519                                                begin
8734 ;        P 5520                                                ERRSOFT (221,    !Error number
8735 ;        P 5521                                                    MSG3,        !Error message
8736 ;          5522                                                    0); !Additional message routine
8737 ;          5523                                                end;
8738 ;          5524
8739 ;          5525                                            UP_SOFT_COUNT (.LUN, .BOARD);
8740 ;          5526                                            end
8741 ;          5527
8742 ;          5528                                    else                    !Not the same error 'soft'
8743 ;          5529                                        begin
8744 ;          5530
8745 ;          5531                                        if .ERROUT      !Print error if enabled
8746 ;          5532                                        then
8747 ;          5533                                            begin
8748 ;        P 5534                                            ERRSOFT (222,        !Error number
8749 ;        P 5535                                                MSG3,    !Error message
8750 ;          5536                                                0);      !Additional message routine
8751 ;          5537                                            end;
8752 ;          5538
8753 ;          5539                                        UP_SOFT_COUNT (.LUN, .BOARD);
8754 ;          5540                                        end;
8755 ;          5541
8756 ;          5542                                end;
8757 ;          5543
8758 ;          5544                        end;
8759 ;          5545
8760 ;          5546                    end;                                !* 13 *  END OF TESTLOOP2
8761 ;          5547                    end;                                !* 12 *  END OF LOGICAL UNIT SELECTION LOOP
8762 ;          5548
8763 ;          5549                end;                                    !* 11 *  END OF LOOP READING SECTION
8764 ;          5550
8765 ;          5551            end;                                        !* 2 *  END OF PATTERN SELECTION LOOP
8766 ;          5552
```

```
 8768 ;MLX4
 8769 ;                  DEFINITION OF OPTION 2                  27-Mar-1982 19:24:42    TOPS-20 Bliss-16 V2(212)
 8770                                                            27-Mar-1982 19:23:44    PA:<NEALE>MLX4.BLI.5 (38)
 8771 ;          5553        return;
 8772 ;          5554        end;                                !* 1 *  END OF ROUTINE
 8776
 8777
                                                 .SBTTL  OPT2 DEFINITION OF OPTION 2
 8781 054054  004167  131254            OPT2:   JSR     R1,$SAVE5                       ;                       4982
 8782 054060  162706  000026                    SUB     #26,SP
 8783 054064  012746  007634                    MOV     #RTN2,-(SP)                     ;                       5045
 8784 054070  012746  007344                    MOV     #WRD34,-(SP)
 8785 054074  012746  007100                    MOV     #SAY2,-(SP)
 8786 054100  012746  000003                    MOV     #3,-(SP)
 8787 054104  010600                            MOV     SP,R0
 8788 054106  104414                            TRAP    14                      ; SP,*
 8789 054110  005067  156562                    CLR     PATTERN
 8790 054114  005767  156554                    TST     QUICK                           ;                       5047
 8791 054120  001403                            BEQ     1$                              ;                       5049
 8792 054122  012705  000001                    MOV     #1,R5
 8793 054126  000402                            BR      2$                      ; *,MAXPAT
 8794 054130  012705  000012            1$:     MOV     #12,R5                  ; *,MAXPAT
 8795 054134  010566  000032            2$:     MOV     R5,32(SP)               ; MAXPAT,*
 8796 054140  006366  000032                    ASL     32(SP)                          ;                       5051
 8797 054144  005066  000030                    CLR     30(SP)                  ; COUNT
 8798 054150  000167  003650                    JMP     74$
 8799 054154  004767  164706            3$:     JSR     PC,SELPAT                       ;                       5053
 8800 054160  005767  156512                    TST     PATTERN                         ;                       5055
 8801 054164  003413                            BLE     4$
 8802 054166  016746  156504                    MOV     PATTERN,-(SP)                   ;                       5057
 8803 054172  012746  010100                    MOV     #PHR9,-(SP)
 8804 054176  012746  006126                    MOV     #FMT1A,-(SP)
 8805 054202  012746  000003                    MOV     #3,-(SP)
 8806 054206  010600                            MOV     SP,R0
 8807 054210  104414                            TRAP    14                      ; SP,*
 8808 054212  000417                            BR      5$
 8809 054214  016766  156456  000034    4$:     MOV     PATTERN,34(SP)                  ;                       5055
 8810 054222  005466  000034                    NEG     34(SP)                  ; *,TEMP                5061
 8811 054226  016646  000034                    MOV     34(SP),-(SP)            ; TEMP
 8812 054232  012746  010100                    MOV     #PHR9,-(SP)             ; TEMP,*                5062
 8813 054236  012746  006142                    MOV     #FMT1B,-(SP)
 8814 054242  012746  000003                    MOV     #3,-(SP)
 8815 054246  010600                            MOV     SP,R0
 8816 054250  104414                            TRAP    14                      ; SP,*
 8817 054252  016716  156420            5$:     MOV     PATTERN,(SP)                    ;
 8818 054256  004767  165150                    JSR     PC,GEN2                         ;                       5066
 8819 054262  016766  125524  000034            MOV     L$UNIT,34(SP)                   ;
 8820 054270  005005                            CLR     R5                              ;                       5068
 8821 054272  000167  001462                    JMP     33$                     ; LUN
 8822 054276  010500                    6$:     MOV     R5,R0                   ; LUN,*                 5073
```

```
                                        ;MLX4
                                        ;               DEFINITION OF OPTION 2
8827 054300  006200                             ASR     R0
8828 054302  006200                             ASR     R0
8829 054304  006200                             ASR     R0
8830 054306  062700  034442                     ADD     #DRIVE.STATUS,R0
8831 054312  010046                             MOV     R0,-(SP)
8832 054314  010546                             MOV     R5,-(SP)
8833 054316  042716  177770                     BIC     #177770,(SP)            ; LUN,*
8834 054322  012746  000001                     MOV     #1,-(SP)
8835 054326  005046                             CLR     -(SP)
8836 054330  004767  130022                     JSR     PC,BL$GT2
8837 054334  062706  000010                     ADD     #10,SP
8838 054340  005300                             DEC     R0
8839 054342  001127                             BNE     11$
8840 054344  010567  125524                     MOV     R5,L$LUN                ; LUN,*                                 5076
8841 054350  012767  012670  156312             MOV     #WBUFF,WPTR             ;                                       5077
8842 054356  012767  022670  156306             MOV     #RBUFF,RPTR             ;                                       5078
8843 054364  010501                             MOV     R5,R1                   ; LUN,*                                 5079
8844 054366  006301                             ASL     R1
8845 054370  016104  034460                     MOV     LOW.SECT(R1),R4         ; *,SECTOR
8846 054374  020461  034500             7$:     CMP     R4,TOP.SECT(R1)         ; SECTOR,*                              5081
8847 054400  101110                             BHI     11$
8848 054402  010446                             MOV     R4,-(SP)                ; SECTOR,*                              5083
8849 054404  016146  034500                     MOV     TOP.SECT(R1),-(SP)
8850 054410  004767  172562                     JSR     PC,GET.WRDCNT
8851 054414  010002                             MOV     R0,R2                   ; *,WRDCNT
8852 054416  010216                             MOV     R2,(SP)                 ; WRDCNT,*                              5084
8853 054420  004767  172506                     JSR     PC,SET.PTRS
8854 054424  010516                             MOV     R5,(SP)                 ; LUN,*                                 5085
8855 054426  010246                             MOV     R2,-(SP)                ; WRDCNT,*
8856 054430  016746  156234                     MOV     WPTR,-(SP)
8857 054434  010446                             MOV     R4,-(SP)                ; SECTOR,*
8858 054436  004767  170762                     JSR     PC,WRITE
8859 054442  010003                             MOV     R0,R3                   ; *,VALUE
8860 054444  020327  000001                     CMP     R3,#1                   ; VALUE,*                               5091
8861 054450  001031                             BNE     8$
8862 054452  012746  000006                     MOV     #6,-(SP)                ;                                       5097
8863 054456  012746  045424                     MOV     #WRITE,-(SP)
8864 054462  010546                             MOV     R5,-(SP)                ; LUN,*
8865 054464  010246                             MOV     R2,-(SP)                ; WRDCNT,*
8866 054466  016746  156176                     MOV     WPTR,-(SP)
8867 054472  010446                             MOV     R4,-(SP)                ; SECTOR,*
8868 054474  004767  171524                     JSR     PC,RETRY
8869 054500  062706  000014                     ADD     #14,SP
8870 054504  005700                             TST     R0
8871 054506  001446                             BEQ     12$
8872 054510  112765  000004  034446             MOVB    #4,WHY.DROPT(R5)        ; *,*(LUN)                              5100
8873 054516  104455                             TRAP    55                      ;                                       5101
8874 054520  000311                             .WORD   311
8875 054522  011070                             .WORD   MSG1
8876 054524  000000                             .WORD   0
8877 054526  010500                             MOV     R5,R0                   ; LUN,*
8878 054530  104451                             TRAP    51                                                              5102
```

SEQ 0213

```
                                        ;MLX4
                                        ;                DEFINITION OF OPTION 2                                  27-Mar-1982 19:24:42    TOPS
                                                                                                                 27-Mar-1982 19:23:44    PA:<
 8883 054532  000431                                    BR      10$
 8884 054534  020327  000002                 8$:      CMP     R3,#2                   ;                       5103
 8885 054540  001012                                    BNE     9$                      ; VALUE,*               5091
 8886 054542  112765  000005  034446                   MOVB    #5,WHY.DROPT(R5)        ; *,*(LUN)              5110
 8887 054550  104455                                    TRAP    55                      ;                       5111
 8888 054552  000312                                    .WORD   312
 8889 054554  011070                                    .WORD   MSG1
 8890 054556  000000                                    .WORD   0
 8891 054560  010500                                    MOV     R5,R0                   ; LUN,*                 5112
 8892 054562  104451                                    TRAP    51
 8893 054564  000414                                    BR      10$                     ;                       5113
 8894 054566  020327  000003                 9$:      CMP     R3,#3                   ; VALUE,*               5091
 8895 054572  001014                                    BNE     12$
 8896 054574  104455                                    TRAP    55                      ;                       5118
 8897 054576  000313                                    .WORD   313
 8898 054600  011070                                    .WORD   MSG1
 8899 054602  000000                                    .WORD   0
 8900 054604  112765  000006  034446                   MOVB    #6,WHY.DROPT(R5)        ; *,*(LUN)              5119
 8901 054612  010500                                    MOV     R5,R0                   ; LUN,*                 5120
 8902 054614  104451                                    TRAP    51
 8903 054616  062706  000012                10$:      ADD     #12,SP                  ;                       5121
 8904 054622  000537                        11$:      BR      15$
 8905 054624  004767  171336                12$:      JSR     PC,CHOOSE               ;                       5125
 8906 054630  010066  000034                           MOV     R0,34(SP)               ; *,COMMAND
 8907 054634  005066  000040                           CLR     40(SP)                  ;                       5127
 8908 054640  020027  045612                           CMP     R0,#READ                ; COMMAND,*
 8909 054644  001015                                    BNE     13$
 8910 054646  005266  000040                           INC     40(SP)
 8911 054652  016766  156014  000036                   MOV     RPTR,36(SP)             ; *,PTR                 5130
 8912 054660  010546                                    MOV     R5,-(SP)                ; LUN,*                 5131
 8913 054662  010246                                    MOV     R2,-(SP)                ; WRDCNT,*
 8914 054664  016746  156002                           MOV     RPTR,-(SP)
 8915 054670  010446                                    MOV     R4,-(SP)                ; SECTOR,*
 8916 054672  004767  170714                           JSR     PC,READ
 8917 054676  000412                                    BR      14$
 8918 054700  016766  155764  000036  13$:      MOV     WPTR,36(SP)             ; *,PTR                 5135
 8919 054706  010546                                    MOV     R5,-(SP)                ; LUN,*                 5136
 8920 054710  010246                                    MOV     R2,-(SP)                ; WRDCNT,*
 8921 054712  016746  155752                           MOV     WPTR,-(SP)
 8922 054716  010446                                    MOV     R4,-(SP)                ; SECTOR,*
 8923 054720  004767  171054                           JSR     PC,CHECK
 8924 054724  010003                        14$:      MOV     R0,R3                   ; *,VALUE
 8925 054726  001076                                    BNE     16$
 8926 054730  032766  000001  000050                   BIT     #1,50(SP)               ;                       5143
 8927 054736  001513                                    BEQ     17$                     ;                       5148
 8928 054740  016746  155724                           MOV     WPTR,-(SP)
 8929 054744  016746  155722                           MOV     RPTR,-(SP)              ;                       5152
 8930 054750  010246                                    MOV     R2,-(SP)                ; WRDCNT,*
 8931 054752  004767  167350                           JSR     PC,DOUBLE.CHECK
 8932 054756  062706  000006                           ADD     #6,SP
 8933 054762  010066  000060                           MOV     R0,60(SP)               ; *,DBL.VALUE
 8934 054766  001477                                    BEQ     17$
```

```
                                        ;MLX4
                                        ;                DEFINITION OF OPTION 2                                  27-Mar-1982 19:24:42    TOPS
                                                                                                                 27-Mar-1982 19:23:44    PA:<
8939 054770  010546                                      MOV     R5,-(SP)                ; LUN,*                                         5155
8940 054772  004767  160526                              JSR     PC,SAYWHO
8941 054776  012716  011216                              MOV     #MSG5,(SP)              ;                                       5156
8942 055002  012746  007072                              MOV     #SAY1,-(SP)
8943 055006  012746  000002                              MOV     #2,-(SP)
8944 055012  010600                                      MOV     SP,R0                   ; SP,*
8945 055014  104414                                      TRAP    14
8946 055016  016616  000066                              MOV     66(SP),(SP)             ; DBL.VALUE,*                           5157
8947 055022  017646  000066                              MOV     @66(SP),-(SP)           ; DBL.VALUE,*
8948 055026  012746  006710                              MOV     #FMT12A,-(SP)
8949 055032  012746  000003                              MOV     #3,-(SP)
8950 055036  010600                                      MOV     SP,R0                   ; SP,*
8951 055040  104414                                      TRAP    14
8952 055042  062766  010000  000074                      ADD     #10000,74(SP)           ; *,DBL.VALUE                           5159
8953 055050  016616  000074                              MOV     74(SP),(SP)             ; DBL.VALUE,*                           5160
8954 055054  017646  000074                              MOV     @74(SP),-(SP)           ; DBL.VALUE,*
8955 055060  012746  006756                              MOV     #FMT12B,-(SP)
8956 055064  012746  000003                              MOV     #3,-(SP)
8957 055070  010600                                      MOV     SP,R0                   ; SP,*
8958 055072  104414                                      TRAP    14
8959 055074  112765  000010  034446                      MOVB    #10,WHY.DROPT(R5)       ; *,*(LUN)                              5162
8960 055102  104455                                      TRAP    55                      ;                                       5163
8961 055104  000314                                      .WORD   314
8962 055106  011070                                      .WORD   MSG1
8963 055110  000000                                      .WORD   0
8964 055112  010500                                      MOV     R5,R0                   ; LUN,*                                 5164
8965 055114  104451                                      TRAP    51
8966 055116  062706  000044                              ADD     #44,SP                  ;                                       5165
8967 055122  000510                          15$:        BR      23$
8968 055124  020327  000001                  16$:        CMP     R3,#1                   ; VALUE,*                               5143
8969 055130  001033                                      BNE     19$
8970 055132  012746  000006                              MOV     #6,-(SP)                ;                                       5173
8971 055136  016646  000046                              MOV     46(SP),-(SP)            ; COMMAND,*
8972 055142  010546                                      MOV     R5,-(SP)                ; LUN,*
8973 055144  010246                                      MOV     R2,-(SP)                ; WRDCNT,*
8974 055146  016646  000056                              MOV     56(SP),-(SP)            ; PTR,*
8975 055152  010446                                      MOV     R4,-(SP)                ; SECTOR,*
8976 055154  004767  171044                              JSR     PC,RETRY
8977 055160  062706  000014                              ADD     #14,SP
8978 055164  005700                                      TST     R0
8979 055166  001002                          17$:        BNE     18$
8980 055170  000167  000520                              JMP     31$
8981 055174  112765  000004  034446          18$:        MOVB    #4,WHY.DROPT(R5)        ; *,*(LUN)                              5176
8982 055202  104455                                      TRAP    55                      ;                                       5177
8983 055204  000315                                      .WORD   315
8984 055206  011070                                      .WORD   MSG1
8985 055210  000000                                      .WORD   0
8986 055212  010500                                      MOV     R5,R0                   ; LUN,*                                 5178
8987 055214  104451                                      TRAP    51
8988 055216  000450                                      BR      22$                     ;                                       5179
8989 055220  020327  000002                  19$:        CMP     R3,#2                   ; VALUE,*                               5143
8990 055224  001012                                      BNE     20$
```

```
                                        ;MLX4
                                        ;                DEFINITION OF OPTION 2                                  27-Mar-1982 19:24:42    TOPS
                                                                                                                 27-Mar-1982 19:23:44    PA:<
055226  112765  000005  034446          MOVB    #5,WHY.DROPT(R5)        ; *,*(LUN)                                                      5186
055234  104455                          TRAP    55                      ;                                                               5187
055236  000316                          .WORD   316
055240  011070                          .WORD   MSG1
055242  000000                          .WORD   0
055244  010500                          MOV     R5,R0                   ; LUN,*                                                         5188
055246  104451                          TRAP    51
055250  000433                          BR      22$                     ;                                                               5189
055252  020327  000003          20$:    CMP     R3,#3                   ; VALUE,*                                                       5143
055256  001012                          BNE     21$
055260  104455                          TRAP    55                      ;                                                               5194
055262  000317                          .WORD   317
055264  011070                          .WORD   MSG1
055266  000000                          .WORD   0
055270  112765  000006  034446          MOVB    #6,WHY.DROPT(R5)        ; *,*(LUN)                                                      5195
055276  010500                          MOV     R5,R0                   ; LUN,*                                                         5196
055300  104451                          TRAP    51
055302  000416                          BR      22$                     ;                                                               5197
055304  020327  000004          21$:    CMP     R3,#4                   ; VALUE,*                                                       5143
055310  001017                          BNE     24$
055312  004767  162014                  JSR     PC,ISOLATE              ;                                                               5202
055316  104455                          TRAP    55                      ;                                                               5203
055320  000320                          .WORD   320
055322  011120                          .WORD   MSG2
055324  000000                          .WORD   0
055326  112765  000007  034446          MOVB    #7,WHY.DROPT(R5)        ; *,*(LUN)                                                      5204
055334  010500                          MOV     R5,R0                   ; LUN,*                                                         5205
055336  104451                          TRAP    51
055340  062706  000022          22$:    ADD     #22,SP                  ;                                                               5206
055344  000167  000406          23$:    JMP     32$
055350  020327  000005          24$:    CMP     R3,#5                   ; VALUE,*                                                       5143
055354  001157                          BNE     31$
055356  004767  161750                  JSR     PC,ISOLATE              ;                                                               5211
055362  032767  000001  124670          BIT     #1,ERROUT               ;                                                               5213
055370  001423                          BEQ     25$
055372  017700  157012                  MOV     @ML.REG+42,R0
055376  006200                          ASR     R0
055400  006200                          ASR     R0
055402  006200                          ASR     R0
055404  006200                          ASR     R0
055406  006200                          ASR     R0
055410  006200                          ASR     R0
055412  042700  177700                  BIC     #177700,R0
055416  010046                          MOV     R0,-(SP)
055420  012746  006640                  MOV     #FMT10B,-(SP)
055424  012746  000002                  MOV     #2,-(SP)
055430  010600                          MOV     SP,R0                   ; SP,*
055432  104414                          TRAP    14
055434  062706  000006                  ADD     #6,SP
055440  017766  156746  000052  25$:    MOV     @ML.REG+44,52(SP)       ; *,OLDSEC                                                      5216
055446  017700  156736                  MOV     @ML.REG+42,R0           ;                                                               5217
055452  006200                          ASR     R0
```

```
                                ;MLX4
                                ;                       DEFINITION OF OPTION 2                                  27-Mar-1982 19:24:42    TOPS
                                ;                                                                               27-Mar-1982 19:23:44    PA:<
 9051 055454  006200                            ASR     R0
 9052 055456  006200                            ASR     R0
 9053 055460  006200                            ASR     R0
 9054 055462  006200                            ASR     R0
 9055 055464  006200                            ASR     R0
 9056 055466  042700  177700                    BIC     #177700,R0
 9057 055472  010066  000054                    MOV     R0,54(SP)       ; *,OLDCHN
 9058 055476  012746  000001                    MOV     #1,-(SP)        ;                                       5219
 9059 055502  016646  000046                    MOV     46(SP),-(SP)    ; COMMAND,*
 9060 055506  010546                            MOV     R5,-(SP)        ; LUN,*
 9061 055510  010246                            MOV     R2,-(SP)        ; WRDCNT,*
 9062 055512  016646  000056                    MOV     56(SP),-(SP)    ; PTR,*
 9063 055516  010446                            MOV     R4,-(SP)        ; SECTOR,*
 9064 055520  004767  170500                    JSR     PC,RETRY
 9065 055524  062706  000014                    ADD     #14,SP
 9066 055530  020027  000005                    CMP     R0,#5
 9067 055534  001051                            BNE     28$
 9068 055536  027766  156650  000052            CMP     @ML.REG+44,52(SP)       ; *,OLDSEC              5222
 9069 055544  001034                            BNE     27$
 9070 055546  016646  000054                    MOV     54(SP),-(SP)    ; OLDCHN,*
 9071 055552  017700  156632                    MOV     @ML.REG+42,R0
 9072 055556  006200                            ASR     R0
 9073 055560  006200                            ASR     R0
 9074 055562  006200                            ASR     R0
 9075 055564  006200                            ASR     R0
 9076 055566  006200                            ASR     R0
 9077 055570  006200                            ASR     R0
 9078 055572  042700  177700                    BIC     #177700,R0
 9079 055576  020026                            CMP     R0,(SP)+
 9080 055600  001016                            BNE     27$
 9081 055602  032767  000001  124450            BIT     #1,ERROUT       ;                                       5226
 9082 055610  001404                            BEQ     26$
 9083 055612  104456                            TRAP    56
 9084 055614  000321                            .WORD   321
 9085 055616  011202                            .WORD   MSG4
 9086 055620  000000                            .WORD   0
 9087 055622  010546                    26$:    MOV     R5,-(SP)        ; LUN,*                                 5228
 9088 055624  016746  156512                    MOV     BOARD,-(SP)
 9089 055630  004767  161566                    JSR     PC,UP.HARD.COUNT
 9090 055634  000426                            BR      30$
 9091 055636  032767  000001  124414    27$:    BIT     #1,ERROUT       ;                                       5222
 9092 055644  001415                            BEQ     29$             ;                                       5233
 9093 055646  104457                            TRAP    57
 9094 055650  000322                            .WORD   322
 9095 055652  011166                            .WORD   MSG3
 9096 055654  000000                            .WORD   0
 9097 055656  000410                            BR      29$
 9098 055660  032767  000001  124372    28$:    BIT     #1,ERROUT       ;                                       5235
 9099 055666  001404                            BEQ     29$             ;                                       5241
 9100 055670  104457                            TRAP    57
 9101 055672  000323                            .WORD   323
 9102 055674  011166                            .WORD   MSG3
```

```
                                        ;MLX4
                                        ;                       DEFINITION OF OPTION 2                                  27-Mar-1982 19:24:42    TOPS
                                                                                                                        27-Mar-1982 19:23:44    PA:<
9104
9105
9106
9107 055676  000000                             .WORD   0
9108 055700  010546                     29$:    MOV     R5,-(SP)
9109 055702  016746  156434                     MOV     BOARD,-(SP)             ; LUN,*                                 5243
9110 055706  004767  162172                     JSR     PC,UP.SOFT.COUNT
9111 055712  022626                     30$:    CMP     (SP)+,(SP)+
9112 055714  010200                     31$:    MOV     R2,R0                   ;                                       5210
9113 055716  006300                             ASL     R0                      ; WRDCNT,*                              5249
9114 055720  066700  154744                     ADD     WPTR,R0
9115 055724  010067  154740                     MOV     R0,WPTR
9116 055730  010216                             MOV     R2,(SP)
9117 055732  012746  000400                     MOV     #400,-(SP)              ; WRDCNT,*                              5250
9118 055736  004767  127226                     JSR     PC,BL$DIV
9119 055742  060400                             ADD     R4,R0
9120 055744  010004                             MOV     R0,R4                   ; SECTOR,*
9121 055746  062706  000024                     ADD     #24,SP                  ; *,SECTOR
9122 055752  000167  176416                     JMP     7$                      ;                                       5082
9123 055756  005205                     32$:    INC     R5                      ;                                       5081
9124 055760  020566  000034             33$:    CMP     R5,34(SP)               ; LUN                                   5068
9125 055764  002002                             BGE     34$                     ; LUN,*
9126 055766  000167  176304                     JMP     6$
9127 055772  005767  154676             34$:    TST     QUICK
9128 055776  001402                             BEQ     35$                     ;                                       5258
9129 056000  000167  002014                     JMP     73$
9130 056004  016766  124002  000034     35$:    MOV     L$UNIT,34(SP)           ;                                       5299
9131 056012  005005                             CLR     R5                      ; LUN
9132 056014  000167  001766                     JMP     72$
9133 056020  010500                     36$:    MOV     R5,R0                   ; LUN,*                                 5304
9134 056022  006200                             ASR     R0
9135 056024  006200                             ASR     R0
9136 056026  006200                             ASR     R0
9137 056030  062700  034442                     ADD     #DRIVE.STATUS,R0
9138 056034  010046                             MOV     R0,-(SP)
9139 056036  010546                             MOV     R5,-(SP)                ; LUN,*
9140 056040  042716  177770                     BIC     #177770,(SP)
9141 056044  012746  000001                     MOV     #1,-(SP)
9142 056050  005046                             CLR     -(SP)
9143 056052  004767  126300                     JSR     PC,BL$GT2
9144 056056  062706  000010                     ADD     #10,SP
9145 056062  005300                             DEC     R0
9146 056064  001402                             BEQ     38$
9147 056066  000167  001246             37$:    JMP     60$
9148 056072  010567  123776             38$:    MOV     R5,L$LUN                ; LUN,*                                 5307
9149 056076  012767  012670  154564             MOV     #WBUFF,WPTR             ;                                       5308
9150 056104  012767  022670  154560             MOV     #RBUFF,RPTR             ;                                       5309
9151 056112  010501                             MOV     R5,R1                   ; LUN,*                                 5310
9152 056114  006301                             ASL     R1
9153 056116  016104  034460                     MOV     LOW.SECT(R1),R4         ; *,SECTOR
9154 056122  020461  034500             39$:    CMP     R4,TOP.SECT(R1)         ; SECTOR,*                              5312
9155 056126  101357                             BHI     37$
9156 056130  010446                             MOV     R4,-(SP)                ; SECTOR,*                              5314
9157 056132  016146  034500                     MOV     TOP.SECT(R1),-(SP)
9158 056136  004767  171034                     JSR     PC,GET.WRDCNT
```

SEQ 0218

```
9160                                    ;MLX4
9161                                    ;
9162                                    ;       DEFINITION OF OPTION 2
9163 056142 010002                              MOV     R0,R2           ; *,WRDCNT
9164 056144 010216                              MOV     R2,(SP)         ; WRDCNT,*
9165 056146 004767 170760                       JSR     PC,SET.PTRS
9166 056152 004767 170010                       JSR     PC,CHOOSE
9167 056156 010066 000026                       MOV     R0,26(SP)       ; *,COMMAND
9168 056162 012766 000001 000032                MOV     #1,32(SP)       ; *,KOUNT
9169 056170 005066 000024               40$:    CLR     24(SP)
9170 056174 026627 000026 045612                CMP     26(SP),#READ    ; COMMAND,*
9171 056202 001015                              BNE     41$
9172 056204 005266 000024                       INC     24(SP)
9173 056210 016766 154456 000030                MOV     RPTR,30(SP)     ; *,PTR
9174 056216 010546                              MOV     R5,-(SP)        ; LUN,*
9175 056220 010246                              MOV     R2,-(SP)        ; WRDCNT,*
9176 056222 016746 154444                       MOV     RPTR,-(SP)
9177 056226 010446                              MOV     R4,-(SP)        ; SECTOR,*
9178 056230 004767 167356                       JSR     PC,READ
9179 056234 000412                              BR      42$
9180 056236 016766 154426 000030        41$:    MOV     WPTR,30(SP)     ; *,PTR
9181 056244 010546                              MOV     R5,-(SP)        ; LUN,*
9182 056246 010246                              MOV     R2,-(SP)        ; WRDCNT,*
9183 056250 016746 154414                       MOV     WPTR,-(SP)
9184 056254 010446                              MOV     R4,-(SP)        ; SECTOR,*
9185 056256 004767 167516                       JSR     PC,CHECK
9186 056262 010003                      42$:    MOV     R0,R3           ; *,VALUE
9187 056264 001076                              BNE     43$             ;
9188 056266 032766 000001 000034                BIT     #1,34(SP)       ;
9189 056274 001513                              BEQ     44$             ;
9190 056276 016746 154366                       MOV     WPTR,-(SP)      ;
9191 056302 016746 154364                       MOV     RPTR,-(SP)      ;
9192 056306 010246                              MOV     R2,-(SP)        ; WRDCNT,*
9193 056310 004767 166012                       JSR     PC,DOUBLE.CHECK
9194 056314 062706 000006                       ADD     #6,SP
9195 056320 010066 000052                       MOV     R0,52(SP)       ; *,DBL.VALUE
9196 056324 001477                              BEQ     44$
9197 056326 010546                              MOV     R5,-(SP)        ; LUN,*
9198 056330 004767 157170                       JSR     PC,SAYWHO
9199 056334 012716 011216                       MOV     #MSG5,(SP)      ;
9200 056340 012746 007072                       MOV     #SAY1,-(SP)
9201 056344 012746 000002                       MOV     #2,-(SP)
9202 056350 010600                              MOV     SP,R0           ; SP,*
9203 056352 104414                              TRAP    14
9204 056354 016616 000060                       MOV     60(SP),(SP)     ; DBL.VALUE,*
9205 056360 017646 000060                       MOV     @60(SP),-(SP)   ; DBL.VALUE,*
9206 056364 012746 006710                       MOV     #FMT12A,-(SP)
9207 056370 012746 000003                       MOV     #3,-(SP)
9208 056374 010600                              MOV     SP,R0           ; SP,*
9209 056376 104414                              TRAP    14
9210 056400 062766 010000 000066                ADD     #10000,66(SP)   ; *,DBL.VALUE
9211 056406 016616 000066                       MOV     66(SP),(SP)     ; DBL.VALUE,*
9212 056412 017646 000066                       MOV     @66(SP),-(SP)   ; DBL.VALUE,*
9213 056416 012746 006756                       MOV     #FMT12B,-(SP)
9214 056422 012746 000003                       MOV     #3,-(SP)
```

27-Mar-1982 19:24:42 TOPS
27-Mar-1982 19:23:44 PA:<

5315
5316
5318
5321
5324
5325
5329
5330
5337
5342
5346
5349
5350
5352
5354
5355

```
                                        ;MLX4
                                        ;                DEFINITION OF OPTION 2                                  27-Mar-1982 19:24:42    TOPS
                                                                                                                 27-Mar-1982 19:23:44    PA:<
9219 056426  010600                                     MOV     SP,R0                   ; SP,*
9220 056430  104414                                     TRAP    14
9221 056432  112765  000010  034446                    MOVB    #10,WHY.DROPT(R5)       ; *,*(LUN)                                      5357
9222 056440  104455                                     TRAP    55                      ;                                               5358
9223 056442  000324                                     .WORD   324
9224 056444  011070                                     .WORD   MSG1
9225 056446  000000                                     .WORD   0
9226 056450  010500                                     MOV     R5,R0                   ; LUN,*
9227 056452  104451                                     TRAP    51                                                                      5359
9228 056454  062706  000036                             ADD     #36,SP                  ;                                               5360
9229 056460  000510                                     BR      50$
9230 056462  020327  000001            43$:             CMP     R3,#1                   ; VALUE,*                                       5337
9231 056466  001033                                     BNE     46$
9232 056470  012746  000006                             MOV     #6,-(SP)                ;                                               5368
9233 056474  016646  000040                             MOV     40(SP),-(SP)            ; COMMAND,*
9234 056500  010546                                     MOV     R5,-(SP)                ; LUN,*
9235 056502  010246                                     MOV     R2,-(SP)                ; WRDCNT,*
9236 056504  016646  000050                             MOV     50(SP),-(SP)            ; PTR,*
9237 056510  010446                                     MOV     R4,-(SP)                ; SECTOR,*
9238 056512  004767  167506                             JSR     PC,RETRY
9239 056516  062706  000014                             ADD     #14,SP
9240 056522  005700                                     TST     R0
9241 056524  001002                            44$:     BNE     45$
9242 056526  000167  000520                             JMP     58$
9243 056532  112765  000004  034446          45$:       MOVB    #4,WHY.DROPT(R5)        ; *,*(LUN)                                      5371
9244 056540  104455                                     TRAP    55                      ;                                               5372
9245 056542  000325                                     .WORD   325
9246 056544  011070                                     .WORD   MSG1
9247 056546  000000                                     .WORD   0
9248 056550  010500                                     MOV     R5,R0                   ; LUN,*
9249 056552  104451                                     TRAP    51                                                                      5373
9250 056554  000450                                     BR      49$                     ;                                               5374
9251 056556  020327  000002            46$:             CMP     R3,#2                   ; VALUE,*                                       5337
9252 056562  001012                                     BNE     47$
9253 056564  112765  000005  034446                    MOVB    #5,WHY.DROPT(R5)        ; *,*(LUN)                                      5381
9254 056572  104455                                     TRAP    55                      ;                                               5382
9255 056574  000326                                     .WORD   326
9256 056576  011070                                     .WORD   MSG1
9257 056600  000000                                     .WORD   0
9258 056602  010500                                     MOV     R5,R0                   ; LUN,*
9259 056604  104451                                     TRAP    51                                                                      5383
9260 056606  000433                                     BR      49$                     ;                                               5384
9261 056610  020327  000003            47$:             CMP     R3,#3                   ; VALUE,*                                       5337
9262 056614  001012                                     BNE     48$
9263 056616  104455                                     TRAP    55                      ;                                               5389
9264 056620  000327                                     .WORD   327
9265 056622  011070                                     .WORD   MSG1
9266 056624  000000                                     .WORD   0
9267 056626  112765  000006  034446                    MOVB    #6,WHY.DROPT(R5)        ; *,*(LUN)                                      5390
9268 056634  010500                                     MOV     R5,R0                   ; LUN,*                                         5391
9269 056636  104451                                     TRAP    51
9270 056640  000416                                     BR      49$                     ;                                               5392
```

```
                                        ;MLX4
                                        ;                DEFINITION OF OPTION 2
 9275 056642  020327  000004            48$:    CMP     R3,#4                   ; VALUE,*                       5337
 9276 056646  001017                             BNE     51$
 9277 056650  004767  160456                     JSR     PC,ISOLATE              ;                               5397
 9278 056654  104455                             TRAP    55                      ;                               5398
 9279 056656  000330                             .WORD   330
 9280 056660  011120                             .WORD   MSG2
 9281 056662  000000                             .WORD   0
 9282 056664  112765  000007  034446            MOVB    #7,WHY.DROPT(R5)        ; *,*(LUN)                      5399
 9283 056672  010500                             MOV     R5,R0                   ; LUN,*                         5400
 9284 056674  104451                             TRAP    51
 9285 056676  062706  000014            49$:    ADD     #14,SP                  ;                               5401
 9286 056702  000167  001076            50$:    JMP     71$
 9287 056706  020327  000005            51$:    CMP     R3,#5                   ; VALUE,*                       5337
 9288 056712  001157                             BNE     58$
 9289 056714  004767  160412                     JSR     PC,ISOLATE              ;                               5406
 9290 056720  032767  000001  123332            BIT     #1,ERROUT               ;                               5408
 9291 056726  001423                             BEQ     52$
 9292 056730  017700  155454                     MOV     @ML.REG+42,R0
 9293 056734  006200                             ASR     R0
 9294 056736  006200                             ASR     R0
 9295 056740  006200                             ASR     R0
 9296 056742  006200                             ASR     R0
 9297 056744  006200                             ASR     R0
 9298 056746  006200                             ASR     R0
 9299 056750  042700  177700                     BIC     #177700,R0
 9300 056754  010046                             MOV     R0,-(SP)
 9301 056756  012746  006640                     MOV     #FMT108,-(SP)
 9302 056762  012746  000002                     MOV     #2,-(SP)
 9303 056766  010600                             MOV     SP,R0                   ; SP,*
 9304 056770  104414                             TRAP    14
 9305 056772  062706  000006                     ADD     #6,SP
 9306 056776  017766  155410  000044    52$:    MOV     @ML.REG+44,44(SP)       ; *,OLDSEC                      5410
 9307 057004  017700  155400                     MOV     @ML.REG+42,R0           ;                               5411
 9308 057010  006200                             ASR     R0
 9309 057012  006200                             ASR     R0
 9310 057014  006200                             ASR     R0
 9311 057016  006200                             ASR     R0
 9312 057020  006200                             ASR     R0
 9313 057022  006200                             ASR     R0
 9314 057024  042700  177700                     BIC     #177700,R0
 9315 057030  010066  000046                     MOV     R0,46(SP)
 9316 057034  012746  000001                     MOV     #1,-(SP)                ; *,OLDCHN                      5413
 9317 057040  016646  000040                     MOV     40(SP),-(SP)            ;
 9318 057044  010546                             MOV     R5,-(SP)                ; COMMAND,*
 9319 057046  010246                             MOV     R2,-(SP)                ; LUN,*
 9320 057050  016646  000050                     MOV     50(SP),-(SP)            ; WRDCNT,*
 9321 057054  010446                             MOV     R4,-(SP)                ; PTR,*
 9322 057056  004767  167142                     JSR     PC,RETRY                ; SECTOR,*
 9323 057062  062706  000014                     ADD     #14,SP
 9324 057066  020027  000005                     CMP     R0,#5
 9325 057072  001051                             BNE     55$
 9326 057074  027766  155312  000044            CMP     @ML.REG+44,44(SP)       ; *,OLDSEC                      5416
```

SEQ 0221

```
                                        ;MLX4
                                        ;                DEFINITION OF OPTION 2                                  27-Mar-1982 19:24:42    TOPS
                                                                                                                 27-Mar-1982 19:23:44    PA:<
 9331 057102  001034                                    BNE     54$
 9332 057104  016646  000046                           MOV     46(SP),-(SP)            ; OLDCHN,*
 9333 057110  017700  155274                           MOV     @ML.REG+42,R0
 9334 057114  006200                                    ASR     R0
 9335 057116  006200                                    ASR     R0
 9336 057120  006200                                    ASR     R0
 9337 057122  006200                                    ASR     R0
 9338 057124  006200                                    ASR     R0
 9339 057126  006200                                    ASR     R0
 9340 057130  042700  177700                           BIC     #177700,R0
 9341 057134  020026                                    CMP     R0,(SP)+
 9342 057136  001016                                    BNE     54$
 9343 057140  032767  000001  123112                    BIT     #1,ERROUT               ;                       5420
 9344 057146  001404                                    BEQ     53$
 9345 057150  104456                                    TRAP    56
 9346 057152  000331                                    .WORD   331
 9347 057154  011202                                    .WORD   MSG4
 9348 057156  000000                                    .WORD   0
 9349 057160  010546                            53$:    MOV     R5,-(SP)                ; LUN,*
 9350 057162  016746  155154                           MOV     BOARD,-(SP)             ;                       5423
 9351 057166  004767  160230                           JSR     PC,UP.HARD.COUNT
 9352 057172  000426                                    BR      57$
 9353 057174  032767  000001  123056  54$:    BIT     #1,ERROUT               ;                       5416
 9354 057202  001415                                    BEQ     56$             ;                       5428
 9355 057204  104457                                    TRAP    57
 9356 057206  000332                                    .WORD   332
 9357 057210  011166                                    .WORD   MSG3
 9358 057212  000000                                    .WORD   0
 9359 057214  000410                                    BR      56$
 9360 057216  032767  000001  123034  55$:    BIT     #1,ERROUT               ;                       5431
 9361 057224  001404                                    BEQ     56$             ;                       5437
 9362 057226  104457                                    TRAP    57
 9363 057230  000333                                    .WORD   333
 9364 057232  011166                                    .WORD   MSG3
 9365 057234  000000                                    .WORD   0
 9366 057236  010546                            56$:    MOV     R5,-(SP)
 9367 057240  016746  155076                           MOV     BOARD,-(SP)             ; LUN,*                 5439
 9368 057244  004767  160634                           JSR     PC,UP.SOFT.COUNT
 9369 057250  022626                            57$:    CMP     (SP)+,(SP)+
 9370 057252  062706  000010                    58$:    ADD     #10,SP                  ;                       5405
 9371 057256  005266  000032                            INC     32(SP)                  ;                       5319
 9372 057262  026627  000032  000002                    CMP     32(SP),#2               ; KOUNT                 5318
 9373 057270  003002                                    BGT     59$                     ; KOUNT,*
 9374 057272  000167  176672                           JMP     40$
 9375 057276  010200                            59$:    MOV     R2,R0
 9376 057300  006300                                    ASL     R0                      ; WRDCNT,*              5447
 9377 057302  066700  153362                           ADD     WPTR,R0
 9378 057306  010067  153356                           MOV     R0,WPTR
 9379 057312  010216                                    MOV     R2,(SP)
 9380 057314  012746  000400                           MOV     #400,-(SP)              ; WRDCNT,*              5448
 9381 057320  004767  125644                           JSR     PC,BL$DIV
 9382 057324  060400                                    ADD     R4,R0                   ; SECTOR,*
```

```
                                        ;MLX4                                                                    27-Mar-1982 19:24:42    TOPS
                                        ;                DEFINITION OF OPTION 2                                  27-Mar-1982 19:23:44    PA:<
 9387 057326 010004                             MOV     R0,R4                   ; *,SECTOR
 9388 057330 062706 000006                      ADD     #6,SP                   ;                                                       5313
 9389 057334 000167 176562                      JMP     39$                     ;                                                       5312
 9390 057340 032767 000001 122714  60$:         BIT     #1,EFN$21               ;                                                       5459
 9391 057346 001002                             BNE     61$
 9392 057350 000167 000430                      JMP     71$
 9393 057354 010500                61$:         MOV     R5,R0                   ; LUN,*                                                 5466
 9394 057356 006300                             ASL     R0
 9395 057360 016066 034500 000026               MOV     TOP.SECT(R0),26(SP)
 9396 057366 016001 034460                      MOV     LOW.SECT(R0),R1         ; *,SECTOR
 9397 057372 000577                             BR      70$
 9398 057374 010546                62$:         MOV     R5,-(SP)                ; LUN,*                                                 5468
 9399 057376 012746 000400                      MOV     #400,-(SP)
 9400 057402 012746 022670                      MOV     #RBUFF,-(SP)
 9401 057406 010146                             MOV     R1,-(SP)                ; SECTOR,*
 9402 057410 004767 166176                      JSR     PC,READ
 9403 057414 062706 000010                      ADD     #10,SP
 9404 057420 020027 000005                      CMP     R0,#5
 9405 057424 001161                             BNE     69$
 9406 057426 004767 157700                      JSR     PC,ISOLATE              ;                                                       5471
 9407 057432 032767 000001 122620               BIT     #1,ERROUT               ;                                                       5473
 9408 057440 001423                             BEQ     63$
 9409 057442 017700 154742                      MOV     @ML.REG+42,R0
 9410 057446 006200                             ASR     R0
 9411 057450 006200                             ASR     R0
 9412 057452 006200                             ASR     R0
 9413 057454 006200                             ASR     R0
 9414 057456 006200                             ASR     R0
 9415 057460 006200                             ASR     R0
 9416 057462 042700 177700                      BIC     #177700,R0
 9417 057466 010046                             MOV     R0,-(SP)
 9418 057470 012746 006640                      MOV     #FMT10B,-(SP)
 9419 057474 012746 000002                      MOV     #2,-(SP)
 9420 057500 010600                             MOV     SP,R0                   ; SP,*
 9421 057502 104414                             TRAP    14
 9422 057504 062706 000006                      ADD     #6,SP
 9423 057510 017766 154676 000030  63$:         MOV     @ML.REG+44,30(SP)       ; *,OLDSEC                                              5482
 9424 057516 017700 154666                      MOV     @ML.REG+42,R0           ;                                                       5483
 9425 057522 006200                             ASR     R0
 9426 057524 006200                             ASR     R0
 9427 057526 006200                             ASR     R0
 9428 057530 006200                             ASR     R0
 9429 057532 006200                             ASR     R0
 9430 057534 006200                             ASR     R0
 9431 057536 042700 177700                      BIC     #177700,R0
 9432 057542 010066 000032                      MOV     R0,32(SP)
 9433 057546 012746 000001                      MOV     #1,-(SP)                ;                                                       5491
 9434 057552 016646 000024                      MOV     24(SP),-(SP)            ; COMMAND,*
 9435 057556 010546                             MOV     R5,-(SP)                ; LUN,*
 9436 057560 012746 000400                      MOV     #400,-(SP)
 9437 057564 016646 000034                      MOV     34(SP),-(SP)            ; PTR,*
 9438 057570 016646 000042                      MOV     42(SP),-(SP)            ; OLDSEC,*
```

SEQ 0223

```
                                        ;MLX4
                                        ;                DEFINITION OF OPTION 2                                  27-Mar-1982 19:24:42    TOPS
                                                                                                                 27-Mar-1982 19:23:44    PA:<
 9440
 9441
 9442
 9443 057574  004767  166424                          JSR     PC,RETRY
 9444 057600  062706  000014                          ADD     #14,SP
 9445 057604  020027  000005                          CMP     R0,#5
 9446 057610  001051                                  BNE     66$
 9447 057612  027766  154574  0^0030                  CMP     @ML.REG+44,30(SP)       ; *,OLDSEC                      5500
 9448 057620  001034                                  BNE     65$
 9449 057622  016646  000032                          MOV     32(SP),-(SP)            ; OLDCHN,*
 9450 057626  017700  154556                          MOV     @ML.REG+42,R0
 9451 057632  006200                                  ASR     R0
 9452 057634  006200                                  ASR     R0
 9453 057636  006200                                  ASR     R0
 9454 057640  006200                                  ASR     R0
 9455 057642  006200                                  ASR     R0
 9456 057644  006200                                  ASR     R0
 9457 057646  042700  177700                          BIC     #177700,R0
 9458 057652  020026                                  CMP     R0,(SP)+
 9459 057654  001016                                  BNE     65$
 9460 057656  032767  000001  122374                  BIT     #1,ERROUT               ;                               5504
 9461 057664  001404                                  BEQ     64$
 9462 057666  104456                                  TRAP    56                      ;                               5509
 9463 057670  000334                                  .WORD   334
 9464 057672  011202                                  .WORD   MSG4
 9465 057674  000000                                  .WORD   0
 9466 057676  010546                          64$:    MOV     R5,-(SP)                ; LUN,*                         5512
 9467 057700  016746  154436                          MOV     BOARD,-(SP)
 9468 057704  004767  157512                          JSR     PC,UP.HARD.COUNT
 9469 057710  000426                                  BR      68$
 9470 057712  032767  000001  122340  65$:    BIT     #1,ERROUT               ;                               5500
 9471 057720  001415                                  BEQ     67$                     ;                               5517
 9472 057722  104457                                  TRAP    57
 9473 057724  000335                                  .WORD   335                     ;                               5522
 9474 057726  011166                                  .WORD   MSG3
 9475 057730  000000                                  .WORD   0
 9476 057732  000410                                  BR      67$
 9477 057734  032767  000001  122316  66$:    BIT     #1,ERROUT               ;                               5525
 9478 057742  001404                                  BEQ     67$                     ;                               5531
 9479 057744  104457                                  TRAP    57
 9480 057746  000336                                  .WORD   336                     ;                               5536
 9481 057750  011166                                  .WORD   MSG3
 9482 057752  000000                                  .WORD   0
 9483 057754  010546                          67$:    MOV     R5,-(SP)                ; LUN,*                         5539
 9484 057756  016746  154360                          MOV     BOARD,-(SP)
 9485 057762  004767  160116                          JSR     PC,UP.SOFT.COUNT
 9486 057766  022626                          68$:    CMP     (SP)+,(SP)+             ;                               5470
 9487 057770  005201                          69$:    INC     R1                      ; SECTOR                        5466
 9488 057772  020166  000026                  70$:    CMP     R1,26(SP)               ; SECTOR,*
 9489 057776  101002                                  BHI     71$
 9490 060000  000167  177370                          JMP     62$
 9491 060004  005205                          71$:    INC     R5                      ; LUN                           5299
 9492 060006  020566  000034                  72$:    CMP     R5,34(SP)               ; LUN,*
 9493 060012  002002                                  BGE     73$
 9494 060014  000167  176000                          JMP     36$
```

SEQ 0224

```
                                        ;MLX4
                                        ;               DEFINITION OF OPTION 2                                  27-Mar-1982 19:24:42    TOPS
                                                                                                                27-Mar-1982 19:23:44    PA:<
 060020  062706  000010                 73$:    ADD     #10,SP                                                                          5052
 060024  005266  000030                 74$:    INC     30(SP)                  ;                               COUNT                   5051
 060030  026666  000030  000032                 CMP     30(SP),32(SP)           ; COUNT,*
 060036  003002                                 BGT     75$
 060040  000167  174110                         JMP     3$
 060044  062706  000036                 75$:    ADD     #36,SP                  ;                                                       4982
 060050  000207                                 RTS     PC

                                        ; Routine Size:  1023 words
                                        ; Maximum stack depth per invocation:  43 words
```

```
;MLX4
;                DEFINITION OF OPTION 3                          27-Mar-1982 19:24:42    TOPS-20 Bliss-16 V2(212)
;                                                                27-Mar-1982 19:23:44    PA:<NEALE>MLX4.BLI.5 (39)
;
;          5555  %sbttl 'DEFINITION OF OPTION 3'
;          5556  routine OPT3 : novalue =
;          5557      begin                                       !* 1 *  START OF ROUTINE
;          5558
;          5559  !++
;          5560  !  ROUTINE:     OPT3
;          5561  !
;          5562  !  PURPOSE:     TO DO A UNIQUE DATA CHECK ON ALL AVAILABLE UNITS.
;          5563  !
;          5564  !  THE CODE FOR 'OPT3' IN BRIEF:
;          5565  !
;          5566  !  BEGIN 1 (START OF ROUTINE)
;          5567  !  SAY ROUTINE IS RUNNING
;          5568  !  INCR COMPLEMENT FLAG FROM 0 TO 1
;          5569  !  : BEGIN 2 (START OF COMPLEMENT FLAG SELECTION LOOP)
;          5570  !  : GENERATE THE PATTERN
;          5571  !  : INCR LUN FROM 0 TO LAST
;          5572  !  : : BEGIN 3 (START OF LOGICAL UNIT SELECTION LOOP)
;          5573  !  : : TESTLOOP:
;          5574  !  : : : BEGIN 4 (START OF LOOP THAT COMPLETELY TESTS 1 UNIT)
;          5575  !  : : : IF UNIT IS ACTIVE
;          5576  !  : : : THEN
;          5577  !  : : : : BEGIN 5 (START OF TEST FOR AN ACTIVE UNIT)
;          5578  !  : : : : INITIALIZE THE WRITE AND READ BUFFER POINTERS
;          5579  !  : : : : SECTOR = LOWEST
;          5580  !  : : : : WHILE SECTOR LEQ HIGHEST DO
;          5581  !  : : : : : BEGIN 6 (START OF SECTOR SELECTION LOOP)
;          5582  !  : : : : : GET_WRDCNT
;          5583  !  : : : : : SET_UP BUFFER POINTERS BEFORE TRANSFER
;          5584  !  : : : : : WRITE
;          5585  !  : : : : : LOOK FOR ERRORS (IF DROP UNIT, LEAVE TESTLOOP)
;          5586  !  : : : : : CHOOSE WHETHER TO WRITE CHECK OR READ
;          5587  !  : : : : : DO THE WRITE CHECK OR READ
;          5588  !  : : : : : LOOK FOR ERRORS (IF DROP UNIT, LEAVE TESTLOOP)
;          5589  !  : : : : : CHANGE BUFFER POINTERS AFTER TRANSFER
;          5590  !  : : : : : UPDATE SECTOR NUMBER BY NUMBER OF SECTORS IN PREVIOUS TRANSFER
;          5591  !  : : : : : END 6 (END OF SECTOR SELECTION LOOP)
;          5592  !  : : : : END 5 (END OF TEST FOR AN ACTIVE UNIT)
;          5593  !  : : : END 4 (END OF TESTLOOP)
;          5594  !  : : END 3 (END OF LOGICAL UNIT SELECTION LOOP)
;          5595  !  : END 2 (END OF COMPLEMENT FLAG SELECTION LOOP)
;          5596  !  RETURN
;          5597  !  END 1 (END OF ROUTINE)
;          5598  !--
;          5599
;          5600      label
;          5601          LOOP;
;          5602
;          5603      local
;          5604          WRDCNT,
;          5605          VALUE,
;          5606          OLDSEC,
```

```
9572 ;MLX4
9573 ;                  DEFINITION OF OPTION 3
9574
9575 ;          5607           OLDCHN,
9576 ;          5608           SECTOR,
9577 ;          5609           PTR,
9578 ;          5610           COMMAND,
9579 ;          5611           DBL_VALUE;
9580 ;          5612
9581 ;          5613       PRINTB (SAY2, WRD34, RTN3);
9582 ;          5614       !'RUNNING OPT3'
9583 ;          5615
9584 ;          5616       incr COMP_FLAG from 0 to 1 do
9585 ;          5617           begin                          !* 2 *  START OF COMPLEMENT FLAG SELECTION LOOP
9586 ;          5618           GEN3 (.COMP_FLAG);
9587 ;          5619
9588 ;          5620           incr LUN from 0 to (.L$UNIT - 1) do
9589 ;          5621               begin                      !* 3 *  START OF LOGICAL UNIT SELECTION LOOP
9590 ;          5622   LOOP :
9591 ;          5623               begin                      !* 4 *  START OF THE LOOP THAT COMPLETELY TESTS 1 UNIT
9592 ;          5624
9593 ;          5625               if .DRIVE_STATUS [.LUN] eql ACTIVE
9594 ;          5626               then
9595 ;          5627                   begin                  !* 5 *  START OF TEST FOR AN ACTIVE UNIT
9596 ;          5628                   L$LUN = .LUN;
9597 ;          5629                   WPTR = WBUFF;
9598 ;          5630                   RPTR = RBUFF;
9599 ;          5631                   SECTOR = LOWEST;
9600 ;          5632
9601 ;          5633                   while .SECTOR lequ HIGHEST do
9602 ;          5634                       begin              !* 6 *  START OF SECTOR SELECTION LOOP
9603 ;          5635                       WRDCNT = GET_WRDCNT (.SECTOR, HIGHEST);
9604 ;          5636                       SET_PTRS (.WRDCNT);
9605 ;          5637                       VALUE = write (.LUN, .WRDCNT, .WPTR, .SECTOR);
9606 ;          5638
9607 ;          5639                       !+
9608 ;          5640                       !  SEE HOW SUCCESSFUL THE WRITE WAS:
9609 ;          5641                       !-
9610 ;          5642
9611 ;          5643                       selectone .VALUE of          !SEE 'SYSERR' FOR DEFINITION
9612 ;          5644                           set                      !OF ERROR # CONTAINED IN 'VALUE'
9613 ;          5645
9614 ;          5646                           [1] :
9615 ;          5647                               begin                !* 6A * RETRY ALLOWED
9616 ;          5648
9617 ;          5649                               if RETRY (SIX, write, .LUN, .WRDCNT, .WPTR, .SECTOR) neq 0
9618 ;          5650                               then                 !THE RETRY FAILED -- SYSTEM FATAL ERROR
9619 ;          5651                                   begin
9620 ;          5652                                   WHY_DROPT [.LUN] = CODE_4;
9621 ;          5653                                   ERRDF (301, MSG1, 0);   !**** OPTION 3 ERROR 01 ****
9622 ;          5654                                   DODU (.LUN);
9623 ;          5655                                   leave LOOP;      !JUMP JUST BEYOND END OF BLOCK * 4 *
9624 ;          5656                                   end;
9625 ;          5657
9626 ;          5658                               end;                 !* 6A *
```

```
9628 ;MLX4
9629 ;
9630
9631 ;          5659
9632 ;          5660                        [2] :
9633 ;          5661                            begin                   !* 6B * FATAL CONTROLLER ERROR -- NO RETRY ALLOWED
9634 ;          5662                            WHY_DROPT [.LUN] = CODE_5;
9635 ;          5663                            ERRDF (302, MSG1, 0);       !**** OPTION 3 ERROR 02 ****
9636 ;          5664                            DODU (.LUN);
9637 ;          5665                            leave LOOP;             !JUMP JUST BEYOND END OF BLOCK * 4 *
9638 ;          5666                            end;                    !* 6B *
9639 ;          5667
9640 ;          5668                        [3] :
9641 ;          5669                            begin                   :* 6C * FATAL DRIVE ERROR -- NO RETRY ALLOWED
9642 ;          5670                            ERRDF (303, MSG1, 0);       !**** OPTION 3 ERROR 03 ****
9643 ;          5671                            WHY_DROPT [.LUN] = CODE_6;
9644 ;          5672                            DODU (.LUN);
9645 ;          5673                            leave LOOP;             !JUMP JUST BEYOND END OF BLOCK * 4 *
9646 ;          5674                            end;                    !* 6C *
9647 ;          5675                        tes;
9648 ;          5676
9649 ;          5677                    COMMAND = CHOOSE ();
9650 ;          5678
9651 ;          5679                    if .COMMAND eql read
9652 ;          5680                    then
9653 ;          5681                        begin
9654 ;          5682                        PTR = .RPTR;
9655 ;          5683                        VALUE = read (.LUN, .WRDCNT, .RPTR, .SECTOR);
9656 ;          5684                        end
9657 ;          5685                    else
9658 ;          5686                        begin
9659 ;          5687                        PTR = .WPTR;
9660 ;          5688                        VALUE = CHECK (.LUN, .WRDCNT, .WPTR, .SECTOR);
9661 ;          5689                        end;
9662 ;          5690
9663 ;          5691                    !+
9664 ;          5692                    ! SEE HOW SUCCESSFUL THE OPERATION WAS:
9665 ;          5693                    !-
9666 ;          5694
9667 ;          5695                    selectone .VALUE of             !SEE 'SYSERR' FOR DEFINITION
9668 ;          5696                        set                         !OF ERROR # CONTAINED IN 'VALUE'
9669 ;          5697
9670 ;          5698                        [0] :
9671 ;          5699
9672 ;          5700                            if .COMMAND eql read
9673 ;          5701                            then
9674 ;          5702                                begin
9675 ;          5703
9676 ;          5704                                if (DBL_VALUE = DOUBLE_CHECK (.WPTR, .RPTR, .WRDCNT)) neq 0
9677 ;          5705                                then
9678 ;          5706                                    begin
9679 ;          5707                                    SAYWHO (.LUN);
9680 ;          5708                                    PRINTB (SAY1, MSG5);        !'ECC LOGIC FAILED TO DETECT DATA ERROR'
9681 ;          5709                                    PRINTB (FMT12A, ..DBL_VALUE, .DBL_VALUE);
9682 ;          5710                                                !'GOOD DATA: XXXXXX AT LOCATION YYYYYY'
```

```
;MLX4
;                    DEFINITION OF OPTION 3                    27-Mar-1982 19:24:42    TOPS-20 Bliss-16 V2(212)
;                                                              27-Mar-1982 19:23:44    PA:<NEALE>MLX4.BLI.5 (39)

;          5711                                                    DBL_VALUE = .DBL_VALUE + BUFSIZ*2;
;          5712                                                    PRINTB (FMT12B, ..DBL_VALUE, .DBL_VALUE);
;          5713                                                            !'BAD DATA:  PPPPPP AT LOCATION QQQQQQ'
;          5714                                                    WHY_DROPT [.LUN] = CODE_8;
;          5715                                                    ERRDF (304, MSG1, 0);       !**** OPTION 3 ERROR 04 ****
;          5716                                                    DODU (.LUN);
;          5717                                                    leave LOOP; !JUMP JUST BEYOND END OF BLOCK * 4 *
;          5718                                                    end;
;          5719
;          5720                                                end;
;          5721
;          5722                                            [1] :
;          5723                                                begin                   !* 6D * RETRY ALLOWED
;          5724
;          5725                                                if RETRY (SIX, .COMMAND, .LUN, .WRDCNT, .PTR, .SECTOR) neq 0
;          5726                                                then                    !THE RETRY FAILED -- SYSTEM FATAL ERROR
;          5727                                                    begin
;          5728                                                    WHY_DROPT [.LUN] = CODE_4;
;          5729                                                    ERRDF (305, MSG1, 0);   !**** OPTION 3 ERROR 05 ****
;          5730                                                    DODU (.LUN);
;          5731                                                    leave LOOP;     !JUMP JUST BEYOND END OF BLOCK * 4 *
;          5732                                                    end;
;          5733
;          5734                                                end;                    !* 6D *
;          5735
;          5736                                            [2] :
;          5737                                                begin                   !* 6E * FATAL CONTROLLER ERROR -- NO RETRY ALLOWED
;          5738                                                WHY_DROPT [.LUN] = CODE_5;
;          5739                                                ERRDF (306, MSG1, 0);       !**** OPTION 3 ERROR 06 ****
;          5740                                                DODU (.LUN);
;          5741                                                leave LOOP;             !JUMP JUST BEYOND END OF BLOCK * 4 *
;          5742                                                end;                    !* 6E *
;          5743
;          5744                                            [3] :
;          5745                                                begin                   !* 6F * FATAL DRIVE ERROR -- NO RETRY ALLOWED
;          5746                                                ERRDF (307, MSG1, 0);       !**** OPTION 3 ERROR 07 ****
;          5747                                                WHY_DROPT [.LUN] = CODE_6;
;          5748                                                DODU (.LUN);
;          5749                                                leave LOOP;             !JUMP JUST BEYOND END OF BLOCK * 4 *
;          5750                                                end;                    !* 6F *
;          5751
;          5752                                            [4] :
;          5753                                                begin                   !* 6G * UNRECOVERABLE DATA ERROR
;          5754                                                ISOLATE ();
;          5755                                                ERRDF (308, MSG2, 0);       !**** OPTION 3 ERROR 08 ****
;          5756                                                WHY_DROPT [.LUN] = CODE_7;
;          5757                                                DODU (.LUN);
;          5758                                                leave LOOP;             !JUMP JUST BEYOND END OF BLOCK * 4 *
;          5759                                                end;                    !* 6G *
;          5760
;          5761                                            [5] :
;          5762                                                begin                   !* 6H * RECOVERABLE DATA ERROR
```

```
9740 ;MLX4
9741 ;
9742
9743 ;          5763                    ISOLATE ();
9744 ;          5764
9745 ;          5765                    if .ERROUT then PRINTB (FMT10B, .CHAN);
9746 ;          5766
9747 ;          5767                    !'    BIT QQ'
9748 ;          5768                    OLDSEC = .MLEL;
9749 ;          5769                    OLDCHN = .CHAN;
9750 ;          5770
9751 ;          5771                    if RETRY (ONE, .COMMAND, .LUN, .WRDCNT, .PTR, .SECTOR) eql 5
9752 ;          5772                    then
9753 ;          5773
9754 ;          5774                        if ((.MLEL eql .OLDSEC) and (.CHAN eql .OLDCHN))
9755 ;          5775                        then
9756 ;          5776                            begin
9757 ;          5777
9758 ;          5778                            if .ERROUT then ERRHRD (309, MSG4, 0);      !**** OPTION 3 ERROR 09 ****
9759 ;          5779
9760 ;          5780                            UP_HARD_COUNT (.LUN, .BOARD);
9761 ;          5781                            end
9762 ;          5782                        else
9763 ;          5783                            begin
9764 ;          5784
9765 ;          5785                            if .ERROUT then ERRSOFT (310, MSG3, 0);     !**** OPTION 3 ERROR 10 ****
9766 ;          5786
9767 ;          5787                            UP_SOFT_COUNT (.LUN, .BOARD);
9768 ;          5788                            end
9769 ;          5789
9770 ;          5790                    else
9771 ;          5791                        begin
9772 ;          5792
9773 ;          5793                        if .ERROUT then ERRSOFT (311, MSG3, 0); !**** OPTION 3 ERROR 11 ****
9774 ;          5794
9775 ;          5795                        UP_SOFT_COUNT (.LUN, .BOARD);
9776 ;          5796                        end;
9777 ;          5797
9778 ;          5798                    end;                    !* 6H *
9779 ;          5799                tes;
9780 ;          5800
9781 ;          5801            WPTR = .WPTR + (.WRDCNT*2);
9782 ;          5802            SECTOR = .SECTOR + (.WRDCNT/256);
9783 ;          5803            end;                            !* 6 *  END OF SECTOR SELECTION LOOP
9784 ;          5804
9785 ;          5805        end;                                !* 5 *  END OF TEST FOR AN ACTIVE UNIT
9786 ;          5806
9787 ;          5807    !+
9788 ;          5808    ! Test to see if this uut's address space is
9789 ;          5809    ! to be read for soft errors.  This test is
9790 ;          5810    ! is intended for DMT purposes.
9791 ;          5811    !-
9792 ;          5812
9793 ;          5813    if .EFN$21                              !Is the background pattern to be read
9794 ;          5814    then
```

```
;MLX4
;                DEFINITION OF OPTION 3
;          5815                  begin
;          5816
;          5817  !
;          5818  ! version czmlbb changed incr to incru
;          5819  !
;          5820                  incru SECTOR from LOWEST to HIGHEST do
;          5821
;          5822                      if read (.LUN, 256, RBUFF, .SECTOR) eql 5
;          5823                      then
;          5824                          begin
;          5825                          ISOLATE ();             !Find the failing bank and board no.
;          5826
;          5827                          if .ERROUT then PRINTB (FMT10B, .CHAN);
;          5828
;          5829                          ! Print where the error is
;          5830                          !
;          5831                          ! Save the contents of the ML error location
;          5832                          ! register so we can compare it to the new
;          5833                          ! contents of this register after the retry.
;          5834                          ! This is done to classify the error.
;          5835                          !
;          5836                          OLDSEC = .MLEL;
;          5837                          OLDCHN = .CHAN;
;          5838                          !
;          5839                          ! Do a classify retry call.  If the same error
;          5840                          ! occures then classify it as a hard error.  If
;          5841                          ! a different error occured or the error went away
;          5842                          ! then classify it as a soft error.
;          5843                          !
;          5844
;          5845                          if RETRY (ONE, .COMMAND, .LUN, 256, .PTR, .OLDSEC) eql 5
;          5846                          then
;          5847                          !
;          5848                          ! The same error occured so see if it is at the same
;          5849                          ! sector and channel number, if so then classify
;          5850                          ! it as a hard error else classiy it as a soft
;          5851                          ! error.
;          5852                          !
;          5853
;          5854                              if ((.MLEL eql .OLDSEC) and (.CHAN eql .OLDCHN))
;          5855                              then
;          5856                                  begin           !Same error occured 'hard'
;          5857
;          5858                                  if .ERROUT      !Print error if enabled
;          5859                                  then
;          5860                                      begin
;        P 5861                                      ERRHRD (312,        !Error number
;        P 5862                                          MSG4,   !Error message
;          5863                                          0);     !Additional message routine
;          5864                                      end;
;          5865
;          5866                                  UP_HARD_COUNT (.LUN, .BOARD);
```

```
;MLX4
;                DEFINITION OF OPTION 3                                  27-Mar-1982 19:24:42    TOPS-20 Bliss-16 V2(212)
;                                                                        27-Mar-1982 19:23:44    PA:<NEALE>MLX4.BLI.5 (39)
;
;       5867                                    end
;       5868                                else
;       5869                                    begin           !Not the same error 'soft'
;       5870
;       5871                                    if .ERROUT      !Print error if enabled
;       5872                                    then
;       5873                                        begin
;     P 5874                                        ERRSOFT (313,       !Error number
;     P 5875                                            MSG3,   !Error message
;       5876                                            0);     !Additional message routine
;       5877                                        end;
;       5878
;       5879                                    UP_SOFT_COUNT (.LUN, .BOARD);
;       5880                                    end
;       5881
;       5882                                else                        !Not the same error 'soft'
;       5883                                    begin
;       5884
;       5885                                    if .ERROUT          !Print error if enabled
;       5886                                    then
;       5887                                        begin
;     P 5888                                        ERRSOFT (314,   !Error number
;     P 5889                                            MSG3,       !Error message
;       5890                                            0);         !Additional message routine
;       5891                                        end;
;       5892
;       5893                                    UP_SOFT_COUNT (.LUN, .BOARD);
;       5894                                    end;
;       5895
;       5896                                end;
;       5897
;       5898                            end;
;       5899
;       5900                        end;                                !* 4 *  END OF TESTLOOP
;       5901                        end;                                !* 3 *  END OF LOGICAL UNIT SELECTION LOOP
;       5902
;       5903                    end;                                    !* 2 *  END OF COMPLEMENT FLAG SELECTION LOOP
;       5904
;       5905            return;
;       5906            end;                                            !* 1 *  END OF ROUTINE
;
;
                                        .SBTTL  OPT3 DEFINITION OF OPTION 3
060052  004167  125256          OPT3:   JSR     R1,$SAVE5                       ;                       5556
060056  162706  000020                  SUB     #20,SP
060062  012746  007642                  MOV     #RTN3,-(SP)                     ;                       5613
060066  012746  007344                  MOV     #WRD34,-(SP)
```

SEQ 0232

```
                                ;MLX4
                                ;                       DEFINITION OF OPTION 3                                  27-Mar-1982 19:24:42    TOPS
                                                                                                                27-Mar-1982 19:23:44    PA:<
 060072  012746  007100                        MOV     #SAY2,-(SP)
 060076  012746  000003                        MOV     #3,-(SP)
 060102  010600                                MOV     SP,R0                   ; SP,*
 060104  104414                                TRAP    14
 060106  005066  000026                        CLR     26(SP)                  ; COMP.FLAG                             5616
 060112  016646  000026                1$:     MOV     26(SP),-(SP)            ; COMP.FLAG,*                           5618
 060116  004767  161344                        JSR     PC,GEN3
 060122  016766  121664  000014              MOV     L$UNIT,14(SP)           ;                                       5620
 060130  005066  000012                        CLR     12(SP)                  ; LUN
 060134  000167  002240                        JMP     41$
 060140  016600  000012                2$:     MOV     12(SP),R0               ; LUN,*                                 5625
 060144  006200                                ASR     R0
 060146  006200                                ASR     R0
 060150  006200                                ASR     R0
 060152  062700  034442                        ADD     #DRIVE.STATUS,R0
 060156  010046                                MOV     R0,-(SP)
 060160  016646  000014                        MOV     14(SP),-(SP)            ; LUN,*
 060164  042716  177770                        BIC     #177770,(SP)
 060170  012746  000001                        MOV     #1,-(SP)
 060174  005046                                CLR     -(SP)
 060176  004767  124154                        JSR     PC,BL$GT2
 060202  062706  000010                        ADD     #10,SP
 060206  0C5300                                DEC     R0
 060210  001402                                BEQ     4$
 060212  000167  001502                3$:     JMP     29$
 060216  016667  000012  121650        4$:     MOV     12(SP),L$LUN            ; LUN,*                                 5628
 060224  012767  012670  152436              MOV     #WBUFF,WPTR             ;                                       5629
 060232  012767  022670  152432              MOV     #RBUFF,RPTR             ;                                       5630
 060240  016605  000012                        MOV     12(SP),R5               ; LUN,*                                 5631
 060244  006305                                ASL     R5
 060246  016503  034460                        MOV     LOW.SECT(R5),R3         ; *,SECTOR
 060252  020365  034500                5$:     CMP     R3,TOP.SECT(R5)         ; SECTOR,*                              5633
 060256  101355                                BHI     3$
 060260  010346                                MOV     R3,-(SP)                ; SECTOR,*                              5635
 060262  016546  034500                        MOV     TOP.SECT(R5),-(SP)
 060266  004767  166704                        JSR     PC,GET.WRDCNT
 060272  010002                                MOV     R0,R2                   ; *,WRDCNT
 060274  010216                                MOV     R2,(SP)                 ; WRDCNT,*                              5636
 060276  004767  166630                        JSR     PC,SET.PTRS
 060302  016616  000016                        MOV     16(SP),(SP)             ; LUN,*                                 5637
 060306  010246                                MOV     R2,-(SP)                ; WRDCNT,*
 060310  016746  152354                        MOV     WPTR,-(SP)
 060314  010346                                MOV     R3,-(SP)                ; SECTOR,*
 060316  004767  165102                        JSR     PC,WRITE
 060322  010004                                MOV     R0,R4                   ; *,VALUE
 060324  020427  000001                        CMP     R4,#1                   ; VALUE,*                               5643
 060330  001035                                BNE     6$
 060332  012746  000006                        MOV     #6,-(SP)                ;                                       5649
 060336  012746  045424                        MOV     #WRITE,-(SP)
 060342  016646  000030                        MOV     30(SP),-(SP)            ; LUN,*
 060346  010246                                MOV     R2,-(SP)                ; WRDCNT,*
 060350  016746  152314                        MOV     WPTR,-(SP)
```

SEQ 0233

```
                                        ;MLX4
                                        ;                DEFINITION OF OPTION 3                                  27-Mar-1982 19:24:42    TOPS
                                                                                                                 27-Mar-1982 19:23:44    PA:<
 060354  010346                                          MOV     R3,-(SP)        ; SECTOR,*
 060356  004767  165642                                 JSR     PC,RETRY
 060362  062706  000014                                 ADD     #14,SP
 060366  005700                                         TST     R0
 060370  001456                                         BEQ     9$
 060372  016601  000024                                 MOV     24(SP),R1
 060376  112761  000004  034446                         MOVB    #4,WHY.DROPT(R1)        ; LUN,*                 5652
 060404  104455                                         TRAP    55                                              5653
 060406  000455                                         .WORD   455             ;
 060410  011070                                         .WORD   MSG1
 060412  000000                                         .WORD   0
 060414  016600  000024                                 MOV     24(SP),R0       ; LUN,*                 5654
 060420  104451                                         TRAP    51
 060422  000436                                         BR      8$              ;                       5655
 060424  020427  000002                 6$:             CMP     R4,#2           ; VALUE,*               5643
 060430  001015                                         BNE     7$
 060432  016601  000024                                 MOV     24(SP),R1
 060436  112761  000005  034446                         MOVB    #5,WHY.DROPT(R1)        ; LUN,*                 5662
 060444  104455                                         TRAP    55                                              5663
 060446  000456                                         .WORD   456             ;
 060450  011070                                         .WORD   MSG1
 060452  000000                                         .WORD   0
 060454  016600  000024                                 MOV     24(SP),R0       ; LUN,*                 5664
 060460  104451                                         TRAP    51
 060462  000416                                         BR      8$              ;                       5665
 060464  020427  000003                 7$:             CMP     R4,#3           ; VALUE,*               5643
 060470  001016                                         BNE     9$
 060472  104455                                         TRAP    55                                              5670
 060474  000457                                         .WORD   457             ;
 060476  011070                                         .WORD   MSG1
 060500  000000                                         .WORD   0
 060502  016601  000024                                 MOV     24(SP),R1
 060506  112761  000006  034446                         MOVB    #6,WHY.DROPT(R1)        ; LUN,*                 5671
 060514  010100                                         MOV     R1,R0
 060516  104451                                         TRAP    51              ; LUN,*                 5672
 060520  062706  000012                 8$:             ADD     #12,SP                                          5673
 060524  000543                                         BR      14$             ;
 060526  004767  165434                 9$:             JSR     PC,CHOOSE                               5677
 060532  010066  000040                                 MOV     R0,40(SP)       ; *,COMMAND
 060536  005001                                         CLR     R1                                      5679
 060540  020027  045612                                 CMP     R0,#READ        ; COMMAND,*
 060544  001015                                         BNE     10$
 060546  005201                                         INC     R1
 060550  016766  152116  000036                         MOV     RPTR,36(SP)     ; *,PTR                 5682
 060556  016646  000024                                 MOV     24(SP),-(SP)    ; LUN,*                 5683
 060562  010246                                         MOV     R2,-(SP)        ; WRDCNT,*
 060564  016746  152102                                 MOV     RPTR,-(SP)
 060570  010346                                         MOV     R3,-(SP)        ; SECTOR,*
 060572  004767  165014                                 JSR     PC,READ
 060576  000413                                         BR      11$             ;
 060600  016766  152064  000036         10$:            MOV     WPTR,36(SP)     ; *,PTR                 5687
 060606  016646  000024                                 MOV     24(SP),-(SP)    ; LUN,*                 5688
```

SEQ 0234

```
10020                                   ;MLX4
10021                                   ;                       DEFINITION OF OPTION 3                          27-Mar-1982 19:24:42    TOPS
10022                                   ;                                                                       27-Mar-1982 19:23:44    PA:<
10023 060612  010246                            MOV     R2,-(SP)                ; WRDCNT,*
10024 060614  016746  152050                      MOV     WPTR,-(SP)
10025 060620  010346                            MOV     R3,-(SP)                ; SECTOR,*
10026 060622  004767  165152                      JSR     PC,CHECK
10027 060626  010004                    11$:    MOV     R0,R4                   ; *,VALUE
10028 060630  001102                            BNE     15$                     ;                                       5695
10029 060632  006001                            ROR     R1                      ;                                       5700
10030 060634  103402                            BLO     13$
10031 060636  000167  001014              12$:  JMP     28$
10032 060642  016746  152022              13$:  MOV     WPTR,-(SP)              ;                                       5704
10033 060646  016746  152020                      MOV     RPTR,-(SP)
10034 060652  010246                            MOV     R2,-(SP)                ; WRDCNT,*
10035 060654  004767  163446                      JSR     PC,DOUBLE.CHECK
10036 060660  062706  000006                      ADD     #6,SP
10037 060664  010066  000042                      MOV     R0,42(SP)               ; *,DBL.VALUE
10038 060670  001762                            BEQ     12$
10039 060672  016646  000034                      MOV     34(SP),-(SP)            ; LUN,*                         5707
10040 060676  004767  154622                      JSR     PC,SAYWHO
10041 060702  012716  011216                      MOV     #MSG5,(SP)              ;                               5708
10042 060706  012746  007072                      MOV     #SAY1,-(SP)
10043 060712  012746  000002                      MOV     #2,-(SP)
10044 060716  010600                            MOV     SP,R0                   ; SP,*
10045 060720  104414                            TRAP    14
10046 060722  016616  000050                      MOV     50(SP),(SP)             ; DBL.VALUE,*                   5709
10047 060726  017646  000050                      MOV     @50(SP),-(SP)           ; DBL.VALUE,*
10048 060732  012746  006710                      MOV     #FMT12A,-(SP)
10049 060736  012746  000003                      MOV     #3,-(SP)
10050 060742  010600                            MOV     SP,R0                   ; SP,*
10051 060744  104414                            TRAP    14
10052 060746  062766  010000  000056              ADD     #10000,56(SP)           ; *,DBL.VALUE                   5711
10053 060754  016616  000056                      MOV     56(SP),(SP)             ; DBL.VALUE,*                   5712
10054 060760  017646  000056                      MOV     @56(SP),-(SP)           ; DBL.VALUE,*
10055 060764  012746  006756                      MOV     #FMT12B,-(SP)
10056 060770  012746  000003                      MOV     #3,-(SP)
10057 060774  010600                            MOV     SP,R0                   ; SP,*
10058 060776  104414                            TRAP    14
10059 061000  016601  000056                      MOV     56(SP),R1               ; LUN,*                         5714
10060 061004  112761  000010  034446              MOVB    #10,WHY.DROPT(R1)
10061 061012  104455                            TRAP    55                      ;                               5715
10062 061014  000460                            .WORD   460
10063 061016  011070                            .WORD   MSG1
10064 061020  000000                            .WORD   0
10065 061022  016600  000056                      MOV     56(SP),R0               ; LUN,*                         5716
10066 061026  104451                            TRAP    51
10067 061030  062706  000044                      ADD     #44,SP                  ;                               5717
10068 061034  000521                    14$:    BR      20$
10069 061036  020427  000001            15$:    CMP     R4,#1                   ; VALUE,*                       5695
10070 061042  001035                            BNE     16$
10071 061044  012746  000006                      MOV     #6,-(SP)                ;                               5725
10072 061050  016646  000052                      MOV     52(SP),-(SP)            ; COMMAND,*
10073 061054  016646  000040                      MOV     40(SP),-(SP)            ; LUN,*
10074 061060  010246                            MOV     R2,-(SP)                ; WRDCNT,*
```

SEQ 0235

```
                                ;MLX4
                                ;                       DEFINITION OF OPTION 3                                  27-Mar-1982 19:24:42    TOPS
                                                                                                                27-Mar-1982 19:23:44    PA:<
10079 061062  016646  000056                   MOV     56(SP),-(SP)            ; PTR,*
10080 061066  010346                           MOV     R3,-(SP)                ; SECTOR,*
10081 061070  004767  165130                   JSR     PC,RETRY
10082 061074  062706  000014                   ADD     #14,SP
10083 061100  005700                           TST     R0
10084 061102  001655                           BEQ     12$
10085 061104  016601  000034                   MOV     34(SP),R1               ; LUN,*                                 5728
10086 061110  112761  000004  034446           MOVB    #4,WHY.DROPT(R1)
10087 061116  104455                           TRAP    55                      ;                                       5729
10088 061120  000461                           .WORD   461
10089 061122  011070                           .WORD   MSG1
10090 061124  000000                           .WORD   0
10091 061126  016600  000034                   MOV     34(SP),R0               ; LUN,*                                 5730
10092 061132  104451                           TRAP    51
10093 061134  000457                           BR      19$                     ;                                       5731
10094 061136  020427  000002           16$:    CMP     R4,#2                   ; VALUE,*                               5695
10095 061142  001015                           BNE     17$
10096 061144  016601  000034                   MOV     34(SP),R1               ; LUN,*                                 5738
10097 061150  112761  000005  034446           MOVB    #5,WHY.DROPT(R1)
10098 061156  104455                           TRAP    55                      ;                                       5739
10099 061160  000462                           .WORD   462
10100 061162  011070                           .WORD   MSG1
10101 061164  000000                           .WORD   0
10102 061166  016600  000034                   MOV     34(SP),R0               ; LUN,*                                 5740
10103 061172  104451                           TRAP    51
10104 061174  000437                           BR      19$                     ;                                       5741
10105 061176  020427  000003           17$:    CMP     R4,#3                   ; VALUE,*                               5695
10106 061202  001014                           BNE     18$
10107 061204  104455                           TRAP    55                      ;                                       5746
10108 061206  000463                           .WORD   463
10109 061210  011070                           .WORD   MSG1
10110 061212  000000                           .WORD   0
10111 061214  016601  000034                   MOV     34(SP),R1               ; LUN,*                                 5747
10112 061220  112761  000006  034446           MOVB    #6,WHY.DROPT(R1)
10113 061226  010100                           MOV     R1,R0                   ; LUN,*                                 5748
10114 061230  104451                           TRAP    51
10115 061232  000420                           BR      19$                     ;                                       5749
10116 061234  020427  000004           18$:    CMP     R4,#4                   ; VALUE,*                               5695
10117 061240  001021                           BNE     21$
10118 061242  004767  156064                   JSR     PC,ISOLATE              ;                                       5754
10119 061246  104455                           TRAP    55                      ;                                       5755
10120 061250  000464                           .WORD   464
10121 061252  011120                           .WORD   MSG2
10122 061254  000000                           .WORD   0
10123 061256  016601  000034                   MOV     34(SP),R1               ; LUN,*                                 5756
10124 061262  112761  000007  034446           MOVB    #7,WHY.DROPT(R1)
10125 061270  010100                           MOV     R1,R0                   ; LUN,*                                 5757
10126 061272  104451                           TRAP    51
10127 061274  062706  000022           19$:    ADD     #22,SP                  ;                                       5758
10128 061300  000167  001070           20$:    JMP     40$
10129 061304  020427  000005           21$:    CMP     R4,#5                   ; VALUE,*                               5695
10130 061310  001162                           BNE     28$
```

SEQ C236

```
10132                                   ;MLX4
10133                                   ;
10134                                   ;       DEFINITION OF OPTION 3
10135 061312  004767  156014                      JSR     PC,ISOLATE
10136 061316  032767  000001  120734              BIT     #1,ERROUT       ;
10137 061324  001423                              BEQ     22$             ;
10138 061326  017701  153056                      MOV     @ML.REG+42,R1
10139 061332  006201                              ASR     R1
10140 061334  006201                              ASR     R1
10141 061336  006201                              ASR     R1
10142 061340  006201                              ASR     R1
10143 061342  006201                              ASR     R1
10144 061344  006201                              ASR     R1
10145 061346  042701  177700                      BIC     #177700,R1
10146 061352  010146                              MOV     R1,-(SP)
10147 061354  012746  006640                      MOV     #FMT10B,-(SP)
10148 061360  012746  000002                      MOV     #2,-(SP)
10149 061364  010600                              MOV     SP,R0
10150 061366  104414                              TRAP    14              ; SP,*
10151 061370  062706  000006                      ADD     #6,SP
10152 061374  017766  153012  000044  22$:        MOV     @ML.REG+44,44(SP)       ; *,OLDSEC
10153 061402  017701  153002                      MOV     @ML.REG+42,R1           ;
10154 061406  006201                              ASR     R1
10155 061410  006201                              ASR     R1
10156 061412  006201                              ASR     R1
10157 061414  006201                              ASR     R1
10158 061416  006201                              ASR     R1
10159 061420  006201                              ASR     R1
10160 061422  042701  177700                      BIC     #177700,R1
10161 061426  010166  000040                      MOV     R1,40(SP)
10162 061432  012746  000001                      MOV     #1,-(SP)                ; *,OLDCHN
10163 061436  016646  000052                      MOV     52(SP),-(SP)            ;
10164 061442  016646  000040                      MOV     40(SP),-(SP)            ; COMMAND,*
10165 061446  010246                              MOV     R2,-(SP)                ; LUN,*
10166 061450  016646  000056                      MOV     56(SP),-(SP)            ; WRDCNT,*
10167 061454  010346                              MOV     R3,-(SP)                ; PTR,*
10168 061456  004767  164542                      JSR     PC,RETRY                ; SECTOR,*
10169 061462  062706  000014                      ADD     #14,SP
10170 061466  020027  000005                      CMP     R0,#5
10171 061472  001052                              BNE     25$
10172 061474  027766  152712  000044              CMP     @ML.REG+44,44(SP)       ; *,OLDSEC
10173 061502  001035                              BNE     24$
10174 061504  016600  000040                      MOV     40(SP),R0               ; OLDCHN,*
10175 061510  017701  152674                      MOV     @ML.REG+42,R1
10176 061514  006201                              ASR     R1
10177 061516  006201                              ASR     R1
10178 061520  006201                              ASR     R1
10179 061522  006201                              ASR     R1
10180 061524  006201                              ASR     R1
10181 061526  006201                              ASR     R1
10182 061530  042701  177700                      BIC     #177700,R1
10183 061534  020100                              CMP     R1,R0
10184 061536  001017                              BNE     24$
10185 061540  032767  000001  120512              BIT     #1,ERROUT       ;
10186 061546  001404                              BEQ     23$
```

27-Mar-1982 19:24:42 TOPS
27-Mar-1982 19:23:44 PA:<

5763
5765
5768
5769
5771
5774
5778

```
10188                                   ;MLX4
10189                                   ;
10190                                   ;               DEFINITION OF OPTION 3                  27-Mar-1982 19:24:42    TOPS
10191                                   ;                                                       27-Mar-1982 19:23:44    PA:<
10192 061550  104456                              TRAP    56
10193 061552  000465                              .WORD   465
10194 061554  011202                              .WORD   MSG4
10195 061556  000000                              .WORD   0
10196 061560  016646  000034            23$:     MOV     34(SP),-(SP)            ; LUN,*                         5780
10197 061564  016746  152552                      MOV     BOARD,-(SP)
10198 061570  004767  155626                      JSR     PC,UP.HARD.COUNT
10199 061574  000427                              BR      27$
10200 061576  032767  000001  120454  24$:      BIT     #1,ERROUT               ;                               5774
10201 061604  001415                              BEQ     26$                     ;                               5785
10202 061606  104457                              TRAP    57
10203 061610  000466                              .WORD   466
10204 061612  011166                              .WORD   MSG3
10205 061614  000000                              .WORD   0
10206 061616  000410                              BR      26$
10207 061620  032767  000001  120432  25$:      BIT     #1,ERROUT               ;                               5787
10208 061626  001404                              BEQ     26$                     ;                               5793
10209 061632  104457                              TRAP    57
10210 061634  000467                              .WORD   467
10211 061636  011166                              .WORD   MSG3
10212 061640  000000                              .WORD   0
10213 061644  016646  000034            26$:     MOV     34(SP),-(SP)            ; LUN,*                         5795
10214 061650  016746  152472                      MOV     BOARD,-(SP)
10215 061654  004767  156230                      JSR     PC,UP.SOFT.COUNT
10216 061656  022626                    27$:     CMP     (SP)+,(SP)+             ;                               5762
10217 061660  010200                    28$:     MOV     R2,R0                   ; WRDCNT,*                      5801
10218 061662  006300                              ASL     R0
10219 061666  066700  151002                      ADD     WPTR,R0
10220 061672  010067  150776                      MOV     R0,WPTR
10221 061674  010216                              MOV     R2,(SP)
10222 061700  012746  000400                      MOV     #400,-(SP)              ; WRDCNT,*                      5802
10223 061704  004767  123264                      JSR     PC,BL$DIV
10224 061706  060300                              ADD     R3,R0
10225 061710  010003                              MOV     R0,R3                   ; SECTOR,*
10226 061714  062706  000024                      ADD     #24,SP                  ; *,SECTOR
10227 061720  000167  176332                      JMP     5$                      ;                               5634
10228 061726  032767  000001  120334  29$:      BIT     #1,EFN$21               ;                               5633
10229 061730  001002                              BNE     30$                     ;                               5813
10230 061734  000167  000440                      JMP     40$
10231 061740  016600  000012            30$:     MOV     12(SP),R0               ; LUN,*                         5820
10232 061742  006300                              ASL     R0
10233 061746  016001  034500                      MOV     TOP.SECT(R0),R1
10234 061752  016005  034460                      MOV     LOW.SECT(R0),R5         ; *,SECTOR
10235 061756  C00167  000406                      JMP     39$
10236 061762  016646  000012            31$:     MOV     12(SP),-(SP)            ; LUN,*                         5822
10237 061766  012746  000400                      MOV     #400,-(SP)
10238 061772  012746  022670                      MOV     #RBUFF,-(SP)
10239 061774  010546                              MOV     R5,-(SP)                ; SECTOR,*
10240 062000  004767  163612                      JSR     PC,READ
10241 062004  062706  000010                      ADD     #10,SP
10242 062010  020027  000005                      CMP     R0,#5
10243 062014  001164                              BNE     38$
```

```
10244                                          ;MLX4
10245                                          :                    DEFINITION OF OPTION 3                                    27-Mar-1982 19:24:42    TOPS
10246                                                                                                                            27-Mar-1982 19:23:44    PA:<
10247 062012  004767  155314                             JSR     PC,ISOLATE
10248 062016  032767  000001  120234                     BIT     #1,ERROUT                  ;                                                          5825
10249 062024  001423                                     BEQ     32$                        ;                                                          5827
10250 062026  017700  152356                             MOV     @ML.REG+42,R0
10251 062032  006200                                     ASR     R0
10252 062034  006200                                     ASR     R0
10253 062036  006200                                     ASR     R0
10254 062040  006200                                     ASR     R0
10255 062042  006200                                     ASR     R0
10256 062044  006200                                     ASR     R0
10257 062046  042700  177700                             BIC     #177700,R0
10258 062052  010046                                     MOV     R0,-(SP)
10259 062054  012746  006640                             MOV     #FMT10B,-(SP)
10260 062060  012746  000002                             MOV     #2,-(SP)
10261 062064  010600                                     MOV     SP,R0
10262 062066  104414                                     TRAP    1,                         ; SP,*
10263 062070  062706  000006                             ADD     #6,SP
10264 062074  017766  152312  000022  32$:               MOV     @ML.REG+44,22(SP)          ; *,OLDSEC
10265 062102  017700  152302                             MOV     @ML.REG+42,R0              ;                                                          5836
10266 062106  006200                                     ASR     R0                                                                                    5837
10267 062110  006200                                     ASR     R0
10268 062112  006200                                     ASR     R0
10269 062114  006200                                     ASR     R0
10270 062116  006200                                     ASR     R0
10271 062120  006200                                     ASR     R0
10272 062122  042700  177700                             BIC     #177700,R0
10273 062126  010066  000016                             MOV     R0,16(SP)
10274 062132  012746  000001                             MOV     #1,-(SP)                   ; *,OLDCHN
10275 062136  016646  000030                             MOV     30(SP),-(SP)               ;                                                          5845
10276 062142  016646  000016                             MOV     16(SP),-(SP)               ; COMMAND,*
10277 062146  012746  000400                             MOV     #400,-(SP)                 ; LUN,*
10278 062152  016646  000034                             MOV     34(SP),-(SP)
10279 062156  016646  000034                             MOV     34(SP),-(SP)               ; PTR,*
10280 062162  004767  164036                             JSR     PC,RETRY                   ; OLDSEC,*
10281 062166  062706  000014                             ADD     #14,SP
10282 062172  020027  000005                             CMP     R0,#5
10283 062176  001052                                     BNE     35$
10284 062200  027766  152206  000022                     CMP     @ML.REG+44,22(SP)          ; *,OLDSEC
10285 062206  001035                                     BNE     34$                                                                                   5854
10286 062210  016646  000016                             MOV     16(SP),-(SP)               ; OLDCHN,*
10287 062214  017700  152170                             MOV     @ML.REG+42,R0
10288 062220  006200                                     ASR     R0
10289 062222  006200                                     ASR     R0
10290 062224  006200                                     ASR     R0
10291 062226  006200                                     ASR     R0
10292 062230  006200                                     ASR     R0
10293 062232  006200                                     ASR     R0
10294 062234  042700  177700                             BIC     #177700,R0
10295 062240  020026                                     CMP     R0,(SP)+
10296 062242  001017                                     BNE     34$
10297 062244  032767  000001  120006                     BIT     #1,ERROUT                  ;                                                          5858
10298 062252  001404                                     BEQ     33$
```

SEQ 0239

```
10300                                   ;MLX4
10301                                   ;                       DEFINITION OF OPTION 3                          27-Mar-1982 19:24:42    TOPS
10302                                                                                                           27-Mar-1982 19:23:44    PA:<
10303 062254  104456                              TRAP    56
10304 062256  000470                              .WORD   470             ;                                                       5863
10305 062260  011202                              .WORD   MSG4
10306 062262  000000                              .WORD   0
10307 062264  016646  000012              33$:    MOV     12(SP),-(SP)    ; LUN,*                                                 5866
10308 062270  016746  152046                      MOV     BOARD,-(SP)
10309 062274  004767  155122                      JSR     PC,UP.HARD.COUNT
10310 062300  000427                              BR      37$
10311 062302  032767  000001  117750    34$:    BIT     #1,ERROUT       ;                                                       5854
10312 062310  001415                              BEQ     36$             ;                                                       5871
10313 062312  104457                              TRAP    57
10314 062314  000471                              .WORD   471             ;                                                       5876
10315 062316  011166                              .WORD   MSG3
10316 062320  000000                              .WORD   0
10317 062322  000410                              BR      36$
10318 062324  032767  000001  117726    35$:    BIT     #1,ERROUT       ;                                                       5879
10319 062332  001404                              BEQ     36$             ;                                                       5885
10320 062334  104457                              TRAP    57
10321 062336  000472                              .WORD   472             ;                                                       5890
10322 062340  011166                              .WORD   MSG3
10323 062342  000000                              .WORD   0
10324 062344  016646  000012              36$:    MOV     12(SP),-(SP)    ; LUN,*                                                 5893
10325 062350  016746  151766                      MOV     BOARD,-(SP)
10326 062354  004767  155524                      JSR     PC,UP.SOFT.COUNT
10327 062360  022626                      37$:    CMP     (SP)+,(SP)+     ;                                                       5824
10328 062362  005205                      38$:    INC     R5              ; SECTOR                                                5820
10329 062364  020501                      39$:    CMP     R5,R1           ; SECTOR,*
10330 062366  101002                              BHI     40$
10331 062370  000167  177362                      JMP     31$
10332 062374  005266  000012              40$:    INC     12(SP)          ; LUN                                                   5620
10333 062400  026666  000012  000014      41$:    CMP     12(SP),14(SP)   ; LUN,*
10334 062406  002002                              BGE     42$
10335 062410  000167  175524                      JMP     2$
10336 062414  005726                      42$:    TST     (SP)+           ;                                                       5617
10337 062416  005266  000026                      INC     26(SP)          ; COMP.FLAG                                             5616
10338 062422  026627  000026  000001            CMP     26(SP),#1       ; COMP.FLAG,*
10339 062430  003002                              BGT     43$
10340 062432  000167  175454                      JMP     1$
10341 062436  062706  000030              43$:    ADD     #30,SP          ;                                                       5556
10342 062442  000207                              RTS     PC
10343
10344                                   ; Routine Size:  637 words
10345                                   ; Maximum stack depth per invocation:  37 words
10350
10351
```

SEQ 0240


```
10353 ;MLX4
10354 ;                 DEFINITION OF OPTION 4
10355 ;
10356 ;          5907  %sbttl 'DEFINITION OF OPTION 4'
10357 ;          5908  routine OPT4 : novalue =
10358 ;          5909      begin
10359 ;          5910
10360 ;          5911  !++
10361 ;          5912  !  ROUTINE:     OPT4
10362 ;          5913  !
10363 ;          5914  !  PURPOSE:     TO LOOK FOR INTEREACTIONS BETWEEN SECTORS
10364 ;          5915  !               USING A MARCH TEST.
10365 ;          5916  !
10366 ;          5917  !  THE CODE IN BRIEF:
10367 ;          5918  !
10368 ;          5919  !  BEGIN 1 (START OF ROUTINE)
10369 ;          5920  !  SAY ROUTINE IS RUNNING
10370 ;          5921  !  WORD COUNT = 256
10371 ;          5922  !  GENERATE A BUFFER OF DATA
10372 ;          5923  !  GENERATE A BUFFER OF COMP
10373 ;          5924  !  INITIALIZE POINTERS TO 4 BUFFERS FOR WRITE/READ DATA/COMP
10374 ;          5925  !  INCR LUN FROM 0 TO LAST
10375 ;          5926  !  : BEGIN 2 (START OF LOGICAL UNIT SELECTION LOOP)
10376 ;          5927  !  : TESTLOOP:
10377 ;          5928  !  : : BEGIN 3 (START OF LOOP THAT COMPLETELY TESTS 1 UNIT)
10378 ;          5929  !  : : IF UNIT IS ACTIVE
10379 ;          5930  !  : : THEN
10380 ;          5931  !  : : : BEGIN 4 (START OF TEST FOR AN ACTIVE UNIT)
10381 ;          5932  !  : : : INCR SECTOR FROM LOWEST TO HIGHEST
10382 ;          5933  !  : : : : BEGIN 4A (START OF 1ST SECTOR SELECTION LOOP)
10383 ;          5934  !  : : : : WRITE 'DATA'
10384 ;          5935  !  : : : : LOOK FOR ERRORS (IF DROP UNIT, LEAVE TESTLOOP)
10385 ;          5936  !  : : : : END 4A (END OF 1ST SECTOR SELECTION LOOP)
10386 ;          5937  !  : : : CHOOSE WHETHER TO WRITE CHECK OR READ 'DATA'
10387 ;          5938  !  : : : INCR SECTOR FROM LOWEST TO HIGHEST
10388 ;          5939  !  : : : : BEGIN 4B (START OF 2ND SECTOR SELECTION LOOP)
10389 ;          5940  !  : : : : DO THE WRITE CHECK OR READ OF 'DATA'
10390 ;          5941  !  : : : : LOOK FOR ERRORS (IF DROP UNIT, LEAVE TESTLOOP)
10391 ;          5942  !  : : : : WRITE 'COMP'
10392 ;          5943  !  : : : : LOOK FOR ERRORS (IF DROP UNIT, LEAVE TESTLOOP)
10393 ;          5944  !  : : : : END 4B (END OF 2ND SECTOR SELECTION LOOP)
10394 ;          5945  !  : : : CHOOSE WHETHER TO WRITE CHECK OR READ 'COMP'
10395 ;          5946  !  : : : DECR SECTOR FROM HIGHEST TO LOWEST
10396 ;          5947  !  : : : : BEGIN 4C (START OF 3RD SECTOR SELECTION LOOP)
10397 ;          5948  !  : : : : DO THE WRITE CHECK OR READ OF 'COMP'
10398 ;          5949  !  : : : : LOOK FOR ERRORS (IF DROP UNIT, LEAVE TESTLOOP)
10399 ;          5950  !  : : : : WRITE DATA
10400 ;          5951  !  : : : : LOOK FOR ERRORS (IF DROP UNIT, LEAVE TESTLOOP)
10401 ;          5952  !  : : : : END 4C (END OF 3RD SECTOR SFLECTION LOOP)
10402 ;          5953  !  : : : CHOOSE WHETHER TO WRITE CHECK OR READ 'DATA'
10403 ;          5954  !  : : : INCR SECTOR FROM LOWEST TO HIGHEST
10404 ;          5955  !  : : : : BEGIN 4D (START OF 4TH SECTOR SELECTION LOOP)
10405 ;          5956  !  : : : : DO THE WRITE CHECK OR READ OF 'DATA'
10406 ;          5957  !  : : : : LOOK FOR ERRORS (IF DROP UNIT, LEAVE TESTLOOP)
10407 ;          5958  !  : : : : END 4D (END OF 4TH SECTOR SELECTION LOOP)
```

```
10353 ;MLX4
10354 ;                 DEFINITION OF OPTION 4
10355 ;
10356 ;          5907  %sbttl 'DEFINITION OF OPTION 4'
10357 ;          5908  routine OPT4 : novalue =
10358 ;          5909      begin
10359 ;          5910
10360 ;          5911  !++
10361 ;          5912  !  ROUTINE:     OPT4
10362 ;          5913  !
10363 ;          5914  !  PURPOSE:     TO LOOK FOR INTEREACTIONS BETWEEN SECTORS
10364 ;          5915  !               USING A MARCH TEST.
10365 ;          5916  !
10366 ;          5917  !  THE CODE IN BRIEF:
10367 ;          5918  !
10368 ;          5919  !  BEGIN 1 (START OF ROUTINE)
10369 ;          5920  !  SAY ROUTINE IS RUNNING
10370 ;          5921  !  WORD COUNT = 256
10371 ;          5922  !  GENERATE A BUFFER OF DATA
10372 ;          5923  !  GENERATE A BUFFER OF COMP
10373 ;          5924  !  INITIALIZE POINTERS TO 4 BUFFERS FOR WRITE/READ DATA/COMP
10374 ;          5925  !  INCR LUN FROM 0 TO LAST
10375 ;          5926  !  : BEGIN 2 (START OF LOGICAL UNIT SELECTION LOOP)
10376 ;          5927  !  : TESTLOOP:
10377 ;          5928  !  : : BEGIN 3 (START OF LOOP THAT COMPLETELY TESTS 1 UNIT)
10378 ;          5929  !  : : IF UNIT IS ACTIVE
10379 ;          5930  !  : : THEN
10380 ;          5931  !  : : : BEGIN 4 (START OF TEST FOR AN ACTIVE UNIT)
10381 ;          5932  !  : : : INCR SECTOR FROM LOWEST TO HIGHEST
10382 ;          5933  !  : : : : BEGIN 4A (START OF 1ST SECTOR SELECTION LOOP)
10383 ;          5934  !  : : : : WRITE 'DATA'
10384 ;          5935  !  : : : : LOOK FOR ERRORS (IF DROP UNIT, LEAVE TESTLOOP)
10385 ;          5936  !  : : : : END 4A (END OF 1ST SECTOR SELECTION LOOP)
10386 ;          5937  !  : : : CHOOSE WHETHER TO WRITE CHECK OR READ 'DATA'
10387 ;          5938  !  : : : INCR SECTOR FROM LOWEST TO HIGHEST
10388 ;          5939  !  : : : : BEGIN 4B (START OF 2ND SECTOR SELECTION LOOP)
10389 ;          5940  !  : : : : DO THE WRITE CHECK OR READ OF 'DATA'
10390 ;          5941  !  : : : : LOOK FOR ERRORS (IF DROP UNIT, LEAVE TESTLOOP)
10391 ;          5942  !  : : : : WRITE 'COMP'
10392 ;          5943  !  : : : : LOOK FOR ERRORS (IF DROP UNIT, LEAVE TESTLOOP)
10393 ;          5944  !  : : : : END 4B (END OF 2ND SECTOR SELECTION LOOP)
10394 ;          5945  !  : : : CHOOSE WHETHER TO WRITE CHECK OR READ 'COMP'
10395 ;          5946  !  : : : DECR SECTOR FROM HIGHEST TO LOWEST
10396 ;          5947  !  : : : : BEGIN 4C (START OF 3RD SECTOR SELECTION LOOP)
10397 ;          5948  !  : : : : DO THE WRITE CHECK OR READ OF 'COMP'
10398 ;          5949  !  : : : : LOOK FOR ERRORS (IF DROP UNIT, LEAVE TESTLOOP)
10399 ;          5950  !  : : : : WRITE DATA
10400 ;          5951  !  : : : : LOOK FOR ERRORS (IF DROP UNIT, LEAVE TESTLOOP)
10401 ;          5952  !  : : : : END 4C (END OF 3RD SECTOR SFLECTION LOOP)
10402 ;          5953  !  : : : CHOOSE WHETHER TO WRITE CHECK OR READ 'DATA'
10403 ;          5954  !  : : : INCR SECTOR FROM LOWEST TO HIGHEST
10404 ;          5955  !  : : : : BEGIN 4D (START OF 4TH SECTOR SELECTION LOOP)
10405 ;          5956  !  : : : : DO THE WRITE CHECK OR READ OF 'DATA'
10406 ;          5957  !  : : : : LOOK FOR ERRORS (IF DROP UNIT, LEAVE TESTLOOP)
10407 ;          5958  !  : : : : END 4D (END OF 4TH SECTOR SELECTION LOOP)
```

SEQ 0241


```
10409 ;MLX4
10410 ;                 DEFINITION OF OPTION 4
10411 ;
10412 ;          5959 !  : : : END 4 (END OF TEST FOR AN ACTIVE UNIT)
10413 ;          5960 !  : : END 3 (END OF TESTLOOP)
10414 ;          5961 !  : END 2 (END OF LOGICAL UNIT SELECTION LOOP)
10415 ;          5962 !  RETURN
10416 ;          5963 !  END 1 (END OF ROUTINE)
10417 ;          5964 !--
10418 ;          5965
10419 ;          5966      label
10420 ;          5967          LOOP;
10421 ;          5968
10422 ;          5969      local
10423 ;          5970          VALUE,
10424 ;          5971          OLDSEC,
10425 ;          5972          OLDCHN,
10426 ;          5973          PTR,
10427 ;          5974          COMMAND,
10428 ;          5975          DBL_VALUE,
10429 ;          5976          WDPTR,
10430 ;          5977          RDPTR,
10431 ;          5978          WCPTR,
10432 ;          5979          RCPTR;
10433 ;          5980
10434 ;          5981      PRINTB (SAY2, WRD34, RTN4);
10435 ;          5982      !:RUNNING OPT4'
10436 ;          5983      GEN4 (.MARPAT, WDBUFF);
10437 ;          5984      GEN4 ((-.MARPAT), WCBUFF);
10438 ;          5985      WDPTR = WDBUFF;                 !POINTER TO 256-WORD WRITE DATA BUFFER
10439 ;          5986      WCPTR = WCBUFF;                 !POINTER TO 256-WORD WRITE COMP BUFFER
10440 ;          5987      RDPTR = RDBUFF;                 !POINTER TO 256-WORD READ  DATA BUFFER
10441 ;          5988      RCPTR = RCBUFF;                 !POINTER TO 256-WORD READ  COMP BUFFER
10442 ;          5989
10443 ;          5990      incr LUN from 0 to (.L$UNIT - 1) do
10444 ;          5991          begin
10445 ;          5992  LOOP :                             !* 2 *  START OF LOGICAL UNIT SELECTION LOOP
10446 ;          5993          begin                       !* 3 *  START OF THE LOOP THAT COMPLETELY TESTS 1 UNIT
10447 ;          5994
10448 ;          5995          if .DRIVE_STATUS [.LUN] eql ACTIVE
10449 ;          5996          then
10450 ;          5997              begin
10451 ;          5998              L$LUN = .LUN;           !* 4 *  START OF TEST FOR AN ACTIVE UNIT
10452 ;          5999              PRINTB (SAY1, WRD35);
10453 ;          6000              !'MARCHING'
10454 ;          6001              PRINTB (FMT2, WRD24);
10455 ;          6002              !'  UP'
10456 ;          6003
10457 ;          6004 !
10458 ;          6005 ! version czmlbb changed incr to incru
10459 ;          6006 !
10460 ;          6007              incru SECTOR from LOWEST to HIGHEST do
10461 ;          6008                  begin               !* 4A *  START OF 1ST SECTOR SELECTION LOOP
10462 ;          6009                  VALUE = write (.LUN, 256, .WDPTR, .SECTOR);
10463 ;          6010
```

```
;MLX4
;                DEFINITION OF OPTION 4
;
;       6011                    !+
;       6012                    !  SEE HOW SUCCESSFUL THE WRITE WAS:
;       6013                    !-
;       6014
;       6015                    selectone .VALUE of             !SEE 'SYSERR' FOR DEFINITION
;       6016                        set                         !OF ERROR # CONTAINED IN 'VALUE'
;       6017
;       6018                        [1] :
;       6019                            begin                   !* 4A1 *        RETRY ALLOWED
;       6020
;       6021                            if RETRY (SIX, write, .LUN, 256, .WDPTR, .SECTOR) neq 0
;       6022                            then                    !THE RETRY FAILED -- SYSTEM FATAL ERROR
;       6023                                begin
;       6024                                WHY_DROPT [.LUN] = CODE_4;
;       6025                                ERRDF (401, MSG1, 0);       !**** OPTION 4 ERROR 01 ****
;       6026                                DODU (.LUN);
;       6027                                leave LOOP;         !JUMP JUST BEYOND END OF BLOCK * 3 *
;       6028                                end;
;       6029
;       6030                            end;                    !* 4A1 *
;       6031
;       6032                        [2] :
;       6033                            begin                   !* 4A2 *        FATAL CONTROLLER ERROR -- NO RETRY ALLOWED
;       6034                            WHY_DROPT [.LUN] = CODE_5;
;       6035                            ERRDF (402, MSG1, 0);   !**** OPTION 4 ERROR 02 ****
;       6036                            DODU (.LUN);
;       6037                            leave LOOP;             !JUMP JUST BEYOND END OF BLOCK * 3 *
;       6038                            end;                    !* 4A2 *
;       6039
;       6040                        [3] :
;       6041                            begin                   !* 4A3 *        FATAL DRIVE ERROR -- NO RETRY ALLOWED
;       6042                            ERRDF (403, MSG1, 0);   !**** OPTION 4 ERROR 03 ****
;       6043                            WHY_DROPT [.LUN] = CODE_6;
;       6044                            DODU (.LUN);
;       6045                            leave LOOP;             !JUMP JUST BEYOND END OF BLOCK * 3 *
;       6046                            end;                    !* 4A3 *
;       6047                        tes;
;       6048
;       6049                    end;                            !* 4A *  END OF 1ST SECTOR SELECTION LOOP
;       6050
;       6051                PRINTB (FMT2, WRD24);
;       6052                !' UP'
;       6053                COMMAND = CHOOSE ();
;       6054
;       6055    !
;       6056    ! version czmlbb changed incr to incru
;       6057    !
;       6058                incru SECTOR from LOWEST to HIGHEST do
;       6059                    begin                           !* 4B *  START OF 2ND SECTOR SELECTION LOOP
;       6060
;       6061                    if .COMMAND eql read
;       6062                    then
```

```
10521 ;MLX4
10522 ;                 DEFINITION OF OPTION 4
10523 ;
10524 ;          6063                       begin
10525 ;          6064                       PTR = .RDPTR;
10526 ;          6065                       VALUE = read (.LUN, 256, .PTR, .SECTOR);
10527 ;          6066                       end
10528 ;          6067                   else
10529 ;          6068                       begin
10530 ;          6069                       PTR = .WDPTR;
10531 ;          6070                       VALUE = CHECK (.LUN, 256, .PTR, .SECTOR);
10532 ;          6071                       end;
10533 ;          6072
10534 ;          6073                   !+
10535 ;          6074                   ! SEE HOW SUCCESSFUL THE OPERATION WAS:
10536 ;          6075                   !-
10537 ;          6076
10538 ;          6077                   selectone .VALUE of              !SEE 'SYSERR' FOR DEFINITION
10539 ;          6078                       set                          !OF ERROR # CONTAINED IN 'VALUE'
10540 ;          6079
10541 ;          6080                       [0] :
10542 ;          6081
10543 ;          6082                           if .COMMAND eql read
10544 ;          6083                           then
10545 ;          6084                               begin
10546 ;          6085
10547 ;          6086                               if (DBL_VALUE = DOUBLE_CHECK (.WDPTR, .RDPTR, 256)) neq 0
10548 ;          6087                               then
10549 ;          6088                                   begin
10550 ;          6089                                   SAYWHO (.LUN);
10551 ;          6090                                   PRINTB (SAY1, MSG5);
10552 ;          6091                                   !'ECC LOGIC FAILED TO DETECT DATA ERROR'
10553 ;          6092                                   PRINTB (FMT12A, ..DBL_VALUE, .DBL_VALUE);
10554 ;          6093                                   !'GOOD DATA: XXXXXX AT LOCATION YYYYYY'
10555 ;          6094                                   DBL_VALUE = .DBL_VALUE + BUFSIZ*2;
10556 ;          6095                                   PRINTB (FMT12B, ..DBL_VALUE, .DBL_VALUE);
10557 ;          6096                                   !'BAD DATA:  PPPPPP AT LOCATION QQQQQQ'
10558 ;          6097                                   WHY_DROPT [.LUN] = CODE_8;
10559 ;          6098                                   ERRDF (404, MSG1, 0);   !**** OPTION 4 ERROR 04 ****
10560 ;          6099                                   DODU (.LUN);
10561 ;          6100                                   leave LOOP;     !JUMP JUST BEYOND END OF BLOCK * 4 *
10562 ;          6101                                   end;
10563 ;          6102
10564 ;          6103                               end;
10565 ;          6104
10566 ;          6105                       [1] :
10567 ;          6106                           begin                    !* 4B1 *        RETRY ALLOWED
10568 ;          6107
10569 ;          6108                           if RETRY (SIX, .COMMAND, .LUN, 256, .PTR, .SECTOR) neq 0
10570 ;          6109                           then                     !THE RETRY FAILED -- SYSTEM FATAL ERROR
10571 ;          6110                               begin
10572 ;          6111                               WHY_DROPT [.LUN] = CODE_4;
10573 ;          6112                               ERRDF (405, MSG1, 0);       !**** OPTION 4 ERROR 05 ****
10574 ;          6113                               DODU (.LUN);
10575 ;          6114                               leave LOOP;          !JUMP JUST BEYOND END OF BLOCK * 3 *
```

```
;MLX4
;                    DEFINITION OF OPTION 4                          27-Mar-1982 19:24:42    TOPS-20 Bliss-16 V2(212)
                                                                     27-Mar-1982 19:23:44    PA:<NEALE>MLX4.BLI.5 (40)
;          6115                                 end;
;          6116
;          6117                             end;                    !* 4B1 *
;          6118
;          6119                        [2] :
;          6120                             begin                   !* 4B2 *        FATAL CONTROLLER ERROR -- NO RETRY ALLOWED
;          6121                             WHY_DROPT [.LUN] = CODE_5;
;          6122                             ERRDF (406, MSG1, 0);   !**** OPTION 4 ERROR 06 ****
;          6123                             DODU (.LUN);
;          6124                             leave LOOP;             !JUMP JUST BEYOND END OF BLOCK * 3 *
;          6125                             end;                    !* 4B2 *
;          6126
;          6127                        [3] :
;          6128                             begin                   !* 4B3 *        FATAL DRIVE ERROR -- NO RETRY ALLOWED
;          6129                             ERRDF (407, MSG1, 0);   !**** OPTION 4 ERROR 07 ****
;          6130                             WHY_DROPT [.LUN] = CODE_6;
;          6131                             DODU (.LUN);
;          6132                             leave LOOP;             !JUMP JUST BEYOND END OF BLOCK * 3 *
;          6133                             end;                    !* 4B3 *
;          6134
;          6135                        [4] :
;          6136                             begin                   !* 4B4 *        UNRECOVERABLE DATA ERROR
;          6137                             ISOLATE ();
;          6138                             ERRDF (408, MSG2, 0);   !**** OPTION 4 ERROR 08 ****
;          6139                             WHY_DROPT [.LUN] = CODE_7;
;          6140                             DODU (.LUN);
;          6141                             leave LOOP;             !JUMP JUST BEYOND END OF BLOCK * 3 *
;          6142                             end;                    !* 4B4 *
;          6143
;          6144                        [5] :
;          6145                             begin                   !* 4B5 *        RECOVERABLE DATA ERROR
;          6146                             ISOLATE ();
;          6147
;          6148                             if .ERROUT then PRINTB (FMT10B, .CHAN);
;          6149
;          6150                             !'    BIT QQ'
;          6151                             OLDSEC = .MLEL;
;          6152                             OLDCHN = .CHAN;
;          6153
;          6154                             if RETRY (ONE, .COMMAND, .LUN, 256, .PTR, .SECTOR) eql 5
;          6155                             then
;          6156
;          6157                                 if ((.MLEL eql .OLDSEC) and (.CHAN eql .OLDCHN))
;          6158                                 then
;          6159                                     begin
;          6160
;          6161                                     if .ERROUT then ERRHRD (409, MSG4, 0);  !**** OPTION 4 ERROR 09 ****
;          6162
;          6163                                     UP_HARD_COUNT (.LUN, .BOARD);
;          6164                                     end
;          6165                                 else
;          6166                                     begin
```

```
10633 ;MLX4
10634 ;
10635
10636 ;          6167
10637 ;          6168                                    if .ERROUT then ERRSOFT (410, MSG3, 0); !**** OPTION 4 ERROR 10 ****
10638 ;          6169
10639 ;          6170                                    UP_SOFT_COUNT (.LUN, .BOARD);
10640 ;          6171                                    end
10641 ;          6172
10642 ;          6173                                ELSE
10643 ;          6174                                    begin
10644 ;          6175
10645 ;          6176                                    if .ERROUT then ERRSOFT (411, MSG3, 0);     !**** OPTION 4 ERROR 11 ****
10646 ;          6177
10647 ;          6178                                    UP_SOFT_COUNT (.LUN, .BOARD);
10648 ;          6179                                    end;
10649 ;          6180
10650 ;          6181                                end;                        !* 4B5 *
10651 ;          6182                            tes;
10652 ;          6183
10653 ;          6184                        VALUE = write (.LUN, 256, .WCPTR, .SECTOR);
10654 ;          6185
10655 ;          6186                        !+
10656 ;          6187                        !  SEE HOW SUCCESSFUL THE WRITE WAS:
10657 ;          6188                        !-
10658 ;          6189
10659 ;          6190                        selectone .VALUE of             !SEE 'SYSERR' FOR DEFINITION
10660 ;          6191                            set                         !OF ERROR # CONTAINED IN 'VALUE'
10661 ;          6192
10662 ;          6193                            [1] :
10663 ;          6194                                begin                   !* 4B6 *         RETRY ALLOWED
10664 ;          6195
10665 ;          6196                            if RETRY (SIX, write, .LUN, 256, .WCPTR, .SECTOR) neg 0
10666 ;          6197                            then                        !THE RETRY FAILED -- SYSTEM FATAL ERROR
10667 ;          6198                                begin
10668 ;          6199                                WHY_DROPT [.LUN] = CODE_4;
10669 ;          6200                                ERRDF (412, MSG1, 0);       !**** OPTION 4 ERROR 12 ****
10670 ;          6201                                DODU (.LUN);
10671 ;          6202                                leave LOOP;             !JUMP JUST BEYOND END OF BLOCK * 3 *
10672 ;          6203                                end;
10673 ;          6204
10674 ;          6205                            end;                        !* 4B6 *
10675 ;          6206
10676 ;          6207                            [2] :
10677 ;          6208                                begin                   !* 4B7 *         FATAL CONTROLLER ERROR -- NO RETRY ALLOWED
10678 ;          6209                                WHY_DROPT [.LUN] = CODE_5;
10679 ;          6210                                ERRDF (413, MSG1, 0);   !**** OPTION 4 ERROR 13 ****
10680 ;          6211                                DODU (.LUN);
10681 ;          6212                                leave LOOP;             !JUMP JUST BEYOND END OF BLOCK * 3 *
10682 ;          6213                                end;                    !* 4B7 *
10683 ;          6214
10684 ;          6215                            [3] :
10685 ;          6216                                begin                   !* 4B8 *         FATAL DRIVE ERROR -- NO RETRY ALLOWED
10686 ;          6217                                ERRDF (414, MSG1, 0);   !**** OPTION 4 ERROR 14 ****
10687 ;          6218                                WHY_DROPT [.LUN] = CODE_6;
```

SEQ 0246


```
10689 ;MLX4
10690 ;                 DEFINITION OF OPTION 4
10691
10692 ;          6219                           DODU (.LUN);
10693 ;          6220                           leave LOOP;                !JUMP JUST BEYOND END OF BLOCK * 3 *
10694 ;          6221                           end;                       !* 488 *
10695 ;          6222                      tes;
10696 ;          6223
10697 ;          6224                  end;                               !* 48 *  END OF 2ND SECTOR SELECTION LOOP
10698 ;          6225
10699 ;          6226              PRINTB (FMT2, WRD25);
10700 ;          6227              !'  DOWN'
10701 ;          6228              COMMAND = CHOOSE ();
10702 ;          6229
10703 ;          6230              decr SECTOR from HIGHEST to LOWEST do
10704 ;          6231                  begin                                  !* 4C *  START OF 3RD SECTOR SELECTION LOOP
10705 ;          6232
10706 ;          6233                  if .COMMAND eql read
10707 ;          6234                  then
10708 ;          6235                      begin
10709 ;          6236                      PTR = .RCPTR;
10710 ;          6237                      VALUE = read (.LUN, 256, .PTR, .SECTOR);
10711 ;          6238                      end
10712 ;          6239                  else
10713 ;          6240                      begin
10714 ;          6241                      PTR = .WCPTR;
10715 ;          6242                      VALUE = CHECK (.LUN, 256, .PTR, .SECTOR);
10716 ;          6243                      end;
10717 ;          6244
10718 ;          6245                  !+
10719 ;          6246                  !  SEE HOW SUCCESSFUL THE OPERATION WAS:
10720 ;          6247                  !-
10721 ;          6248
10722 ;          6249              selectone .VALUE of                    !SEE 'SYSERR' FOR DEFINITION
10723 ;          6250                  set                                !OF ERROR # CONTAINED IN 'VALUE'
10724 ;          6251
10725 ;          6252                  [0] :
10726 ;          6253
10727 ;          6254                      if .COMMAND eql read
10728 ;          6255                      then
10729 ;          6256                          begin
10730 ;          6257
10731 ;          6258                          if (DBL_VALUE = DOUBLE_CHECK (.WCPTR, .RCPTR, 256)) neq 0
10732 ;          6259                          then
10733 ;          6260                              begin
10734 ;          6261                              SAYWHO (.LUN);
10735 ;          6262                              PRINTB (SAY1, MSG5);
10736 ;          6263                              !'ECC LOGIC FAILED TO DETECT DATA ERROR'
10737 ;          6264                              PRINTB (FMT12A, ..DBL_VALUE, .DBL_VALUE);
10738 ;          6265                              !'GOOD DATA: XXXXXX AT LOCATION YYYYYY'
10739 ;          6266                              DBL_VALUE = .DBL_VALUE + BUFSIZ*2;
10740 ;          6267                              PRINTB (FMT12B, ..DBL_VALUE, .DBL_VALUE);
10741 ;          6268                              !'BAD DATA:  PPPPPP AT LOCATION QQQQQQ'
10742 ;          6269                              WHY_DROPT [.LUN] = CODE_8;
10743 ;          6270                              ERRDF (415, MSG1, 0);   !**** OPTION 4 ERROR 15 ****
```

```
10745 ;MLX4
10746 ;                 DEFINITION OF OPTION 4
10747
10748 ;          6271                                   DODU (.LUN);
10749 ;          6272                                   leave LOOP·      !JUMP JUST BEYOND END OF BLOCK * 4 *
10750 ;          6273                                   end;
10751 ;          6274
10752 ;          6275                               end;
10753 ;          6276
10754 ;          6277                           [1] :
10755 ;          6278                               begin                !* 4C1 *        RETRY ALLOWED
10756 ;          6279
10757 ;          6280                               if RETRY (SIX, .COMMAND, .LUN, 256, .PTR, .SECTOR) neq 0
10758 ;          6281                               then                 !THE RETRY FAILED -- SYSTEM FATAL ERROR
10759 ;          6282                                   begin
10760 ;          6283                                   WHY_DROPT [.LUN] = CODE_4;
10761 ;          6284                                   ERRDF (416, MSG1, 0);       !**** OPTION 4 ERROR 16 ****
10762 ;          6285                                   DODU (.LUN);
10763 ;          6286                                   leave LOOP;          !JUMP JUST BEYOND END OF BLOCK * 3 *
10764 ;          6287                                   end;
10765 ;          6288
10766 ;          6289                               end;                 !* 4C1 *
10767 ;          6290
10768 ;          6291                           [2] :
10769 ;          6292                               begin                !* 4C2 *        FATAL CONTROLLER ERROR -- NO RETRY ALLOWED
10770 ;          6293                               WHY_DROPT [.LUN] = CODE_5;
10771 ;          6294                               ERRDF (417, MSG1, 0);   !**** OPTION 4 ERROR 17 ****
10772 ;          6295                               DODU (.LUN);
10773 ;          6296                               leave LOOP;          !JUMP JUST BEYOND END OF BLOCK * 3 *
10774 ;          6297                               end;                 !* 4C2 *
10775 ;          6298
10776 ;          6299                           [3] :
10777 ;          6300                               begin                !* 4C3 *        FATAL DRIVE ERROR -- NO RETRY ALLOWED
10778 ;          6301                               ERRDF (418, MSG1, 0);   !**** OPTION 4 ERROR 18 ****
10779 ;          6302                               WHY_DROPT [.LUN] = CODE_6;
10780 ;          6303                               DODU (.LUN);
10781 ;          6304                               leave LOOP;          !JUMP JUST BEYOND END OF BLOCK * 3 *
10782 ;          6305                               end;                 !* 4C3 *
10783 ;          6306
10784 ;          6307                           [4] :
10785 ;          6308                               begin                !* 4C4 *        UNRECOVERABLE DATA ERROR
10786 ;          6309                               ISOLATE ();
10787 ;          6310                               ERRDF (419, MSG2, 0);   !**** OPTION 4 ERROR 19 ****
10788 ;          6311                               WHY_DROPT [.LUN] = CODE_7;
10789 ;          6312                               DODU (.LUN);
10790 ;          6313                               leave LOOP;          !JUMP JUST BEYOND END OF BLOCK * 3 *
10791 ;          6314                               end;                 !* 4C4 *
10792 ;          6315
10793 ;          6316                           [5] :
10794 ;          6317                               begin                !* 4C5 *        RECOVERABLE DATA ERROR
10795 ;          6318                               ISOLATE ();
10796 ;          6319
10797 ;          6320                               if .ERROUT then PRINTB (FMT10B, .CHAN);
10798 ;          6321
10799 ;          6322                               !'    BIT 00'
```

```
10801 ;MLX4
10802 ;                 DEFINITION OF OPTION 4
10803
10804 ;          6323                           OLDSEC = .MLEL;
10805 ;          6324                           OLDCHN = .CHAN;
10806 ;          6325
10807 ;          6326                           if RETRY (ONE, .COMMAND, .LUN, 256, .PTR, .SECTOR) eql 5
10808 ;          6327                           then
10809 ;          6328
10810 ;          6329                               if ((.MLEL eql .OLDSEC) and (.CHAN eql .OLDCHN))
10811 ;          6330                               then
10812 ;          6331                                   begin
10813 ;          6332
10814 ;          6333                                   if .ERROUT then ERRHRD (420, MSG4, 0);  !**** OPTION 4 ERROR 20 ****
10815 ;          6334
10816 ;          6335                                   UP_HARD_COUNT (.LUN, .BOARD);
10817 ;          6336                                   end
10818 ;          6337                               else
10819 ;          6338                                   begin
10820 ;          6339
10821 ;          6340                                   if .ERROUT then ERRSOFT (421, MSG3, 0); !**** OPTION 4 ERROR 21 ****
10822 ;          6341
10823 ;          6342                                   UP_SOFT_COUNT (.LUN, .BOARD);
10824 ;          6343                                   end
10825 ;          6344
10826 ;          6345                           else
10827 ;          6346                               begin
10828 ;          6347
10829 ;          6348                               if .ERROUT then ERRSOFT (422, MSG3, 0);     !**** OPTION 4 ERROR 22 ****
10830 ;          6349
10831 ;          6350                               UP_SOFT_COUNT (.LUN, .BOARD);
10832 ;          6351                               end;
10833 ;          6352
10834 ;          6353                           end;                        !* 4C5 *
10835 ;          6354                       tes;
10836 ;          6355
10837 ;          6356                   VALUE = write (.LUN, 256, .WDPTR, .SECTOR);
10838 ;          6357
10839 ;          6358                   !+
10840 ;          6359                   ! SEE HOW SUCCESSFUL THE WRITE WAS:
10841 ;          6360                   !-
10842 ;          6361
10843 ;          6362                   selectone .VALUE of                 !SEE 'SYSERR' FOR DEFINITION
10844 ;          6363                       set                             !OF ERROR # CONTAINED IN 'VALUE'
10845 ;          6364
10846 ;          6365                       [1] :
10847 ;          6366                           begin                       !* 4C6 *        RETRY ALLOWED
10848 ;          6367
10849 ;          6368                           if RETRY (SIX, write, .LUN, 256, .WDPTR, .SECTOR) neg 0
10850 ;          6369                           then                        !THE RETRY FAILED -- SYSTEM FATAL ERROR
10851 ;          6370                               begin
10852 ;          6371                               WHY_DROPT [.LUN] = CODE_4;
10853 ;          6372                               ERRDF (423, MSG1, 0);       !**** OPTION 4 ERROR 23 ****
10854 ;          6373                               DODU (.LUN);
10855 ;          6374                               leave LOOP;             !JUMP JUST BEYOND END OF BLOCK * 3 *
```

```
10857 ;MLX4
10858 ;                 DEFINITION OF OPTION 4
10859 ;
10860 ;          6375                                   end;
10861 ;          6376
10862 ;          6377                               end;                    !* 4C6 *
10863 ;          6378
10864 ;          6379                           [2] :
10865 ;          6380                               begin                   !* 4C7 *        FATAL CONTROLLER ERROR -- NO RETRY ALLOWED
10866 ;          6381                               WHY_DROPT [.LUN] = CODE_5;
10867 ;          6382                               ERRDF (424, MSG1, 0);   !**** OPTION 4 ERROR 24 ****
10868 ;          6383                               DODU (.LUN);
10869 ;          6384                               leave LOOP;             !JUMP JUST BEYOND END OF BLOCK * 3 *
10870 ;          6385                               end;                    !* 4C7 *
10871 ;          6386
10872 ;          6387                           [3] :
10873 ;          6388                               begin                   !* 4C8 *        FATAL DRIVE ERROR -- NO RETRY ALLOWED
10874 ;          6389                               ERRDF (425, MSG1, 0);   !**** OPTION 4 ERROR 25 ****
10875 ;          6390                               WHY_DROPT [.LUN] = CODE_6;
10876 ;          6391                               DODU (.LUN);
10877 ;          6392                               leave LOOP;             !JUMP JUST BEYOND END OF BLOCK * 3 *
10878 ;          6393                               end;                    !* 4C8 *
10879 ;          6394                           tes;
10880 ;          6395
10881 ;          6396                       end;                            !* 4C *  END OF 3RD SECTOR SELECTION LOOP
10882 ;          6397
10883 ;          6398                   PRINTB (FMT2, WRD24);
10884 ;          6399                   !' UP'
10885 ;          6400                   COMMAND = CHOOSE ();
10886 ;          6401
10887 ;          6402   !
10888 ;          6403   ! version czmlbb changed incr to incru
10889 ;          6404   !
10890 ;          6405                   incru SECTOR from LOWEST to HIGHEST do
10891 ;          6406                       begin                           !* 4D *  START OF 4TH SECTOR SELECTION LOOP
10892 ;          6407
10893 ;          6408                       if .COMMAND eql read
10894 ;          6409                       then
10895 ;          6410                           begin
10896 ;          6411                           PTR = .RDPTR;
10897 ;          6412                           VALUE = read (.LUN, 256, .PTR, .SECTOR);
10898 ;          6413                           end
10899 ;          6414                       else
10900 ;          6415                           begin
10901 ;          6416                           PTR = .WDPTR;
10902 ;          6417                           VALUE = CHECK (.LUN, 256, .PTR, .SECTOR);
10903 ;          6418                           end;
10904 ;          6419
10905 ;          6420                       !+
10906 ;          6421                       ! SEE HOW SUCCESSFUL THE OPERATION WAS:
10907 ;          6422                       !-
10908 ;          6423
10909 ;          6424                       selectone .VALUE of             !SEE 'SYSERR' FOR DEFINITION
10910 ;          6425                           set                         !OF ERROR # CONTAINED IN 'VALUE'
10911 ;          6426
```

```
10913 ;MLX4
10914 ;
10915 ;                 DEFINITION OF OPTION 4
10916 ;          6427
10917 ;          6428                          [0] :
10918 ;          6429                              if .COMMAND eql read
10919 ;          6430                              then
10920 ;          6431                                  begin
10921 ;          6432
10922 ;          6433                                  if (DBL_VALUE = DOUBLE_CHECK (.WDPTR, .RDPTR, 256)) neq 0
10923 ;          6434                                  then
10924 ;          6435                                      begin
10925 ;          6436                                      SAYWHO (.LUN);
10926 ;          6437                                      PRINTB (SAY1, MSG5);
10927 ;          6438                                      !'ECC LOGIC FAILED TO DETECT DATA ERROR'
10928 ;          6439                                      PRINTB (FMT12A, ..DBL_VALUE, .DBL_VALUE);
10929 ;          6440                                      !'GOOD DATA: XXXXXX AT LOCATION YYYYYY'
10930 ;          6441                                      DBL_VALUE = .DBL_VALUE + BUFSIZ*2;
10931 ;          6442                                      PRINTB (FMT12B, ..DBL_VALUE, .DBL_VALUE);
10932 ;          6443                                      !'BAD DATA:  PPPPPP AT LOCATION QQQQQQ'
10933 ;          6444                                      WHY_DROPT [.LUN] = CODE_8;
10934 ;          6445                                      ERRDF (426, MSG1, 0);   !**** OPTION 4 ERROR 26 ****
10935 ;          6446                                      DODU (.LUN);
10936 ;          6447                                      leave LOOP;     !JUMP JUST BEYOND END OF BLOCK * 4 *
10937 ;          6448                                      end;
10938 ;          6449
10939 ;          6450                                  end;
10940 ;          6451
10941 ;          6452                          [1] :
10942 ;          6453                              begin                   !* 4D1 *        RETRY ALLOWED
10943 ;          6454
10944 ;          6455                              if RETRY (SIX, .COMMAND, .LUN, 256, .PTR, .SECTOR) neq 0
10945 ;          6456                              then                    !THE RETRY FAILED -- SYSTEM FATAL ERROR
10946 ;          6457                                  begin
10947 ;          6458                                  WHY_DROPT [.LUN] = CODE_4;
10948 ;          6459                                  ERRDF (427, MSG1, 0);       !**** OPTION 4 FRROR 27 ****
10949 ;          6460                                  DODU (.LUN);
10950 ;          6461                                  leave LOOP;         !JUMP JUST BEYOND END OF BLOCK * 3 *
10951 ;          6462                                  end;
10952 ;          6463
10953 ;          6464                              end;                    !* 4D1 *
10954 ;          6465
10955 ;          6466                          [2] :
10956 ;          6467                              begin                   !* 4D2 *        FATAL CONTROLLER ERROR -- NO RETRY ALLOWED
10957 ;          6468                              WHY_DROPT [.LUN] = CODE_5;
10958 ;          6469                              ERRDF (428, MSG1, 0);   !**** OPTION 4 ERROR 28 ****
10959 ;          6470                              DODU (.LUN);
10960 ;          6471                              leave LOOP;             !JUMP JUST BEYOND END OF BLOCK * 3 *
10961 ;          6472                              end;                    !* 4D2 *
10962 ;          6473
10963 ;          6474                          [3] :
10964 ;          6475                              begin                   !* 4D3 *        FATAL DRIVE ERROR -- NO RETRY ALLOWED
10965 ;          6476                              ERRDF (429, MSG1, 0);   !**** OPTION 4 ERROR 29 ****
10966 ;          6477                              WHY_DROPT [.LUN] = CODE_6;
10967 ;          6478                              DODU (.LUN);
```

```
10969 ;MLX4
10970 ;                 DEFINITION OF OPTION 4
10971
10972 ;          6479                           leave LOOP;                 !JUMP JUST BEYOND END OF BLOCK * 3 *
10973 ;          6480                           end;                        !* 4D3 *
10974 ;          6481
10975 ;          6482                       [4] :
10976 ;          6483                           begin                       !* 4D4 *        UNRECOVERABLE DATA ERROR
10977 ;          6484                           ISOLATE ();
10978 ;          6485                           ERRDF (430, MSG2, 0);       !**** OPTION 4 ERROR 30 ****
10979 ;          6486                           WHY_DROPT [.LUN] = CODE_7;
10980 ;          6487                           DODO (.LUN);
10981 ;          6488                           leave LOOP;                 !JUMP JUST BEYOND END OF BLOCK * 3 *
10982 ;          6489                           end;                        !* 4D4 *
10983 ;          6490
10984 ;          6491                       [5] :
10985 ;          6492                           begin                       !* 4D5 *        RECOVERABLE DATA ERROR
10986 ;          6493                           ISOLATE ();
10987 ;          6494
10988 ;          6495                           if .ERROUT then PRINTB (FMT10B, .CHAN);
10989 ;          6496
10990 ;          6497                           !'    BIT QQ'
10991 ;          6498                           OLDSEC = .MLEL;
10992 ;          6499                           OLDCHN = .CHAN;
10993 ;          6500
10994 ;          6501                           if RETRY (ONE, .COMMAND, .LUN, 256, .PTR, .SECTOR) eql 5
10995 ;          6502                           then
10996 ;          6503
10997 ;          6504                               if ((.MLEL eql .OLDSEC) and (.CHAN eql .OLDCHN))
10998 ;          6505                               then
10999 ;          6506                                   begin
11000 ;          6507
11001 ;          6508                                   if .ERROUT then ERRHRD (431, MSG4, 0);  !**** OPTION 4 ERROR 31 ****
11002 ;          6509
11003 ;          6510                                   UP_HARD_COUNT (.LUN, .BOARD);
11004 ;          6511                                   end
11005 ;          6512                               else
11006 ;          6513                                   begin
11007 ;          6514
11008 ;          6515                                   if .ERROUT then ERRSOFT (432, MSG3, 0); !**** OPTION 4 ERROR 32 ****
11009 ;          6516
11010 ;          6517                                   UP_SOFT_COUNT (.LUN, .BOARD);
11011 ;          6518                                   end
11012 ;          6519
11013 ;          6520                           else
11014 ;          6521                               begin
11015 ;          6522
11016 ;          6523                               if .ERROUT then ERRSOFT (433, MSG3, 0);     !**** OPTION 4 ERROR 33 ****
11017 ;          6524
11018 ;          6525                               UP_SOFT_COUNT (.LUN, .BOARD);
11019 ;          6526                               end;
11020 ;          6527
11021 ;          6528                           end;                        !* 4D5 *
11022 ;          6529                   tes;
11023 ;          6530
```

```
11025 ;MLX4
11026 ;                 DEFINITION OF OPTION 4
11027 ;
11028 ;          6531                   end;                          !* 4D *  END OF 4TH SECTOR SELECTION LOOP
11029 ;          6532
11030 ;          6533               end;                              !* 4 *  END OF TEST FOR AN ACTIVE UNIT
11031 ;          6534
11032 ;          6535           !+
11033 ;          6536           ! Test to see if this uut's address space is
11034 ;          6537           ! to be read for soft errors.  This test is
11035 ;          6538           ! is intended for DMT purposes.
11036 ;          6539           !-
11037 ;          6540
11038 ;          6541           if .EFN$21                            !Is the background pattern to be read
11039 ;          6542           then
11040 ;          6543               begin
11041 ;          6544
11042 ;          6545   !
11043 ;          6546   ! version czmlbb changed incr to incru
11044 ;          6547   !
11045 ;          6548               incru SECTOR from LOWEST to HIGHEST do
11046 ;          6549
11047 ;          6550                   if read (.LUN, 256, RBUFF, .SECTOR) eql 5
11048 ;          6551                   then
11049 ;          6552                       begin
11050 ;          6553                       ISOLATE ();               !Find the failing bank and board no.
11051 ;          6554
11052 ;          6555                       if .ERROUT then PRINTB (FMT10B, .CHAN);
11053 ;          6556
11054 ;          6557                       ! Print where the error is
11055 ;          6558                       !
11056 ;          6559                       ! Save the contents of the ML error location
11057 ;          6560                       ! register so we can compare it to the new
11058 ;          6561                       ! contents of this register after the retry.
11059 ;          6562                       ! This is done to classify the error.
11060 ;          6563                       !
11061 ;          6564                       OLDSEC = .MLEL;
11062 ;          6565                       OLDCHN = .CHAN;
11063 ;          6566                       !
11064 ;          6567                       ! Do a classify retry call.  If the same error
11065 ;          6568                       ! occures then classify it as a hard error.  If
11066 ;          6569                       ! a different error occured or the error went away
11067 ;          6570                       ! then classify it as a soft error.
11068 ;          6571                       !
11069 ;          6572
11070 ;          6573                       if RETRY (ONE, .COMMAND, .LUN, 256, .PTR, .OLDSEC) eql 5
11071 ;          6574                       then
11072 ;          6575                       !
11073 ;          6576                       ! The same error occured so see if it is at the same
11074 ;          6577                       ! sector and channel number, if so then classify
11075 ;          6578                       ! it as a hard error else classiy it as a soft
11076 ;          6579                       ! error.
11077 ;          6580                       !
11078 ;          6581
11079 ;          6582                           if ((.MLEL eql .OLDSEC) and (.CHAN eql .OLDCHN))
```

SEQ 0253


```
11081 ;MLX4
11082 ;                 DEFINITION OF OPTION 4
11083
11084 ;          6583                               then
11085 ;          6584                                   begin                   !Same error occured 'hard'
11086 ;          6585
11087 ;          6586                                   if .ERROUT              !Print error if enabled
11088 ;          6587                                   then
11089 ;          6588                                       begin
11090 ;        P 6589                                       ERRHRD (434,        !Error number
11091 ;        P 6590                                           MSG4,           !Error message
11092 ;          6591                                           0);             !Additional message routine
11093 ;          6592                                       end;
11094 ;          6593
11095 ;          6594                                   UP_HARD_COUNT (.LUN, .BOARD);
11096 ;          6595                                   end
11097 ;          6596                               else
11098 ;          6597                                   begin                   !Not the same error 'soft'
11099 ;          6598
11100 ;          6599                                   if .ERROUT              !Print error if enabled
11101 ;          6600                                   then
11102 ;          6601                                       begin
11103 ;        P 6602                                       ERRSOFT (435,       !Error number
11104 ;        P 6603                                           MSG3,           !Error message
11105 ;          6604                                           0);             !Additional message routine
11106 ;          6605                                       end;
11107 ;          6606
11108 ;          6607                                   UP_SOFT_COUNT (.LUN, .BOARD);
11109 ;          6608                                   end
11110 ;          6609
11111 ;          6610                           else                            !Not the same error 'soft'
11112 ;          6611                               begin
11113 ;          6612
11114 ;          6613                               if .ERROUT                  !Print error if enabled
11115 ;          6614                               then
11116 ;          6615                                   begin
11117 ;        P 6616                                   ERRSOFT (436,           !Error number
11118 ;        P 6617                                       MSG3,               !Error message
11119 ;          6618                                       0);                 !Additional message routine
11120 ;          6619                                   end;
11121 ;          6620
11122 ;          6621                               UP_SOFT_COUNT (.LUN, .BOARD);
11123 ;          6622                               end;
11124 ;          6623
11125 ;          6624                           end;
11126 ;          6625
11127 ;          6626                   end;
11128 ;          6627
11129 ;          6628               end;                                        !* 3 *  END OF LOOP THAT COMPLETELY TESTS 1 UNIT
11130 ;          6629               end;                                        !* 2 *  END OF LOGICAL UNIT SELECTION LOOP
11131 ;          6630
11132 ;          6631       return;
11133 ;          6632       end;                                                !* 1 *  END OF ROUTINE
```

```
11141
11142
11146 062444  004167  122664          OPT4:   .SBTTL  OPT4 DEFINITION OF OPTION 4
11147 062450  162706  000030                  JSR     R1,$SAVE5                       ;                       5908
11148 062454  012746  007650                  SUB     #30,SP
11149 062460  012746  007344                  MOV     #RTN4,-(SP)                     ;                       5981
11150 062464  012746  007100                  MOV     #WRD34,-(SP)
11151 062470  012746  000003                  MOV     #SAY2,-(SP)
11152 062474  010600                          MOV     #3,-(SP)
11153 062476  104414                          MOV     SP,R0                           ; SP,*
11154 062500  016716  117542                  TRAP    14
11155 062504  012746  012670                  MOV     MARPAT,(SP)
11156 062510  004767  157050                  MOV     #WDBUFF,-(SP)                   ;                       5983
11157 062514  016716  117526                  JSR     PC,GEN4
11158 062520  005416                          MOV     MARPAT,(SP)
11159 062522  012746  013670                  NEG     (SP)                            ;                       5984
11160 062526  004767  157032                  MOV     #WCBUFF,-(SP)
11161 062532  012766  012670  000016          JSR     PC,GEN4
11162 062540  012766  013670  000026          MOV     #WDBUFF,16(SP)                  ; *,WDPTR               5985
11163 062546  012766  022670  000024          MOV     #WCBUFF,26(SP)                  ; *,WCPTR               5986
11164 062554  012766  023670  000034          MOV     #RDBUFF,24(SP)                  ; *,RDPTR               5987
11165 062562  016766  117224  000042          MOV     #RCBUFF,34(SP)                  ; *,RCPTR               5988
11166 062570  005066  000014                  MOV     LSLUNIT,42(SP)                  ;                       5990
11167 062574  000167  005362                  CLR     14(SP)                          ; LUN
11168 062600  016600  000014          1$:     JMP     95$
11169 062604  006200                          MOV     14(SP),R0                       ; LUN,*                 5995
11170 062606  006200                          ASR     R0
11171 062610  006200                          ASR     R0
11172 062612  062700  034442                  ASR     R0
11173 062616  010046                          ADD     #DRIVE.STATUS,R0
11174 062620  016646  000016                  MOV     R0,-(SP)
11175 062624  042716  177770                  MOV     16(SP),-(SP)                    ; LUN,*
11176 062630  012746  000001                  BIC     #177770,(SP)
11177 062634  005046                          MOV     #1,-(SP)
11178 062636  004767  121514                  CLR     -(SP)
11179 062642  062706  000010                  JSR     PC,BL$GT2
11180 062646  005300                          ADD     #10,SP
11181 062650  001402                          DEC     R0
11182 062652  000167  004630                  BEQ     2$
11183 062656  016667  000014  117210  2$:     JMP     83$
11184 062664  012746  007354                  MOV     14(SP),L$LUN                    ; LUN,*                 5998
11185 062670  012746  007072                  MOV     #WRD35,-(SP)                    ;                       5999
11186 062674  012746  000002                  MOV     #SAY1,-(SP)
11187 062700  010600                          MOV     #2,-(SP)
11188 062702  104414                          MOV     SP,R0                           ; SP,*
11189 062704  012716  007332                  TRAP    14
11190 062710  012746  006160                  MOV     #WRD24,(SP)                     ;                       6001
11191 062714  012746  000002                  MOV     #FMT2,-(SP)
                                              MOV     #2,-(SP)
```

SEQ 0255

```
                                        ;MLX4
                                        ;
                                        ;       DEFINITION OF OPTION 4                  27-Mar-1982 19:24:42    TOPS
                                        ;                                               27-Mar-1982 19:23:44    PA:<
11193
11194
11195
11196 062720  010600                          MOV     SP,R0                   ; SP,*
11197 062722  104414                          TRAP    14
11198 062724  016600  000026                  MOV     26(SP),R0               ; LUN,*                         6007
11199 062730  006300                          ASL     R0
11200 062732  012766  034460  000052          MOV     #LOW.SECT,52(SP)
11201 062740  060066  000052                  ADD     R0,52(SP)
11202 062744  012766  034500  000050          MOV     #TOP.SECT,50(SP)
11203 062752  060066  000050                  ADD     R0,50(SP)
11204 062756  017601  000050                  MOV     @50(SP),R1
11205 062762  017604  000052                  MOV     @52(SP),R4
11206 062766  000516                          BR      8$                      ; *,SECTOR
11207 062770  016646  000026          3$:     MOV     26(SP),-(SP)            ; LUN,*                         6009
11208 062774  012746  000400                  MOV     #400,-(SP)
11209 063000  016646  000034                  MOV     34(SP),-(SP)            ; WDPTR,*
11210 063004  010446                          MOV     R4,-(SP)                ; SECTOR,*
11211 063006  004767  162412                  JSR     PC,WRITE
11212 063012  010005                          MOV     R0,R5                   ; *,VALUE
11213 063014  020527  000001                  CMP     R5,#1                   ; VALUE,*
11214 063020  001036                          BNE     4$                              6015
11215 063022  012746  000006                  MOV     #6,-(SP)                ;               6021
11216 063026  012746  045424                  MOV     #WRITE,-(SP)
11217 063032  016646  000042                  MOV     42(SP),-(SP)            ; LUN,*
11218 063036  012746  000400                  MOV     #400,-(SP)
11219 063042  016646  000050                  MOV     50(SP),-(SP)            ; WDPTR,*
11220 063046  010446                          MOV     R4,-(SP)                ; SECTOR,*
11221 063050  004767  163150                  JSR     PC,RETRY
11222 063054  062706  000014                  ADD     #14,SP
11223 063060  005700                          TST     R0
11224 063062  001455                          BEQ     7$
11225 063064  016600  000036                  MOV     36(SP),R0
11226 063070  112760  000004  034446          MOVB    #4,WHY.DROPT(R0)        ; LUN,*                         6024
11227 063076  104455                          TRAP    55
11228 063100  000621                          .WORD   621                     ;                               6025
11229 063102  011070                          .WORD   MSG1
11230 063104  000000                          .WORD   0
11231 063106  016600  000036                  MOV     36(SP),R0
11232 063112  104451                          TRAP    51                      ; LUN,*                         6026
11233 063114  000435                          BR      6$                      ;                               6027
11234 063116  020527  000002          4$:     CMP     R5,#2                   ; VALUE,*                       6015
11235 063122  001015                          BNE     5$
11236 063124  016600  000036                  MOV     36(SP),R0
11237 063130  112760  000005  034446          MOVB    #5,WHY.DROPT(R0)        ; LUN,*                         6034
11238 063136  104455                          TRAP    55
11239 063140  000622                          .WORD   622                     ;                               6035
11240 063142  011070                          .WORD   MSG1
11241 063144  000000                          .WORD   0
11242 063146  016600  000036                  MOV     36(SP),R0
11243 063152  104451                          TRAP    51                      ; LUN,*                         6036
11244 063154  000415                          BR      6$                      ;                               6037
11245 063156  020527  000003          5$:     CMP     R5,#3                   ; VALUE,*                       6015
11246 063162  001015                          BNE     7$
11247 063164  104455                          TRAP    55                      ;                               6042
```

SEQ 0256

```
                                        ;MLX4
                                        ;
                                                DEFINITION OF OPTION 4                                  27-Mar-1982 19:24:42    TOPS
                                                                                                        27-Mar-1982 19:23:44    PA:<
11249
11250
11251
11252 063166  000623                              .WORD   623
11253 063170  011070                              .WORD   MSG1
11254 063172  000000                              .WORD   0
11255 063174  016600  000036                      MOV     36(SP),R0               ; LUN,*                         6043
11256 063200  112760  000006  034446              MOVB    #6,WHY.DROPT(R0)
11257 063206  104451                              TRAP    51
11258 063210  062706  000022              6$:     ADD     #22,SP                  ;                               6044
11259 063214  000565                              BR      14$                     ;                               6045
11260 063216  062706  000010              7$:     ADD     #10,SP                  ;                               6008
11261 063222  005204                              INC     R4                      ; SECTOR                        6007
11262 063224  020401                      8$:     CMP     R4,R1                   ; SECTOR,*
11263 063226  101660                              BLOS    3$
11264 063230  012716  007332                      MOV     #WRD24,(SP)             ;                               6051
11265 063234  012746  006160                      MOV     #FMT2,-(SP)
11266 063240  012746  000002                      MOV     #2,-(SP)
11267 063244  010600                              MOV     SP,R0                   ; SP,*
11268 063246  104414                              TRAP    14
11269 063250  004767  162712                      JSR     PC,CHOOSE               ;                               6053
11270 063254  010002                              MOV     R0,R2                   ; *,COMMAND
11271 063256  017666  000054  000046              MOV     @54(SP),46(SP)          ;                               6058
11272 063264  017604  000056                      MOV     @56(SP),R4              ; *,SECTOR
11273 063270  000167  001350                      JMP     33$
11274 063274  005001              9$:     CLR     R1                      ;                               6061
11275 063276  020227  045612                      CMP     R2,#READ                ; COMMAND *
11276 063302  001014                              BNE     10$
11277 063304  005201                              INC     R1
11278 063306  016603  000042                      MOV     42(SP),R3               ; RDPTR,PTR                     6064
11279 063312  016646  000032                      MOV     32(SP),-(SP)            ; LUN,*                         6065
11280 063316  012746  000400                      MOV     #400,-(SP)
11281 063322  010346                              MOV     R3,-(SP)                ; PTR,*
11282 063324  010446                              MOV     R4,-(SP)                ; SECTOR,*
11283 063326  004767  162260                      JSR     PC,READ
11284 063332  000412                              BR      11$
11285 063334  016603  000034              10$:    MOV     34(SP),R3               ; WDPTR,PTR                     6069
11286 063340  016646  000032                      MOV     32(SP),-(SP)            ; LUN,*                         6070
11287 063344  012746  000400                      MOV     #400,-(SP)
11288 063350  010346                              MOV     R3,-(SP)                ; PTR,*
11289 063352  010446                              MOV     R4,-(SP)                ; SECTOR,*
11290 063354  004767  162420                      JSR     PC,CHECK
11291 063360  010005              11$:    MOV     R0,R5                   ; *,VALUE
11292 063362  001103                              BNE     15$                     ;
11293 063364  006001                              ROR     R1                      ;                               6077
11294 063366  103402                              BLO     13$                     ;                               6082
11295 063370  000167  001012              12$:    JMP     28$
11296 063374  016646  000044              13$:    MOV     44(SP),-(SP)            ; WDPTR,*                       6086
11297 063400  016646  000054                      MOV     54(SP),-(SP)            ; RDPTR,*
11298 063404  012746  000400                      MOV     #400,-(SP)
11299 063410  004767  160712                      JSR     PC,DOUBLE.CHECK
11300 063414  062706  000006                      ADD     #6,SP
11301 063420  010066  000060                      MOV     R0,60(SP)               ; *,DBL.VALUE
11302 063424  001761                              BEQ     12$
11303 063426  016646  000042                      MOV     42(SP),-(SP)            ; LUN,*                         6089
```

SEQ 0257

```
                                        ;MLX4
                                        ;                DEFINITION OF OPTION 4                                  27-Mar-1982 19:24:42   TOPS
                                                                                                                 27-Mar-1982 19:23:44   PA:<
11308 063432 004767 152066              JSR     PC,SAYWHO
11309 063436 012716 011216              MOV     #MSG5,(SP)
11310 063442 012746 007072              MOV     #SAY1,-(SP)             ;                                                               6090
11311 063446 012746 000002              MOV     #2,-(SP)
11312 063452 010600                     MOV     SP,R0
11313 063454 104414                     TRAP    14                      ; SP,*
11314 063456 016616 000066              MOV     66(SP),(SP)
11315 063462 017646 000066              MOV     @66(SP),-(SP)           ; DBL.VALUE,*                                                   6092
11316 063466 012746 006710              MOV     #FMT12A,-(SP)           ; DBL.VALUE,*
11317 063472 012746 000003              MOV     #3,-(SP)
11318 063476 010600                     MOV     SP,R0
11319 063500 104414                     TRAP    14                      ; SP,*
11320 063502 062766 010000 000074       ADD     #10000,74(SP)           ; *,DBL.VALUE
11321 063510 016616 000074              MOV     74(SP),(SP)                                                                             6094
11322 063514 017646 000074              MOV     @74(SP),-(SP)           ; DBL.VALUE,*                                                   6095
11323 063520 012746 006756              MOV     #FMT12B,-(SP)           ; DBL.VALUE,*
11324 063524 012746 000003              MOV     #3,-(SP)
11325 063530 010600                     MOV     SP,R0
11326 063532 104414                     TRAP    14                      ; SP,*
11327 063534 016601 000064              MOV     64(SP),R1
11328 063540 112761 000010 034446       MOVB    #10,WHY.DROPT(R1)       ; LUN,*                                                         6097
11329 063546 104455                     TRAP    55
11330 063550 000624                     .WORD   624                     ;                                                               6098
11331 063552 011070                     .WORD   MSG1
11332 063554 000000                     .WORD   0
11333 063556 016600 000064              MOV     64(SP),R0
11334 063562 104451                     TRAP    51                      ; LUN,*                                                         6099
11335 063564 062706 000050              ADD     #50,SP
11336 063570 000520             14$:    BR      20$                     ;                                                               6100
11337 063572 020527 000001      15$:    CMP     R5,#1
11338 063576 001034                     BNE     16$                     ; VALUE,*                                                       6077
11339 063600 012746 000006              MOV     #6,-(SP)
11340 063604 010246                     MOV     R2,-(SP)                ;                                                               6108
11341 063606 016646 000046              MOV     46(SP),-(SP)            ; COMMAND,*
11342 063612 012746 000400              MOV     #400,-(SP)              ; LUN,*
11343 063616 010346                     MOV     R3,-(SP)
11344 063620 010446                     MOV     R4,-(SP)                ; PTR,*
11345 063622 004767 162376              JSR     PC,RETRY                ; SECTOR,*
11346 063626 062706 000014              ADD     #14,SP
11347 063632 005700                     TST     R0
11348 063634 001655                     BEQ     12$
11349 063636 016601 000042              MOV     42(SP),R1
11350 063642 112761 000004 034446       MOVB    #4,WHY.DROPT(R1)        ; LUN,*                                                         6111
11351 063650 104455                     TRAP    55
11352 063652 000625                     .WORD   625                     ;                                                               6112
11353 063654 011070                     .WORD   MSG1
11354 063656 000000                     .WORD   0
11355 063660 016600 000042              MOV     42(SP),R0
11356 063664 104451                     TRAP    51                      ; LUN,*                                                         6113
11357 063666 000457                     BR      19$                     ;                                                               6114
11358 063670 020527 000002      16$:    CMP     R5,#2                   ; VALUE,*                                                       6077
11359 063674 001015                     BNE     17$
```

```
11361                                   ;MLX4
11362                                   ;
11363                                   ;               DEFINITION OF OPTION 4
11364 063676  016601  000042                    MOV     42(SP),R1
11365 063702  112761  000005  034446            MOVB    #5,WHY.DROPT(R1)        ; LUN,*                 6121
11366 063710  104455                            TRAP    55                      ;                       6122
11367 063712  000626                            .WORD   626
11368 063714  011070                            .WORD   MSG1
11369 063716  000000                            .WORD   0
11370 063720  016600  000042                    MOV     42(SP),R0               ; LUN,*                 6123
11371 063724  104451                            TRAP    51
11372 063726  000437                            BR      19$                     ;                       6124
11373 063730  020527  000003          17$:      CMP     R5,#3                   ; VALUE,*               6077
11374 063734  001014                            BNE     18$
11375 063736  104455                            TRAP    55                      ;                       6129
11376 063740  000627                            .WORD   627
11377 063742  011070                            .WORD   MSG1
11378 063744  000000                            .WORD   0
11379 063746  016601  000042                    MOV     42(SP),R1
11380 063752  112761  000006  034446            MOVB    #6,WHY.DROPT(R1)        ; LUN,*                 6130
11381 063760  010100                            MOV     R1,R0                   ; LUN,*                 6131
11382 063762  104451                            TRAP    51
11383 063764  000420                            BR      19$                     ;                       6132
11384 063766  020527  000004          18$:      CMP     R5,#4                   ; VALUE,*               6077
11385 063772  001021                            BNE     21$
11386 063774  004767  153332                    JSR     PC,ISOLATE              ;                       6137
11387 064000  104455                            TRAP    55                      ;                       6138
11388 064002  000630                            .WORD   630
11389 064004  011120                            .WORD   MSG2
11390 064006  000000                            .WORD   0
11391 064010  016601  000042                    MOV     42(SP),R1
11392 064014  112761  000007  034446            MOVB    #7,WHY.DROPT(R1)        ; LUN,*                 6139
11393 064022  010100                            MOV     R1,R0                   ; LUN,*                 6140
11394 064024  104451                            TRAP    51
11395 064026  062706  000026          19$:      ADD     #26,SP                  ;                       6141
11396 064032  000167  004120          20$:      JMP     94$
11397 064036  020527  000005          21$:      CMP     R5,#5                   ; VALUE,*               6077
11398 064042  001161                            BNE     28$
11399 064044  004767  153262                    JSR     PC,ISOLATE              ;                       6146
11400 064050  032767  000001  116202            BIT     #1,ERROUT               ;                       6148
11401 064056  001423                            BEQ     22$
11402 064060  017701  150324                    MOV     @ML.REG+42,R1
11403 064064  006201                            ASR     R1
11404 064066  006201                            ASR     R1
11405 064070  006201                            ASR     R1
11406 064072  006201                            ASR     R1
11407 064074  006201                            ASR     R1
11408 064076  006201                            ASR     R1
11409 064100  042701  177700                    BIC     #177700,R1
11410 064104  010146                            MOV     R1,-(SP)
11411 064106  012746  006640                    MOV     #FMT10B,-(SP)
11412 064112  012746  000002                    MOV     #2,-(SP)
11413 064116  010600                            MOV     SP,R0
11414 064120  104414                            TRAP    14                      ; SP,*
11415 064122  062706  000006                    ADD     #6,SP
```

```
                                        ;MLX4
                                        ;                DEFINITION OF OPTION 4                                  27-Mar-1982 19:24:42    TOPS
                                        ;                                                                        27-Mar-1982 19:23.44    PA:<
11420 064126  017766  150260  000046  22$:    MOV     @ML.REG+44,46(SP)       ; *,OLDSEC                                              6151
11421 064134  017701  150250                   MOV     @ML.REG+42,R1           ;                                                       6152
11422 064140  006201                           ASR     R1
11423 064142  006201                           ASR     R1
11424 064144  006201                           ASR     R1
11425 064146  006201                           ASR     R1
11426 064150  006201                           ASR     R1
11427 064152  006201                           ASR     R1
11428 064154  042701  177700                   BIC     #177700,R1
11429 064160  010166  000050                   MOV     R1,50(SP)
11430 064164  012746  000001                   MOV     #1,-(SP)                ; *,OLDCHN
11431 064170  010246                           MOV     R2,-(SP)                ;                                                       6154
11432 064172  016646  000046                   MOV     46(SP),-(SP)            ; COMMAND,*
11433 064176  012746  000400                   MOV     #400,-(SP)              ; LUN,*
11434 064202  010346                           MOV     R3,-(SP)
11435 064204  010446                           MOV     R4,-(SP)                ; PTR,*
11436 064206  004767  162012                   JSR     PC,RETRY                ; SECTOR,*
11437 064212  062706  000014                   ADD     #14,SP
11438 064216  020027  000005                   CMP     R0,#5
11439 064222  001052                           BNE     25$
11440 064224  027766  150162  000046          CMP     @ML.REG+44,46(SP)       ; *,OLDSEC                                              6157
11441 064232  001035                           BNE     24$
11442 064234  016600  000050                   MOV     50(SP),R0               ; OLDCHN,*
11443 064240  017701  150144                   MOV     @ML.REG+42,R1
11444 064244  006201                           ASR     R1
11445 064246  006201                           ASR     R1
11446 064250  006201                           ASR     R1
11447 064252  006201                           ASR     R1
11448 064254  006201                           ASR     R1
11449 064256  006201                           ASR     R1
11450 064260  042701  177700                   BIC     #177700,R1
11451 064264  020100                           CMP     R1,R0
11452 064266  001017                           BNE     24$
11453 064270  032767  000001  115762          BIT     #1,ERROUT               ;                                                       6161
11454 064276  001404                           BEQ     23$
11455 064300  104456                           TRAP    56
11456 064302  000631                           .WORD   631
11457 064304  011202                           .WORD   MSG4
11458 064306  000000                           .WORD   0
11459 064310  016646  000042          23$:     MOV     42(SP),-(SP)            ; LUN,*                                                 6163
11460 064314  016746  150022                   MOV     BOARD,-(SP)
11461 064320  004767  153076                   JSR     PC,UP.HARD.COUNT
11462 064324  000427                           BR      27$
11463 064326  032767  000001  115724  24$:     BIT     #1,ERROUT               ;                                                       6157
11464 064334  001415                           BEQ     26$                     ;                                                       6168
11465 064336  104457                           TRAP    57
11466 064340  000632                           .WORD   632
11467 064342  011166                           .WORD   MSG3
11468 064344  000000                           .WORD   0
11469 064346  000410                           BR      26$
11470 064350  032767  000001  115702  25$:     BIT     #1,ERROUT               ;                                                       6170
11471 064356  001404                           BEQ     26$                     ;                                                       6176
```

SEQ 0260

```
11473                                   ;MLX4
11474                                   ;
11475                                   ;               DEFINITION OF OPTION 4                  27-Mar-1982 19:24:42    TOPS
11476 064360  104457                                    TRAP    57                              27-Mar-1982 19:23:44    PA:<
11477 064362  000633                                    .WORD   633
11478 064364  011166                                    .WORD   MSG3
11479 064366  000000                                    .WORD   0
11480 064370  016646  000042                    26$:    MOV     42(SP),-(SP)            ; LUN,*                         6178
11481 064374  016746  147742                            MOV     BOARD,-(SP)
11482 064400  004767  153500                            JSR     PC,UP.SOFT.COUNT
11483 064404  022626                            27$:    CMP     (SP)+,(SP)+             ;                               6145
11484 064406  016616  000042                    28$:    MOV     42(SP),(SP)             ; LUN,*                         6184
11485 064412  012746  000400                            MOV     #400,-(SP)
11486 064416  016646  000056                            MOV     56(SP),-(SP)            ; WCPTR,*
11487 064422  010446                                    MOV     R4,-(SP)                ; SECTOR,*
11488 064424  004767  160774                            JSR     PC,WRITE
11489 064430  010005                                    MOV     R0,R5                   ; *,VALUE
11490 064432  020527  000001                            CMP     R5,#1                   ; VALUE,*                       6190
11491 064436  001036                                    BNE     29$
11492 064440  012746  000006                            MOV     #6,-(SP)                ;                               6196
11493 064444  012746  045424                            MOV     #WRITE,-(SP)
11494 064450  016646  000054                            MOV     54(SP),-(SP)            ; LUN,*
11495 064454  012746  000400                            MOV     #400,-(SP)
11496 064460  016646  000072                            MOV     72(SP),-(SP)            ; WCPTR,*
11497 064464  010446                                    MOV     R4,-(SP)                ; SECTOR,*
11498 064466  004767  161532                            JSR     PC,RETRY
11499 064472  062706  000014                            ADD     #14,SP
11500 064476  005700                                    TST     R0
11501 064500  001456                                    BEQ     32$
11502 064502  016601  000050                            MOV     50(SP),R1               ; LUN,*                         6199
11503 064506  112761  000004  034446                    MOVB    #4,WHY.DROPT(R1)
11504 064514  104455                                    TRAP    55                      ;                               6200
11505 064516  000634                                    .WORD   634
11506 064520  011070                                    .WORD   MSG1
11507 064522  000000                                    .WORD   0
11508 064524  016600  000050                            MOV     50(SP),R0               ; LUN,*                         6201
11509 064530  104451                                    TRAP    51
11510 064532  000436                                    BR      31$                     ;                               6202
11511 064534  020527  000002                    29$:    CMP     R5,#2                   ; VALUE,*                       6190
11512 064540  001015                                    BNE     30$
11513 064542  016601  000050                            MOV     50(SP),R1               ; LUN,*                         6209
11514 064546  112761  000005  034446                    MOVB    #5,WHY.DROPT(R1)
11515 064554  104455                                    TRAP    55                      ;                               6210
11516 064556  000635                                    .WORD   635
11517 064560  011070                                    .WORD   MSG1
11518 064562  000000                                    .WORD   0
11519 064564  016600  000050                            MOV     50(SP),R0               ; LUN,*                         6211
11520 064570  104451                                    TRAP    51
11521 064572  000416                                    BR      31$                     ;                               6212
11522 064574  020527  000003                    30$:    CMP     R5,#3                   ; VALUE,*                       6190
11523 064600  001016                                    BNE     32$
11524 064602  104455                                    TRAP    55                      ;                               6217
11525 064604  000636                                    .WORD   636
11526 064606  011070                                    .WORD   MSG1
11527 064610  000000                                    .WORD   0
```

SEQ 0261

```
                                        ;MLX4
                                        ;
                                        ;       DEFINITION OF OPTION 4                                  27-Mar-1982 19:24:42    TOPS
                                        ;                                                               27-Mar-1982 19:23:44    PA:<
11532 064612 016601 000050                      MOV     50(SP),R1
11533 064616 112761 000006 034446               MOVB    #6,WHY.DROPT(R1)        ; LUN,*                                 6218
11534 064624 010100                             MOV     R1,R0
11535 064626 104451                             TRAP    51                      ; LUN,*                                 6219
11536 064630 062706 000034              31$:    ADD     #34,SP
11537 064634 000570                             BR      40$                     ;                                       6220
11538 064636 062706 000016              32$:    ADD     #16,SP
11539 064642 005204                             INC     R4                      ;                                       6059
11540 064644 020466 000046              33$:    CMP     R4,46(SP)               ; SECTOR
11541 064650 101002                             BHI     34$                     ; SECTOR,*                              6058
11542 064652 000167 176416                      JMP     9$
11543 064656 012716 007336              34$:    MOV     #WRD25,(SP)
11544 064662 012746 006160                      MOV     #FMT2,-(SP)             ;                                       6226
11545 064666 012746 000002                      MOV     #2,-(SP)
11546 064672 010600                             MOV     SP,R0
11547 064674 104414                             TRAP    14                      ; SP,*
11548 064676 004767 161264                      JSR     PC,CHOOSE
11549 064702 010002                             MOV     R0,R2                   ;                                       6228
11550 064704 017666 000062 000052               MOV     @62(SP),52(SP)          ; *,COMMAND
11551 064712 017604 000060                      MOV     @60(SP),R4              ;                                       6230
11552 064716 000167 001350                      JMP     59$                     ; *,SECTOR
11553 064722 005001                     35$:    CLR     R1
11554 064724 020227 045612                      CMP     R2,#READ                ;                                       6233
11555 064730 001014                             BNE     36$                     ; COMMAND,*
11556 064732 005201                             INC     R1
11557 064734 016603 000056                      MOV     56(SP),R3
11558 064740 016646 000036                      MOV     36(SP),-(SP)            ; RCPTR,PTR                             6236
11559 064744 012746 000400                      MOV     #400,-(SP)              ; LUN,*                                 6237
11560 064750 010346                             MOV     R3,-(SP)
11561 064752 010446                             MOV     R4,-(SP)                ; PTR,*
11562 064754 004767 160632                      JSR     PC,READ                 ; SECTOR,*
11563 064760 000412                             BR      37$
11564 064762 016603 000050              36$:    MOV     50(SP),R3
11565 064766 016646 000036                      MOV     36(SP),-(SP)            ; WCPTR,PTR                             6241
11566 064772 012746 000400                      MOV     #400,-(SP)              ; LUN,*                                 6242
11567 064776 010346                             MOV     R3,-(SP)
11568 065000 010446                             MOV     R4,-(SP)                ; PTR,*
11569 065002 004767 160772                      JSR     PC,CHECK                ; SECTOR,*
11570 065006 010005                     37$:    MOV     R0,R5
11571 065010 001103                             BNE     41$                     ; *,VALUE
11572 065012 006001                             ROR     R1                      ;                                       6249
11573 065014 103402                             BLO     39$                     ;                                       6254
11574 065016 000167 001012              38$:    JMP     54$
11575 065022 016646 000060              39$:    MOV     60(SP),-(SP)            ; WCPTR,*
11576 065026 016646 000070                      MOV     70(SP),-(SP)            ; RCPTR,*                               6258
11577 065032 012746 000400                      MOV     #400,-(SP)
11578 065036 004767 157264                      JSR     PC,DOUBLE.CHECK
11579 065042 062706 000006                      ADD     #6,SP
11580 065046 010066 000064                      MOV     R0,64(SP)
11581 065052 001761                             BEQ     38$                     ; *,DBL.VALUE
11582 065054 016646 000046                      MOV     46(SP),-(SP)            ; LUN,*
11583 065060 004767 150440                      JSR     PC,SAYWHO                                                       6261
```

```
11585                                   ;MLX4
11586                                   ;                DEFINITION OF OPTION 4                    27-Mar-1982 19:24:42   TOPS
11587                                                                                             27-Mar-1982 19:23:44   PA:<
11588 065064  012716  011216                     MOV     #MSG5,(SP)
11589 065070  012746  007072                     MOV     #SAY1,-(SP)          ;                                          6262
11590 065074  012746  000002                     MOV     #2,-(SP)
11591 065100  010600                             MOV     SP,R0
11592 065102  104414                             TRAP    14                   ; SP,*
11593 065104  016616  000072                     MOV     72(SP),(SP)
11594 065110  017646  000072                     MOV     @72(SP),-(SP)        ; DBL.VALUE,*                              6264
11595 065114  012746  006710                     MOV     #FMT12A,-(SP)        ; DBL.VALUE,*
11596 065120  012746  000003                     MOV     #3,-(SP)
11597 065124  010600                             MOV     SP,R0
11598 065126  104414                             TRAP    14                   ; SP,*
11599 065130  062766  010000  000100             ADD     #10000,100(SP)       ; *,DBL.VALUE
11600 065136  016616  000100                     MOV     100(SP),(SP)         ; DBL.VALUE,*                              6266
11601 065142  017646  000100                     MOV     @100(SP),-(SP)       ; DBL.VALUE,*                              6267
11602 065146  012746  006756                     MOV     #FMT12B,-(SP)
11603 065152  012746  000003                     MOV     #3,-(SP)
11604 065156  010600                             MOV     SP,R0
11605 065160  104414                             TRAP    14                   ; SP,*
11606 065162  016601  000070                     MOV     70(SP),R1
11607 065166  112761  000010  034446             MOVB    #10,WHY.DROPT(R1)    ; LUN,*                                    6269
11608 065174  104455                             TRAP    55
11609 065176  000637                             .WORD   637                  ;                                          6270
11610 065200  011070                             .WORD   MSG1
11611 065202  000000                             .WORD   0
11612 065204  016600  000070                     MOV     70(SP),R0
11613 065210  104451                             TRAP    51                   ; LUN,*                                    6271
11614 065212  062706  000054                     ADD     #54,SP
11615 065216  000520                             BR      46$                  ;                                          6272
11616 065220  020527  000001          41$:       CMP     R5,#1
11617 065224  001034                             BNE     42$                  ; VALUE,*                                  6249
11618 065226  012746  000006                     MOV     #6,-(SP)
11619 065232  010246                             MOV     R2,-(SP)             ;                                          6280
11620 065234  016646  000052                     MOV     52(SP),-(SP)         ; COMMAND,*
11621 065240  012746  000400                     MOV     #400,-(SP)           ; LUN,*
11622 065244  010346                             MOV     R3,-(SP)
11623 065246  010446                             MOV     R4,-(SP)             ; PTR,*
11624 065250  004767  160750                     JSR     PC,RETRY             ; SECTOR,*
11625 065254  062706  000014                     ADD     #14,SP
11626 065260  005700                             TST     R0
11627 065262  001655                             BEQ     38$
11628 065264  016601  000046                     MOV     46(SP),R1
11629 065270  112761  000004  034446             MOVB    #4,WHY.DROPT(R1)     ; LUN,*                                    6283
11630 065276  104455                             TRAP    55
11631 065300  000640                             .WORD   640                  ;                                          6284
11632 065302  011070                             .WORD   MSG1
11633 065304  000000                             .WORD   0
11634 065306  016600  000046                     MOV     46(SP),R0
11635 065312  104451                             TRAP    51                   ; LUN,*                                    6285
11636 065314  000457                             BR      45$                  ;                                          6286
11637 065316  020527  000002          42$:       CMP     R5,#2                ; VALUE,*                                  6249
11638 065322  001015                             BNE     43$
11639 065324  016601  000046                     MOV     46(SP),R1            ; LUN,*                                    6293
```

```
11641                                   ;MLX4
11642                                   ;                       DEFINITION OF OPTION 4                  27-Mar-1982 19:24:42    TOPS
11643                                                                                                   27-Mar-1982 19:23:44    PA:<
11644 065330  112761  000005  034446            MOVB    #5,WHY.DROPT(R1)
11645 065336  104455                            TRAP    55
11646 065340  000641                            .WORD   641                     ;                                               6294
11647 065342  011070                            .WORD   MSG1
11648 065344  000000                            .WORD   0
11649 065346  016600  000046                    MOV     46(SP),R0               ; LUN,*                                         6295
11650 065352  104451                            TRAP    51
11651 065354  000437                            BR      45$
11652 065356  020527  000003            43$:    CMP     R5,#3                   ;                                               6296
11653 065362  001014                            BNE     44$                     ; VALUE,*                                       6249
11654 065364  104455                            TRAP    55
11655 065366  000642                            .WORD   642                     ;                                               6301
11656 065370  011070                            .WORD   MSG1
11657 065372  000000                            .WORD   0
11658 065374  016601  000046                    MOV     46(SP),R1
11659 065400  112761  000006  034446            MOVB    #6,WHY.DROPT(R1)        ; LUN,*                                         6302
11660 065406  010100                            MOV     R1,R0
11661 065410  104451                            TRAP    51                      ; LUN,*                                         6303
11662 065412  000420                            BR      45$
11663 065414  020527  000004            44$:    CMP     R5,#4                   ;                                               6304
11664 065420  001021                            BNE     47$                     ; VALUE,*                                       6249
11665 065422  004767  151704                    JSR     PC,ISOLATE
11666 065426  104455                            TRAP    55                      ;                                               6309
11667 065430  000643                            .WORD   643                     ;                                               6310
11668 065432  011120                            .WORD   MSG2
11669 065434  000000                            .WORD   0
11670 065436  016601  000046                    MOV     46(SP),R1
11671 065442  112761  000007  034446            MOVB    #7,WHY.DROPT(R1)        ; LUN,*                                         6311
11672 065450  010100                            MOV     R1,R0
11673 065452  104451                            TRAP    51                      ; LUN,*                                         6312
11674 065454  062706  000032            45$:    ADD     #32,SP
11675 065460  000167  002472            46$:    JMP     94$                     ;                                               6313
11676 065464  020527  000005            47$:    CMP     R5,#5
11677 065470  001161                            BNE     54$                     ; VALUE,*                                       6249
11678 065472  004767  151634                    JSR     PC,ISOLATE
11679 065476  032767  000001  114554            BIT     #1,ERROUT               ;                                               6318
11680 065504  001423                            BEQ     48$                     ;                                               6320
11681 065506  017701  146676                    MOV     @ML.REG+42,R1
11682 065512  006201                            ASR     R1
11683 065514  006201                            ASR     R1
11684 065516  006201                            ASR     R1
11685 065520  006201                            ASR     R1
11686 065522  006201                            ASR     R1
11687 065524  006201                            ASR     R1
11688 065526  042701  177700                    BIC     #177700,R1
11689 065532  010146                            MOV     R1,-(SP)
11690 065534  012746  006640                    MOV     #FMT10B,-(SP)
11691 065540  012746  000002                    MOV     #2,-(SP)
11692 065544  010600                            MOV     SP,R0                   ; SP,*
11693 065546  104414                            TRAP    14
11694 065550  062706  000006                    ADD     #6,SP
11695 065554  017766  146632  000052  48$:     MOV     @ML.REG+44,52(SP)       ; *,OLDSEC                                      6323
```

```
                                        ;MLX4
                                        ;                DEFINITION OF OPTION 4                                  27-Mar-1982 19:24:42    TOPS
                                                                                                                 27-Mar-1982 19:23:44    PA:<
11700 065562  017701  146622                        MOV     @ML.REG+42,R1           ;                                               6324
11701 065566  006201                                ASR     R1
11702 065570  006201                                ASR     R1
11703 065572  006201                                ASR     R1
11704 065574  006201                                ASR     R1
11705 065576  006201                                ASR     R1
11706 065600  006201                                ASR     R1
11707 065602  042701  177700                        BIC     #177700,R1
11708 065606  010166  000054                        MOV     R1,54(SP)               ; *,OLDCHN
11709 065612  012746  000001                        MOV     #1,-(SP)                ;                               6326
11710 065616  010246                                MOV     R2,-(SP)                ; COMMAND,*
11711 065620  016646  000052                        MOV     52(SP),-(SP)            ; LUN,*
11712 065624  012746  000400                        MOV     #400,-(SP)
11713 065630  010346                                MOV     R3,-(SP)                ; PTR,*
11714 065632  010446                                MOV     R4,-(SP)                ; SECTOR,*
11715 065634  004767  160364                        JSR     PC,RETRY
11716 065640  062706  000014                        ADD     #14,SP
11717 065644  020027  000005                        CMP     R0,#5
11718 065650  001052                                BNE     51$
11719 065652  027766  146534  000052                CMP     @ML.REG+44,52(SP)       ; *,OLDSEC                      6329
11720 065660  001035                                BNE     50$
11721 065662  016600  000054                        MOV     54(SP),R0               ; OLDCHN,*
11722 065666  017701  146516                        MOV     @ML.REG+42,R1
11723 065672  006201                                ASR     R1
11724 065674  006201                                ASR     R1
11725 065676  006201                                ASR     R1
11726 065700  006201                                ASR     R1
11727 065702  006201                                ASR     R1
11728 065704  006201                                ASR     R1
11729 065706  042701  177700                        BIC     #177700,R1
11730 065712  020100                                CMP     R1,R0
11731 065714  001017                                BNE     50$
11732 065716  032767  000001  114334                BIT     #1,ERROUT               ;                               6333
11733 065724  001404                                BEQ     49$
11734 065726  104456                                TRAP    56
11735 065730  000644                                .WORD   644
11736 065732  011202                                .WORD   MSG4
11737 065734  000000                                .WORD   0
11738 065736  016646  000046          49$:          MOV     46(SP),-(SP)            ; LUN,*                         6335
11739 065742  016746  146374                        MOV     BOARD,-(SP)
11740 065746  004767  151450                        JSR     PC,UP.HARD.COUNT
11741 065752  000427                                BR      53$
11742 065754  032767  000001  114276  50$:          BIT     #1,ERROUT               ;                               6329
11743 065762  001415                                BEQ     52$                     ;                               6340
11744 065764  104457                                TRAP    57
11745 065766  000645                                .WORD   645
11746 065770  011166                                .WORD   MSG3
11747 065772  000000                                .WORD   0
11748 065774  000410                                BR      52$
11749 065776  032767  000001  114254  51$:          BIT     #1,ERROUT               ;                               6342
11750 066004  001404                                BEQ     52$                     ;                               6348
11751 066006  104457                                TRAP    57
```

SEQ 0265

```
                                        ;MLX4
                                        ;                DEFINITION OF OPTION 4                                  27-Mar-1982 19:24:42    TOPS
                                                                                                                 27-Mar-1982 19:23:44    PA:<
11753
11754
11755
11756 066010  000646                              .WORD   646
11757 066012  011166                              .WORD   MSG3
11758 066014  000000                              .WORD   0
11759 066016  016646  000046              52$:    MOV     46(SP),-(SP)            ; LUN,*                         6350
11760 066022  016746  146314                      MOV     BOARD,-(SP)
11761 066026  004767  152052                      JSR     PC,UP.SOFT.COUNT
11762 066032  022626                      53$:    CMP     (SP)+,(SP)+             ;                               6317
11763 066034  016616  000046              54$:    MOV     46(SP),(SP)             ; LUN,*                         6356
11764 066040  012746  000400                      MOV     #400,-(SP)
11765 066044  016646  000052                      MOV     52(SP),-(SP)            ; WDPTR,*
11766 066050  010446                              MOV     R4,-(SP)                ; SECTOR,*
11767 066052  004767  157346                      JSR     PC,WRITE
11768 066056  010005                              MOV     R0,R5                   ; *,VALUE
11769 066060  020527  000001                      CMP     R5,#1                   ; VALUE,*
11770 066064  001036                              BNE     55$                             6362
11771 066066  012746  000006                      MOV     #6,-(SP)                ;                               6368
11772 066072  012746  045424                      MOV     #WRITE,-(SP)
11773 066076  016646  000060                      MOV     60(SP),-(SP)            ; LUN,*
11774 066102  012746  000400                      MOV     #400,-(SP)
11775 066106  016646  000066                      MOV     66(SP),-(SP)            ; WDPTR,*
11776 066112  010446                              MOV     R4,-(SP)                ; SECTOR,*
11777 066114  004767  160104                      JSR     PC,RETRY
11778 066120  062706  000014                      ADD     #14,SP
11779 066124  005700                              TST     R0
11780 066126  001456                              BEQ     58$
11781 066130  016601  000054                      MOV     54(SP),R1
11782 066134  112761  000004  034446              MOVB    #4,WHY.DROPT(R1)        ; LUN,*                         6371
11783 066142  104455                              TRAP    55
11784 066144  000647                              .WORD   647                     ;                               6372
11785 066146  011070                              .WORD   MSG1
11786 066150  000000                              .WORD   0
11787 066152  016600  000054                      MOV     54(SP),R0               ; LUN,*
11788 066156  104451                              TRAP    51                              6373
11789 066160  000436                              BR      57$
11790 066162  020527  000002              55$:    CMP     R5,#2                   ;                               6374
11791 066166  001015                              BNE     56$                     ; VALUE,*                       6362
11792 066170  016601  000054                      MOV     54(SP),R1
11793 066174  112761  000005  034446              MOVB    #5,WHY.DROPT(R1)        ; LUN,*                         6381
11794 066202  104455                              TRAP    55
11795 066204  000650                              .WORD   650                     ;                               6382
11796 066206  011070                              .WORD   MSG1
11797 066210  000000                              .WORD   0
11798 066212  016600  000054                      MOV     54(SP),R0               ; LUN,*
11799 066216  104451                              TRAP    51                              6383
11800 066220  000416                              BR      57$
11801 066222  020527  000003              56$:    CMP     R5,#3                   ;                               6384
11802 066226  001016                              BNE     58$                     ; VALUE,*                       6362
11803 066230  104455                              TRAP    55
11804 066232  000651                              .WORD   651                     ;                               6389
11805 066234  011070                              .WORD   MSG1
11806 066236  000000                              .WORD   0
11807 066240  016601  000054                      MOV     54(SP),R1               ; LUN,*                         6390
```

SEQ 0266

```
                                        ;MLX4
                                        ;                DEFINITION OF OPTION 4                                  27-Mar-1982 19:24:42    TOPS
                                                                                                                 27-Mar-1982 19:23:44    PA:<
11809
11810
11811
11812 066244  112761  000006  034446            MOVB    #6,WHY.DROPT(R1)
11813 066252  010100                            MOV     R1,R0
11814 066254  104451                            TRAP    51                      ; LUN,*                         6391
11815 066256  062706  000040          57$:      ADD     #40,SP                  ;                               6392
11816 066262  000570                            BR      66$
11817 066264  062706  000016          58$:      ADD     #16,SP                  ;                               6231
11818 066270  005304                            DEC     R4                      ; SECTOR                        6230
11819 066272  020466  000052          59$:      CMP     R4,52(SP)               ; SECTOR,*
11820 066276  002402                            BLT     60$
11821 066300  000167  176416                    JMP     35$
11822 066304  012716  007332          60$:      MOV     #WRD24,(SP)             ;                               6398
11823 066310  012746  006160                    MOV     #FMT2,-(SP)
11824 066314  012746  000002                    MOV     #2,-(SP)
11825 066320  010600                            MOV     SP,R0                   ; SP,*
11826 066322  104414                            TRAP    14
11827 066324  004767  157636                    JSR     PC,CHOOSE               ;                               6400
11828 066330  010002                            MOV     R0,R2                   ; *,COMMAND
11829 066332  017666  000064  000064            MOV     @64(SP),64(SP)          ;                               6405
11830 066340  017604  000066                    MOV     @66(SP),R4              ; *,SECTOR
11831 066344  000167  001120                    JMP     81$
11832 066350  005001          61$:      CLR     R1                      ;                               6408
11833 066352  020227  045612                    CMP     R2,#READ                ; COMMAND,*
11834 066356  001014                            BNE     62$
11835 066360  005201                            INC     R1
11836 066362  016603  000052                    MOV     52(SP),R3               ; RDPTR,PTR                     6411
11837 066366  016646  000042                    MOV     42(SP),-(SP)            ; LUN,*                         6412
11838 066372  012746  000400                    MOV     #400,-(SP)
11839 066376  010346                            MOV     R3,-(SP)                ; PTR,*
11840 066400  010446                            MOV     R4,-(SP)                ; SECTOR,*
11841 066402  004767  157204                    JSR     PC,READ
11842 066406  000412                            BR      63$
11843 066410  016603  000044          62$:      MOV     44(SP),R3               ; WDPTR,PTR                     6416
11844 066414  016646  000042                    MOV     42(SP),-(SP)            ; LUN,*                         6417
11845 066420  012746  000400                    MOV     #400,-(SP)
11846 066424  010346                            MOV     R3,-(SP)                ; PTR,*
11847 066426  010446                            MOV     R4,-(SP)                ; SECTOR,*
11848 066430  004767  157344                    JSR     PC,CHECK
11849 066434  010005          63$:      MOV     R0,R5                   ; *,VALUE
11850 066436  001103                            BNE     67$                     ;
11851 066440  006001                            ROR     R1                      ;                               6424
11852 066442  103402                            BLO     65$                     ;                               6429
11853 066444  000167  001012          64$:      JMP     80$
11854 066450  016646  000054          65$:      MOV     54(SP),-(SP)            ; WDPTR,*                       6433
11855 066454  016646  000064                    MOV     64(SP),-(SP)            ; RDPTR,*
11856 066460  012746  000400                    MOV     #400,-(SP)
11857 066464  004767  155636                    JSR     PC,DOUBLE.CHECK
11858 066470  062706  000006                    ADD     #6,SP
11859 066474  010066  000070                    MOV     R0,70(SP)               ; *,DBL.VALUE
11860 066500  001761                            BEQ     64$
11861 066502  016646  000052                    MOV     52(SP),-(SP)            ; LUN,*                         6436
11862 066506  004767  147012                    JSR     PC,SAYWHO
11863 066512  012716  011216                    MOV     #MSG5,(SP)              ;                               6437
```

SEQ 0267

```
11865                                   ;MLX4
11866                                   ;
11867                                   ;       DEFINITION OF OPTION 4                  27-Mar-1982 19:24:42    TOPS
11868 066516  012746  007072                    MOV     #SAY1,-(SP)                     27-Mar-1982 19:23:44    PA:<
11869 066522  012746  000002                    MOV     #2,-(SP)
11870 066526  010600                            MOV     SP,R0
11871 066530  104414                            TRAP    14                      ; SP,*
11872 066532  016616  000076                    MOV     76(SP),(SP)             ; DBL.VALUE,*                   6439
11873 066536  017646  000076                    MOV     @76(SP),-(SP)           ; DBL.VALUE,*
11874 066542  012746  006710                    MOV     #FMT12A,-(SP)
11875 066546  012746  000003                    MOV     #3,-(SP)
11876 066552  010600                            MOV     SP,R0
11877 066554  104414                            TRAP    14                      ; SP,*
11878 066556  062766  010000  000104            ADD     #10000,104(SP)          ; *,DBL.VALUE                   6441
11879 066564  016616  000104                    MOV     104(SP),(SP)            ; DBL.VALUE,*                   6442
11880 066570  017646  000104                    MOV     @104(SP),-(SP)          ; DBL.VALUE,*
11881 066574  012746  006756                    MOV     #FMT12B,-(SP)
11882 066600  012746  000003                    MOV     #3,-(SP)
11883 066604  010600                            MOV     SP,R0
11884 066606  104414                            TRAP    14                      ; SP,*
11885 066610  016601  000074                    MOV     74(SP),R1
11886 066614  112761  000010  034446            MOVB    #10,WHY.DROPT(R1)       ; LUN,*                         6444
11887 066622  104455                            TRAP    55
11888 066624  000652                            .WORD   652                     ;                               6445
11889 066626  011070                            .WORD   MSG1
11890 066630  000000                            .WORD   0
11891 066632  016600  000074                    MOV     74(SP),R0
11892 066636  104451                            TRAP    51                      ; LUN,*                         6446
11893 066640  062706  000060                    ADD     #60,SP
11894 066644  000520                    66$:    BR      72$                     ;                               6447
11895 066646  020527  000001            67$:    CMP     R5,#1
11896 066652  001034                            BNE     68$                     ; VALUE,*                       6424
11897 066654  012746  000006                    MOV     #6,-(SP)
11898 066660  010246                            MOV     R2,-(SP)                ;                               6455
11899 066662  016646  000056                    MOV     56(SP),-(SP)            ; COMMAND,*
11900 066666  012746  000400                    MOV     #400,-(SP)              ; LUN,*
11901 066672  010346                            MOV     R3,-(SP)
11902 066674  010446                            MOV     R4,-(SP)                ; PTR,*
11903 066676  004767  157322                    JSR     PC,RETRY                ; SECTOR,*
11904 066702  062706  000014                    ADD     #14,SP
11905 066706  005700                            TST     R0
11906 066710  001655                            BEQ     64$
11907 066712  016601  000052                    MOV     52(SP),R1
11908 066716  112761  000004  034446            MOVB    #4,WHY.DROPT(R1)        ; LUN,*                         6458
11909 066724  104455                            TRAP    55
11910 066726  000653                            .WORD   653                     ;                               6459
11911 066730  011070                            .WORD   MSG1
11912 066732  000000                            .WORD   0
11913 066734  016600  000052                    MOV     52(SP),R0
11914 066740  104451                            TRAP    51                      ; LUN,*                         6460
11915 066742  000457                            BR      71$                     ;
11916 066744  020527  000002            68$:    CMP     R5,#2                   ; VALUE,*                       6461
11917 066750  001015                            BNE     69$                                                     6424
11918 066752  016601  000052                    MOV     52(SP),R1
11919 066756  112761  000005  034446            MOVB    #5,WHY.DROPT(R1)        ; LUN,*                         6468
```

```
                                                                                                                    SEQ 0268

                        ;MLX4                                                                   27-Mar-1982 19:24:42    TOPS
                        ;               DEFINITION OF OPTION 4                                  27-Mar-1982 19:23:44    PA:<

066764  104455                  TRAP    55                                                                              6469
066766  000654                  .WORD   654                      ;
066770  011070                  .WORD   MSG1
066772  000000                  .WORD   0
066774  016600  000052          MOV     52(SP),R0                ; LUN,*                                                6470
067000  104451                  TRAP    51
067002  000437                  BR      71$                      ;                                                      6471
067004  020527  000003  69$:    CMP     R5,#3                    ; VALUE,*                                              6424
067010  001014                  BNE     70$
067012  104455                  TRAP    55                       ;                                                      6476
067014  000655                  .WORD   655
067016  011070                  .WORD   MSG1
067020  000000                  .WORD   0
067022  016601  000052          MOV     52(SP),R1
067026  112761  000006  034446  MOVB    #6,WHY.DROPT(R1)         ; LUN,*                                                6477
067034  010100                  MOV     R1,R0
067036  104451                  TRAP    51                       ; LUN,*                                                6478
067040  000420                  BR      71$                      ;                                                      6479
067042  020527  000004  70$:    CMP     R5,#4                    ; VALUE,*                                              6424
067046  001021                  BNE     73$
067050  004767  150256          JSR     PC,ISOLATE               ;                                                      6484
067054  104455                  TRAP    55                       ;                                                      6485
067056  000656                  .WORD   656
067060  011120                  .WORD   MSG2
067062  000000                  .WORD   0
067064  016601  000052          MOV     52(SP),R1
067070  112761  000007  034446  MOVB    #7,WHY.DROPT(R1)         ; LUN,*                                                6486
067076  010100                  MOV     R1,R0
067100  104451                  TRAP    51                       ; LUN,*                                                6487
067102  062706  000036  71$:    ADD     #36,SP
067106  000167  001044  72$:    JMP     94$                      ;                                                      6488
067112  020527  000005  73$:    CMP     R5,#5
067116  001161                  BNE     80$                      ; VALUE,*                                              6424
067120  004767  150206          JSR     PC,ISOLATE               ;                                                      6493
067124  032767  000001  113126  BIT     #1,ERROUT                ;                                                      6495
067132  001423                  BEQ     74$
067134  017701  145250          MOV     @ML.REG+42,R1
067140  006201                  ASR     R1
067142  006201                  ASR     R1
067144  006201                  ASR     R1
067146  006201                  ASR     R1
067150  006201                  ASR     R1
067152  006201                  ASR     R1
067154  042701  177700          BIC     #177700,R1
067160  010146                  MOV     R1,-(SP)
067162  012746  006640          MOV     #FMT10B,-(SP)
067166  012746  000002          MOV     #2,-(SP)
067172  010600                  MOV     SP,R0
067174  104414                  TRAP    14                       ; SP,*
067176  062706  000006          ADD     #6,SP
067202  017766  145204  000056  74$:    MOV     @ML.REG+44,56(SP)  ; *,OLDSEC                                          6498
067210  017701  145174          MOV     @ML.REG+42,R1            ;                                                      6499
```

```
                                ;MLX4                                                    27-Mar-1982 19:24:42    TOPS
                                ;                DEFINITION OF OPTION 4                  27-Mar-1982 19:23:44    PA:<
 11977
 11978
 11979
 11980 067214  006201                          ASR     R1
 11981 067216  006201                          ASR     R1
 11982 067220  006201                          ASR     R1
 11983 067222  006201                          ASR     R1
 11984 067224  006201                          ASR     R1
 11985 067226  006201                          ASR     R1
 11986 067230  042701  177700                  BIC     #177700,R1
 11987 067234  010166  000060                  MOV     R1,60(SP)
 11988 067240  012746  000001                  MOV     #1,-(SP)        ; *,OLDCHN
 11989 067244  010246                          MOV     R2,-(SP)        ;                       6501
 11990 067246  016646  000056                  MOV     56(SP),-(SP)    ; COMMAND,*
 11991 067252  012746  000400                  MOV     #400,-(SP)      ; LUN,*
 11992 067256  010346                          MOV     R3,-(SP)
 11993 067260  010446                          MOV     R4,-(SP)        ; PTR,*
 11994 067262  004767  156736                  JSR     PC,RETRY        ; SECTOR,*
 11995 067266  062706  000014                  ADD     #14,SP
 11996 067272  020027  000005                  CMP     R0,#5
 11997 067276  001052                          BNE     77$
 11998 067300  027766  145106  000056          CMP     @ML.REG+44,56(SP)       ; *,OLDSEC      6504
 11999 067306  001035                          BNE     76$
 12000 067310  016600  000060                  MOV     60(SP),R0               ; OLDCHN,*
 12001 067314  017701  145070                  MOV     @ML.REG+42,R1
 12002 067320  006201                          ASR     R1
 12003 067322  006201                          ASR     R1
 12004 067324  006201                          ASR     R1
 12005 067326  006201                          ASR     R1
 12006 067330  006201                          ASR     R1
 12007 067332  006201                          ASR     R1
 12008 067334  042701  177700                  BIC     #177700,R1
 12009 067340  020100                          CMP     R1,R0
 12010 067342  001017                          BNE     76$
 12011 067344  032767  000001  112706          BIT     #1,ERROUT       ;               6508
 12012 067352  001404                          BEQ     75$
 12013 067354  104456                          TRAP    56
 12014 067356  000657                          .WORD   657
 12015 067360  011202                          .WORD   MSG4
 12016 067362  000000                          .WORD   0
 12017 067364  016646  000052          75$:    MOV     52(SP),-(SP)    ; LUN,*         6510
 12018 067370  016746  144746                  MOV     BOARD,-(SP)
 12019 067374  004767  150022                  JSR     PC,UP.HARD.COUNT
 12020 067400  000427                          BR      79$
 12021 067402  032767  000001  112650  76$:    BIT     #1,ERROUT       ;               6504
 12022 067410  001415                          BEQ     78$             ;               6515
 12023 067412  104457                          TRAP    57
 12024 067414  000660                          .WORD   660
 12025 067416  011166                          .WORD   MSG3
 12026 067420  000000                          .WORD   0
 12027 067422  000410                          BR      78$
 12028 067424  032767  000001  112626  77$:    BIT     #1,ERROUT       ;               6517
 12029 067432  001404                          BEQ     78$             ;               6523
 12030 067434  104457                          TRAP    57
 12031 067436  000661                          .WORD   661
```

```
12033                                   ;MLX4
12034                                   ;                                                       27-Mar-1982 19:24:42    TOPS
12035                                   ;               DEFINITION OF OPTION 4                  27-Mar-1982 19:23:44    PA:<
12036 067440  011166                              .WORD   MSG3
12037 067442  000000                              .WORD   0
12038 067444  016646  000052            78$:     MOV     52(SP),-(SP)            ; LUN,*                                 6525
12039 067450  016746  144666                      MOV     BOARD,-(SP)
12040 067454  004767  150424                      JSR     PC,UP.SOFT.COUNT
12041 067460  022626                      79$:    CMP     (SP)+,(SP)+             ;                                       6492
12042 067462  062706  000010            80$:     ADD     #10,SP                  ;                                       6406
12043 067466  005204                              INC     R4                      ; SECTOR                                6405
12044 067470  020466  000064            81$:     CMP     R4,64(SP)               ; SECTOR,*
12045 067474  101002                              BHI     82$
12046 067476  000167  176646                      JMP     61$
12047 067502  062706  000026            82$:     ADD     #26,SP                  ;                                       5997
12048 067506  032767  000001  112546    83$:     BIT     #1,EFN$21               ;                                       6541
12049 067514  001002                              BNE     84$
12050 067516  000167  000434                      JMP     94$
12051 067522  016600  000014            84$:     MOV     14(SP),R0               ; LUN,*                                 6548
12052 067526  006300                              ASL     R0
12053 067530  016001  034500                      MOV     TOP.SECT(R0),R1
12054 067534  016004  034460                      MOV     LOW.SECT(R0),R4         ; *,SECTOR
12055 067540  000167  000402                      JMP     93$
12056 067544  016646  000014            85$:     MOV     14(SP),-(SP)            ; LUN,*                                 6550
12057 067550  012746  000400                      MOV     #400,-(SP)
12058 067554  012746  022670                      MOV     #RBUFF,-(SP)
12059 067560  010446                              MOV     R4,-(SP)                ; SECTOR,*
12060 067562  004767  156024                      JSR     PC,READ
12061 067566  062706  000010                      ADD     #10,SP
12062 067572  020027  000005                      CMP     R0,#5
12063 067576  001162                              BNE     92$
12064 067600  004767  147526                      JSR     PC,ISOLATE
12065 067604  032767  000001  112446            BIT     #1,ERROUT               ;                                       6553
12066 067612  001423                              BEQ     86$                     ;                                       6555
12067 067614  017700  144570                      MOV     @ML.REG+42,R0
12068 067620  006200                              ASR     R0
12069 067622  006200                              ASR     R0
12070 067624  006200                              ASR     R0
12071 067626  006200                              ASR     R0
12072 067630  006200                              ASR     R0
12073 067632  006200                              ASR     R0
12074 067634  042700  177700                      BIC     #177700,R0
12075 067640  010046                              MOV     R0,-(SP)
12076 067642  012746  006640                      MOV     #FMT10B,-(SP)
12077 067646  012746  000002                      MOV     #2,-(SP)
12078 067652  010600                              MOV     SP,R0
12079 067654  104414                              TRAP    14                      ; SP,*
12080 067656  062706  000006                      ADD     #6,SP
12081 067662  017766  144524  000020  86$:       MOV     @ML.REG+44,20(SP)       ; *,OLDSEC
12082 067670  017700  144514                      MOV     @ML.REG+42,R0           ;                                       6564
12083 067674  006200                              ASR     R0                      ;                                       6565
12084 067676  006200                              ASR     R0
12085 067700  006200                              ASR     R0
12086 067702  006200                              ASR     R0
12087 067704  006200                              ASR     R0
```

```
                                                                                                                    SEQ 0271
                                ;MLX4                                                                  27-Mar-1982 19:24:42    TOPS
                                ;                       DEFINITION OF OPTION 4                         27-Mar-1982 19:23:44    PA:<
12092 067706  006200                          ASR       R0
12093 067710  042700  177700                  BIC       #177700,R0
12094 067714  010066  000022                  MOV       R0,22(SP)
12095 067720  012746  000001                  MOV       #1,-(SP)          ; *,OLDCHN                                    6573
12096 067724  010246                          MOV       R2,-(SP)          ;
12097 067726  016646  000020                  MOV       20(SP),-(SP)      ; COMMAND,*
12098 067732  012746  000400                  MOV       #400,-(SP)        ; LUN,*
12099 067736  010346                          MOV       R3,-(SP)          ; PTR,*
12100 067740  016646  000032                  MOV       32(SP),-(SP)      ; OLDSEC,*
12101 067744  004767  156254                  JSR       PC,RETRY
12102 067750  062706  000014                  ADD       #14,SP
12103 067754  020027  000005                  CMP       R0,#5
12104 067760  001052                          BNE       89$
12105 067762  027766  144424  000020          CMP       @ML.REG+44,20(SP) ; *,OLDSEC                                  6582
12106 067770  001035                          BNE       88$
12107 067772  016646  000022                  MOV       22(SP),-(SP)      ; OLDCHN,*
12108 067776  017700  144406                  MOV       @ML.REG+42,R0
12109 070002  006200                          ASR       R0
12110 070004  006200                          ASR       R0
12111 070006  006200                          ASR       R0
12112 070010  006200                          ASR       R0
12113 070012  006200                          ASR       R0
12114 070014  006200                          ASR       R0
12115 070016  042700  177700                  BIC       #177700,R0
12116 070022  020026                          CMP       R0,(SP)+
12117 070024  001017                          BNE       88$
12118 070026  032767  000001  112224          BIT       #1,ERROUT         ;                                           6586
12119 070034  001404                          BEQ       87$
12120 070036  104456                          TRAP      56                ;                                           6591
12121 070040  000662                          .WORD     662
12122 070042  011202                          .WORD     MSG4
12123 070044  000000                          .WORD     0
12124 070046  016646  000014          87$:    MOV       14(SP),-(SP)      ; LUN,*                                     6594
12125 070052  016746  144264                  MOV       BOARD,-(SP)
12126 070056  004767  147340                  JSR       PC,UP.HARD.COUNT
12127 070062  000427                          BR        91$
12128 070064  032767  000001  112166  88$:    BIT       #1,ERROUT         ;                                           6582
12129 070072  001415                          BEQ       90$               ;                                           6599
12130 070074  104457                          TRAP      57                ;                                           6604
12131 070076  000663                          .WORD     663
12132 070100  011166                          .WORD     MSG3
12133 070102  000000                          .WORD     0
12134 070104  000410                          BR        90$
12135 070106  032767  000001  112144  89$:    BIT       #1,ERROUT         ;                                           6607
12136 070114  001404                          BEQ       90$               ;                                           6613
12137 070116  104457                          TRAP      57                ;                                           6618
12138 070120  000664                          .WORD     664
12139 070122  011166                          .WORD     MSG3
12140 070124  000000                          .WORD     0
12141 070126  016646  000014          90$:    MOV       14(SP),-(SP)      ; LUN,*                                     6621
12142 070132  016746  144204                  MOV       BOARD,-(SP)
12143 070136  004767  147742                  JSR       PC,UP.SOFT.COUNT
```

```
12145                                   ;MLX4
12146                                   .                       DEFINITION OF OPTION 4                                  27-Mar-1982 19:24:42    TOPS
12147                                                                                                                   27-Mar-1982 19:23:44    PA:<
12148 070142  022626                    91$:    CMP     (SP)+,(SP)+                     ;
12149 070144  005204                    92$:    INC     R4                              ; SECTOR                                                6552
12150 070146  020401                    93$:    CMP     R4,R1                           ; SECTOR,*                                              6548
12151 070150  101002                            BHI     94$
12152 070152  000167  177366                      JMP     85$
12153 070156  005266  000014              94$:    INC     14(SP)
12154 070162  026666  000014  000042    95$:    CMP     14(SP),42(SP)                   ; LUN                                                   5990
12155 070170  002002                            BGE     96$                             ; LUN,*
12156 070172  000167  172402                      JMP     1$
12157 070176  062706  000044              96$:    ADD     #44,SP                          ;                                                       5908
12158 070202  000207                            RTS     PC
12159
12160                                   ; Routine Size:  1456 words
12161                                   ; Maximum stack depth per invocation:  48 words
12166
12167
```

SEQ 0273

27-Mar-1982 19:24:42 TOPS-20 Bliss-16 V2(212)
27-Mar-1982 19:23:44 PA:<NEALE>MLX4.BLI.5 (41)

```
12169 ;MLX4
12170 ;                 SELECTING A RANDOM WORD COUNT
12171 ;
12172 ;          6633  %sbttl 'SELECTING A RANDOM WORD COUNT'
12173 ;          6634  routine RNDWC =
12174 ;          6635      begin
12175 ;          6636
12176 ;          6637  !++
12177 ;          6638  !  ROUTINE:     RNDWC
12178 ;          6639  !
12179 ;          6640  !  PURPOSE:     TO SELECT A RANDOM WORD COUNT WITHIN THE RANGE
12180 ;          6641  !               1 TO BUFSIZ.
12181 ;          6642  !
12182 ;          6643  !  RESULT:      THE VALUE RETURNED WILL BE USED BY THE CALLER
12183 ;          6644  !               AS ITS 'WRDCNT'
12184 ;          6645  !--
12185 ;          6646
12186 ;          6647      local
12187 ;          6648          WRDCNT;
12188 ;          6649
12189 ;          6650      RN ();
12190 ;          6651      RANDOM = .RANDOM and %o'077777';          !IGNORE SIGN BIT
12191 ;          6652      WRDCNT = ((.RANDOM mod BUFSIZ) + 1);      !FORCE THE RANGE
12192 ;          6653      return .WRDCNT;
12193 ;          6654      end;
12197
12198
                                        .SBTTL  RNDWC SELECTING A RANDOM WORD COUNT
12202 070204  004767  115160          RNDWC:  JSR     PC,RN                           ;          6650
12203 070210  042767  100000  115246          BIC     #100000,RANDOM                  ;          6651
12204 070216  016746  115242                  MOV     RANDOM,-(SP)                    ;          6652
12205 070222  012746  004000                  MOV     #4000,-(SP)
12206 070226  004767  114750                  JSR     PC,BL$MOD
12207 070232  005200                          INC     R0
12208 070234  022626                          CMP     (SP)+,(SP)+                     ;          6634
12209 070236  000207                          RTS     PC
12210
12211                                   ; Routine Size:  14 words
12212                                   ; Maximum stack depth per invocation:  2 words
12217
12218
```

```
12220 ;MLX4
12221 ;
12222
12223 ;          6655  %sbttl 'SELECTING A RANDOM SECTOR'
12224 ;          6656  routine RNDSEC (LUN) =
12225 ;          6657      begin
12226 ;          6658
12227 ;          6659  !++
12228 ;          6660  !  ROUTINE:     RNDSEC(LUN)
12229 ;          6661  !
12230 ;          6662  !  PURPOSE:     TO SELECT A RANDOM SECTOR NUMBER WITHIN THE RANGE
12231 ;          6663  !               OF TESTABLE SECTORS (LOWEST TO HIGHEST)
12232 ;          6664  !
12233 ;          6665  !  ARGUMENT:    LUN = THE CURRENT LOGICAL UNIT
12234 ;          6666  !               NOTE: 'LUN' IS REQUIRED TO CALCULATE LOWEST/HIGHEST
12235 ;          6667  !               FOR THE PARTICULAR LOGICAL UNIT.
12236 ;          6668  !
12237 ;          6669  !  RESULT:      THE VALUE RETURNED WILL BE USED BY THE CALLER
12238 ;          6670  !               AS ITS 'SECTOR'.
12239 ;          6671  !--
12240 ;          6672
12241 ;            73      local
12242 ;          6674          SECTOR,
12243 ;          6675          SIZE;
12244 ;          6676
12245 ;          6677  !
12246 ;          6678  ! version czmlbb changed eql to eqlu
12247 ;          6679  !
12248 ;          6680      if LOWEST eqlu HIGHEST
12249 ;          6681      then
12250 ;          6682          SECTOR = LOWEST
12251 ;          6683      else
12252 ;          6684          begin
12253 ;          6685          RN ();
12254 ;          6686          RANDOM = .RANDOM and %o'077777';        !IGNORE THE SIGN BIT
12255 ;          6687  !
12256 ;          6688  ! version czmlbb
12257 ;          6689  !
12258 ;          6690  ! 'Highest - Lowest + 1' when testing an ML-11B with
12259 ;          6691  ! 16 array modules results in a negitive number which
12260 ;          6692  ! causes the Bliss operator 'mod' to hang the CPU.
12261 ;          6693  ! Due to this malfunction the '+ 1' has been deleted
12262 ;          6694  ! which will avoid this malfunction
12263 ;          6695  !
12264 ;          6696          SIZE = HIGHEST - LOWEST;                        !FIND THE SECTOR RANGE
12265 ;          6697          SECTOR = (LOWEST + (.RANDOM mod .SIZE));        !FORCE RANGE
12266 ;          6698          end;
12267 ;          6699
12268 ;          6700      return .SECTOR;
12269 ;          6701      end;
12273
12274
                                    .SBTTL  RNDSEC SELECTING A RANDOM SECTOR
```

```
12276                                   ;MLX4
12277                                   ;                       SELECTING A RANDOM SECTOR                       27-Mar-1982 19:24:42    TOPS
12278                                                                                                           27-Mar-1982 19:23:44    PA:<
12282 070240  004167  115032            RNDSEC: JSR     R1,$SAVE3
12283 070244  016601  000012                    MOV     12(SP),R1       ;                                                               6656
12284 070250  006301                            ASL     R1              ; LUN,*                                                         6680
12285 070252  012703  034460                    MOV     #LOW.SECT,R3
12286 070256  060103                            ADD     R1,R3
12287 070260  021361  C34500                    CMP     (R3),TOP.SECT(R1)
12288 070264  001002                            BNE     1$
12289 070266  011302                            MOV     (R3),R2         ;                                                               6682
12290 070270  000420                            BR      2$              ; *,SECTOR                                                      6680
12291 070272  004767  115072            1$:     JSR     PC,RN           ;                                                               6685
12292 070276  042767  100000  115160            BIC     #100000,RANDOM  ;                                                               6686
12293 070304  016101  034500                    MOV     TOP.SECT(R1),R1 ;                                                               6696
12294 070310  161301                            SUB     (R3),R1         ; *,SIZE
12295 070312  016746  115146                    MOV     RANDOM,-(SP)    ; *,SIZE
12296 070316  010146                            MOV     R1,-(SP)        ;                                                               6697
12297 070320  004767  114656                    JSR     PC,BL$MOD       ; SIZE,*
12298 070324  061300                            ADD     (R3),R0
12299 070326  010002                            MOV     R0,R2
12300 070330  022626                            CMP     (SP)+,(SP)+     ; *,SECTOR
12301 070332  010200                            MOV     R2,R0           ;                                                               6684
12302 070334  000207            2$:     RTS     PC              ; SECTOR,*                                                      6657
12303                                                           ;                                                               6656
12304
12305                                   ; Routine Size:  31 words
12310                                   ; Maximum stack depth per invocation:  6 words
12311
```

```
12313 ;MLX4
12314 ;                 SELECTING A RANDOM UNIT                          27-Mar-1982 19:24:42    TOPS-20 Bliss-16 V2(212)
12315 ;                                                                  27-Mar-1982 19:23:44    PA:<NEALE>MLX4.BLI.5 (43)
12316 ;          6702  %sbttl 'SELECTING A RANDOM UNIT'
12317 ;          6703  routine RNDU =
12318 ;          6704      begin                                          !* 1 *
12319 ;          6705
12320 ;          6706  !++
12321 ;          6707  !  ROUTINE:     RNDU
12322 ;          6708  !
12323 ;          6709  !  PURPOSE:     TO SELECT A RANDOM LOGICAL UNIT NUMBER WITHIN
12324 ;          6710  !               THE RANGE OF TESTABLE UNITS (0 TO .L$UNIT-1)
12325 ;          6711  !--
12326 ;          6712
12327 ;          6713      local
12328 ;          6714          LUN;
12329 ;          6715
12330 ;          6716      if .L$UNIT eql 1
12331 ;          6717      then
12332 ;          6718          LUN = 0
12333 ;          6719      else
12334 ;          6720          begin                                      !* 2 *
12335 ;          6721          RN ();
12336 ;          6722          RANDOM = .RANDOM and %o'077777';           !IGNORE THE SIGN BIT
12337 ;          6723          RANDOM = (.RANDOM mod .L$UNIT);            !FORCE THE RANGE
12338 ;          6724
12339 ;          6725          !+
12340 ;          6726          ! MAKE SURE THE DRIVE IS ACTIVE.  IF IT ISN'T,
12341 ;          6727          ! THEN FIND THE NEXT AVAILABLE ACTIVE DRIVE:
12342 ;          6728          !-
12343 ;          6729
12344 ;          6730          incr COUNT from 0 to (.L$UNIT - 1) do
12345 ;          6731
12346 ;          6732              if .DRIVE_STATUS [.RANDOM] eql ACTIVE
12347 ;          6733              then
12348 ;          6734                  begin
12349 ;          6735                  LUN = .RANDOM;
12350 ;          6736                  exitloop;
12351 ;          6737                  end
12352 ;          6738              else
12353 ;          6739                  RANDOM = ((.RANDOM + 1) mod .L$UNIT);
12354 ;          6740
12355 ;          6741          end;                                       !* 2 *
12356 ;          6742
12357 ;          6743      return .LUN;
12358 ;          6744      end;                                           !* 1 *
12362
12363
12367 070336  004167  114752          RNDU:   .SBTTL  RNDU SELECTING A RANDOM UNIT
                                              JSR     R1,$SAVE4                    ;                                  6703
```

```
                                                                                                                    27-Mar-1982 19:24:42    TOPS
                                                                                                                    27-Mar-1982 19:23:44    PA:<
12369                                                   ;MLX4
12370                                                   ;
12371                                                   ;       SELECTING A RANDOM UNIT
12372 070342  026727  111444  000001                    CMP     L$UNIT,#1                       ;                                       6716
12373 070350  001002                                    BNE     1$
12374 070352  005004                                    CLR     R4                              ; LUN                                   6718
12375 070354  000467                                    BR      6$                              ;                                       6716
12376 070356  004767  115006                  1$:       JSR     PC,RN                           ;                                       6721
12377 070362  042767  100000  115074                    BIC     #100000,RANDOM                  ;                                       6722
12378 070370  016746  115070                            MOV     RANDOM,-(SP)                    ;                                       6723
12379 070374  016746  111412                            MOV     L$UNIT,-(SP)                    ;
12380 070400  004767  114576                            JSR     PC,BL$MOD
12381 070404  010067  115054                            MOV     R0,RANDOM
12382 070410  016703  111376                            MOV     L$UNIT,R3
12383 070414  005001                                    CLR     R1                              ;                                       6730
12384 070416  000443                                    BR      4$                              ; COUNT
12385 070420  016702  115040                  2$:       MOV     RANDOM,R2                       ;                                       6732
12386 070424  006202                                    ASR     R2
12387 070426  006202                                    ASR     R2
12388 070430  006202                                    ASR     R2
12389 070432  062702  034442                            ADD     #DRIVE.STATUS,R2
12390 070436  010246                                    MOV     R2,-(SP)
12391 070440  016746  115020                            MOV     RANDOM,-(SP)
12392 070444  042716  177770                            BIC     #177770,(SP)
12393 070450  012746  000001                            MOV     #1,-(SP)
12394 070454  005046                                    CLR     -(SP)
12395 070456  004767  113674                            JSR     PC,BL$GT2
12396 070462  062706  000010                            ADD     #10,SP
12397 070466  005300                                    DEC     R0
12398 070470  001003                                    BNE     3$
12399 070472  016704  114766                            MOV     RANDOM,R4                       ;                                       6735
12400 070476  000415                                    BR      5$                              ; *,LUN                                 6736
12401 070500  016746  114760                  3$:       MOV     RANDOM,-(SP)                    ;                                       6739
12402 070504  005216                                    INC     (SP)
12403 070506  016746  111300                            MOV     L$UNIT,-(SP)
12404 070512  004767  114464                            JSR     PC,BL$MOD
12405 070516  010067  114742                            MOV     R0,RANDOM
12406 070522  022626                                    CMP     (SP)+,(SP)+
12407 070524  005201                                    INC     R1                              ; COUNT
12408 070526  020103                  4$:       CMP     R1,R3                           ; COUNT,*                               6730
12409 070530  002733                                    BLT     2$
12410 070532  022626                  5$:       CMP     (SP)+,(SP)+                     ;                                       6720
12411 070534  010400                  6$:       MOV     R4,R0                           ; LUN,*                                 6704
12412 070536  000207                            RTS     PC                              ;                                       6703
12413
12414
12415                                   ; Routine Size:  65 words
12420                                   ; Maximum stack depth per invocation:  11 words
12421
```

SEQ 0278


```
12423 ;MLX4
12424 ;                 TESTING RANDOM DATA                                    !* 1 *  START OF ROUTINE
12425
12426 ;          6745  %sbttl 'TESTING RANDOM DATA'
12427 ;          6746  routine RAND1 (REPEAT) : novalue =
12428 ;          6747      begin
12429 ;          6748
12430 ;          6749  !++
12431 ;          6750  !  ROUTINE:      RAND1(REPEAT)
12432 ;          6751  !
12433 ;          6752  !  PURPOSE:      TO TEST USING RANDOM DATA
12434 ;          6753  !
12435 ;          6754  !  ARGUMENT:     REPEAT = NUMBER OF TIMES TO EXECUTE THIS ROUTINE
12436 ;          6755  !                         BEFORE RETURNING TO THE CALLER (OPT5).
12437 ;          6756  !
12438 ;          6757  !      NOTE:     THIS TEST CODE FOLLOWS THE SAME FLOW AS OPT3,
12439 ;          6758  !                BUT USES A RANDOM DATA PATTERN.
12440 ;          6759  !
12441 ;          6760  !  THE CODE FOR 'RAND1' IN BRIEF:
12442 ;          6761  !
12443 ;          6762  !  BEGIN 1  (START OF ROUTINE)
12444 ;          6763  !  SAY ROUTINE IS RUNNING
12445 ;          6764  !  INCR COUNT FROM 1 TO REPEAT
12446 ;          6765  !  : BEGIN 2 (START OF REPEAT LOOP FOR THE ROUTINE)
12447 ;          6766  !  : GENERATE THE RANDOM PATTERN
12448 ;          6767  !  : INCR LUN FROM 0 TO LAST
12449 ;          6768  !  : : BEGIN 3 (START OF LOGICAL UNIT SELECTION LOOP)
12450 ;          6769  !  : : TESTLOOP:
12451 ;          6770  !  : : : BEGIN 4 (START OF LOOP THAT COMPLETELY TESTS 1 UNIT)
12452 ;          6771  !  : : : IF UNIT IS ACTIVE
12453 ;          6772  !  : : : THEN
12454 ;          6773  !  : : : : BEGIN 5 (START OF TEST FOR AN ACTIVE UNIT)
12455 ;          6774  !  : : : : INITIALIZE THE WRITE AND READ BUFFER POINTERS
12456 ;          6775  !  : : : : SECTOR = LOWEST
12457 ;          6776  !  : : : : WHILE SECTOR LEQ HIGHEST DO
12458 ;          6777  !  : : : : : BEGIN 6 (START OF SECTOR SELECTION LOOP)
12459 ;          6778  !  : : : : : GET_WRDCNT
12460 ;          6779  !  : : : : : SET_UP BUFFER POINTERS BEFORE TRANSFER
12461 ;          6780  !  : : : : : WRITE
12462 ;          6781  !  : : : : : LOOK FOR ERRORS (IF DROP UNIT, LEAVE TESTLOOP)
12463 ;          6782  !  : : : : : CHOOSE WHETHER TO WRITE CHECK OR READ
12464 ;          6783  !  : : : : : DO THE WRITE CHECK OR READ
12465 ;          6784  !  : : : : : LOOK FOR ERRORS (IF DROP UNIT, LEAVE TESTLOOP)
12466 ;          6785  !  : : : : : CHANGE BUFFER POINTERS AFTER TRANSFER
12467 ;          6786  !  : : : : : UPDATE SECTOR NUMBER BY NUMBER OF SECTORS IN PREVIOUS TRANSFER
12468 ;          6787  !  : : : : : END 6 (END OF SECTOR SELECTION LOOP)
12469 ;          6788  !  : : : : END 5 (END OF TEST FOR AN ACTIVE UNIT)
12470 ;          6789  !  : : : END 4 (END OF TESTLOOP)
12471 ;          6790  !  : : END 3 (END OF LOGICAL UNIT SELECTION LOOP)
12472 ;          6791  !  : END 2 (END OF REPEAT LOOP FOR THIS ROUTINE)
12473 ;          6792  !  RETURN
12474 ;          6793  !  END 1 (END OF ROUTINE)
12475 ;          6794  !--
12476 ;          6795
12477 ;          6796      label
```

```
12479 ;MLX4
12480 ;                 TESTING RANDOM DATA
12481
12482 ;          6797           LOOP;
12483 ;          6798
12484 ;          6799       local
12485 ;          6800           WRDCNT,
12486 ;          6801           VALUE,
12487 ;          6802           OLDSEC,
12488 ;          6803           OLDCHN,
12489 ;          6804           SECTOR,
12490 ;          6805           PTR,
12491 ;          6806           COMMAND,
12492 ;          6807           DBL_VALUE;
12493 ;          6808
12494 ;          6809       PRINTB (SAY1, RTN5A);
12495 ;          6810       !'RAND1'
12496 ;          6811
12497 ;          6812       incr COUNT from 1 to .REPEAT do
12498 ;          6813           begin
12499 ;          6814           GEN5 ();                                  !* 2 *  START OF REPEAT LOOP FOR THIS ROUTINE
12500 ;          6815                                                     !FIRST 3 WORDS OF WBUFF ARE THE SEEDS
125C1 ;          6816           incr LUN from 0 to (.L$UNIT - 1) do
12502 ;          6817               begin
12503 ;          6818   LOOP :                                            !* 3 *  START OF LOGICAL UNIT SELECTION LOOP
12504 ;          6819               begin
12505 ;          6820                                                     !* 4 *  START OF THE LOOP THAT COMPLETELY TESTS 1 UNIT
12506 ;          6821               if .DRIVE_STATUS [.LUN] eql ACTIVE
12507 ;          6822               then
12508 ;          6823                   begin
12509 ;          6824                   L$LUN = .LUN;                     !* 5 *  START OF TEST FOR AN ACTIVE UNIT
12510 ;          6825                   WPTR = WBUFF;
12511 ;          6826                   RPTR = RBUFF;
12512 ;          6827                   SECTOR = LOWEST;
12513 ;          6828
12514 ;          6829                   while .SECTOR lequ HIGHEST do
12515 ;          6830                       begin                         !* 6 *  START OF SECTOR SELECTION LOOP
12516 ;          6831                       WRDCNT = GET_WRDCNT (.SECTOR, HIGHEST);
12517 ;          6832                       SET_PTRS (.WRDCNT);
12518 ;          6833                       VALUE = write (.LUN, .WRDCNT, .WPTR, .SECTOR);
12519 ;          6834
12520 ;          6835                       !+
12521 ;          6836                       !  SEE HOW SUCCESSFUL THE WRITE WAS:
12522 ;          6837                       !-
12523 ;          6838
12524 ;          6839                       selectone .VALUE of           !SEE 'SYSERR' FOR DEFINITION
12525 ;          6840                           set                       !OF ERROR # CONTAINED IN 'VALUE'
12526 ;          6841
12527 ;          6842                           [1] :
12528 ;          6843                               begin                 !* 6A * RETRY ALLOWED
12529 ;          6844
12530 ;          6845                               if RETRY (SIX, write, .LUN, .WRDCNT, .WPTR, .SECTOR) neq 0
12531 ;          6846                               then                  !THE RETRY FAILED -- SYSTEM FATAL ERROR
12532 ;          6847                                   begin
12533 ;          6848                                   WHY_DROPT [.LUN] = CODE_4;
```

```
12535 ;MLX4
12536 ;
12537
12538 ;          6849                                    ERRDF (5101, MSG1, 0);  !**** OPTION 5, RAND1 ERROR 01 ****
12539 ;          6850                                    DODU (.LUN);
12540 ;          6851                                    leave LOOP;     !JUMP JUST BEYOND END OF BLOCK * 4 *
12541 ;          6852                                    end;
12542 ;          6853
12543 ;          6854                                end;                !* 6A *
12544 ;          6855
12545 ;          6856                            [2] :
12546 ;          6857                                begin               !* 6B * FATAL CONTROLLER ERROR -- NO RETRY ALLOWED
12547 ;          6858                                WHY_DROPT [.LUN] = CODE_5;
12548 ;          6859                                ERRDF (5102, MSG1, 0);      !**** OPTION 5, RAND1 ERROR 02 ****
12549 ;          6860                                DODU (.LUN);
12550 ;          6861                                leave LOOP;         !JUMP JUST BEYOND END OF BLOCK * 4 *
12551 ;          6862                                end;                !* 6B *
12552 ;          6863
12553 ;          6864                            [3] :
12554 ;          6865                                begin               !* 6C * FATAL DRIVE ERROR -- NO RETRY ALLOWED
12555 ;          6866                                ERRDF (5103, MSG1, 0);      !**** OPTION 5, RAND1 ERROR 03 ****
12556 ;          6867                                WHY_DROPT [.LUN] = CODE_6;
12557 ;          6868                                DODU (.LUN);
12558 ;          6869                                leave LOOP;         !JUMP JUST BEYOND END OF BLOCK * 4 *
12559 ;          6870                                end;                !* 6C *
12560 ;          6871                            tes;
12561 ;          6872
12562 ;          6873                        COMMAND = CHOOSE ();
12563 ;          6874
12564 ;          6875                        if .COMMAND eql read
12565 ;          6876                        then
12566 ;          6877                            begin
12567 ;          6878                            PTR = .RPTR;
12568 ;          6879                            VALUE = read (.LUN, .WRDCNT, .RPTR, .SECTOR);
12569 ;          6880                            end
12570 ;          6881                        else
12571 ;          6882                            begin
12572 ;          6883                            PTR = .WPTR;
12573 ;          6884                            VALUE = CHECK (.LUN, .WRDCNT, .WPTR, .SECTOR);
12574 ;          6885                            end;
12575 ;          6886
12576 ;          6887                        !+
12577 ;          6888                        !  SEE HOW SUCCESSFUL THE OPERATION WAS:
12578 ;          6889                        !-
12579 ;          6890
12580 ;          6891                        selectone .VALUE of         !SEE 'SYSERR' FOR DEFINITION
12581 ;          6892                            set                     !OF ERROR # CONTAINED IN 'VALUE'
12582 ;          6893
12583 ;          6894                            [0] :
12584 ;          6895
12585 ;          6896                                if .COMMAND eql read
12586 ;          6897                                then
12587 ;          6898                                    begin
12588 ;          6899
12589 ;          6900                                    if (DBL_VALUE = DOUBLE_CHECK (.WPTR, .RPTR, .WRDCNT)) neq 0
```

27-Mar-1982 19:24:42 TOPS-20 Bliss-16 V2(212)
27-Mar-1982 19:23:44 PA:<NEALE>MLX4.BLI.5 (44)

```
12591 ;MLX4
12592 ;                 TESTING RANDOM DATA
12593
12594 ;          6901                                   then
12595 ;          6902                                       begin
12596 ;          6903                                       SAYWHO (.LUN);
12597 ;          6904                                       PRINTB (SAY1, MSG5);          !'ECC LOGIC FAILED TO DETECT DATA ERROR'
12598 ;          6905                                       PRINTB (FMT12A, ..DBL_VALUE, .DBL_VALUE);
12599 ;          6906                                                                     !'GOOD DATA: XXXXXX AT LOCATION YYYYYY'
12600 ;          6907                                       DBL_VALUE = .DBL_VALUE + BUFSIZ*2;
12601 ;          6908                                       PRINTB (FMT12B, ..DBL_VALUE, .DBL_VALUE);
12602 ;          6909                                                                     !'BAD DATA:  PPPPPP AT LOCATION QQQQQQ'
12603 ;          6910                                       WHY_DROPT [.LUN] = CODE_8;
12604 ;          6911                                       ERRDF (5104, MSG1, 0);        !**** OPTION 5, RAND1 ERROR 04 ****
12605 ;          6912                                       DODU (.LUN);
12606 ;          6913                                       leave LOOP; !JUMP JUST BEYOND END OF BLOCK * 4 *
12607 ;          6914                                       end;
12608 ;          6915
12609 ;          6916                                   end;
12610 ;          6917
12611 ;          6918                               [1] :
12612 ;          6919                                   begin                 !* 6D * RETRY ALLOWED
12613 ;          6920
12614 ;          6921                                   if RETRY (SIX, .COMMAND, .LUN, .WRDCNT, .PTR, .SECTOR) neq 0
12615 ;          6922                                   then                  !THE RETRY FAILED -- SYSTEM FATAL ERROR
12616 ;          6923                                       begin
12617 ;          6924                                       WHY_DROPT [.LUN] = CODE_4;
12618 ;          6925                                       ERRDF (5105, MSG1, 0);  !**** OPTION 5, RAND1 ERROR 05 ****
12619 ;          6926                                       DODU (.LUN);
12620 ;          6927                                       leave LOOP;     !JUMP JUST BEYOND END OF BLOCK * 4 *
12621 ;          6928                                       end;
12622 ;          6929
12623 ;          6930                                   end;                  !* 6D *
12624 ;          6931
12625 ;          6932                               [2] :
12626 ;          6933                                   begin                 !* 6E * FATAL CONTROLLER ERROR -- NO RETRY ALLOWED
12627 ;          6934                                   WHY_DROPT [.LUN] = CODE_5;
12628 ;          6935                                   ERRDF (5106, MSG1, 0);        !**** OPTION 5, RAND1 ERROR 06 ****
12629 ;          6936                                   DODU (.LUN);
12630 ;          6937                                   leave LOOP;           !JUMP JUST BEYOND END OF BLOCK * 4 *
12631 ;          6938                                   end;                  !* 6E *
12632 ;          6939
12633 ;          6940                               [3] :
12634 ;          6941                                   begin                 !* 6F * FATAL DRIVE ERROR -- NO RETRY ALLOWED
12635 ;          6942                                   ERRDF (5107, MSG1, 0);        !**** OPTION 5, RAND1 ERROR 07 ****
12636 ;          6943                                   WHY_DROPT [.LUN] = CODE_6;
12637 ;          6944                                   DODU (.LUN);
12638 ;          6945                                   leave LOOP;           !JUMP JUST BEYOND END OF BLOCK * 4 *
12639 ;          6946                                   end;                  !* 6F *
12640 ;          6947
12641 ;          6948                               [4] :
12642 ;          6949                                   begin                 !* 6G * UNRECOVERABLE DATA ERROR
12643 ;          6950                                   ISOLATE ();
12644 ;          6951                                   ERRDF (5108, MSG2, 0);        !**** OPTION 5, RAND1 ERROR 08 ****
12645 ;          6952                                   WHY_DROPT [.LUN] = CODE_7;
```

```
12647 ;MLX4
12648 ;                 TESTING RANDOM DATA
12649
12650 ;          6953                                DODU (.LUN);
12651 ;          6954                                leave LOOP;             !JUMP JUST BEYOND END OF BLOCK * 4 *
12652 ;          6955                                end;                    !* 6G *
12653 ;          6956
12654 ;          6957                            [5] :
12655 ;          6958                                begin                   !* 6H * RECOVERABLE DATA ERROR
12656 ;          6959                                ISOLATE ();
12657 ;          6960
12658 ;          6961                                if .ERROUT then PRINTB (FMT10B, .CHAN);
12659 ;          6962
12660 ;          6963                                !'    BIT QQ'
12661 ;          6964                                OLDSEC = .MLEL;
12662 ;          6965                                OLDCHN = .CHAN;
12663 ;          6966
12664 ;          6967                                if RETRY (ONE, .COMMAND, .LUN, .WRDCNT, .PTR, .SECTOR) eql 5
12665 ;          6968                                then
12666 ;          6969
12667 ;          6970                                    if ((.MLEL eql .OLDSEC) and (.CHAN eql .OLDCHN))
12668 ;          6971                                    then
12669 ;          6972                                        begin
12670 ;          6973
12671 ;          6974                                        if .ERROUT then ERRHRD (5109, MSG4, 0);
12672 ;          6975
12673 ;          6976                                                        !**** OPTION 5, RAND1 ERROR 09 ****
12674 ;          6977                                        UP_HARD_COUNT (.LUN, .BOARD);
12675 ;          6978                                        end
12676 ;          6979                                    else
12677 ;          6980                                        begin
12678 ;          6981
12679 ;          6982                                        if .ERROUT then ERRSOFT (5110, MSG3, 0);
12680 ;          6983
12681 ;          6984                                                        !**** OPTION 5, RAND1 ERROR 10 ****
12682 ;          6985                                        UP_SOFT_COUNT (.LUN, .BOARD);
12683 ;          6986                                        end
12684 ;          6987
12685 ;          6988                                else
12686 ;          6989                                    begin
12687 ;          6990
12688 ;          6991                                    if .ERROUT then ERRSOFT (5111, MSG3, 0);
12689 ;          6992
12690 ;          6993                                                        !**** OPTION 5, RAND1 ERROR 11 ****
12691 ;          6994                                    UP_SOFT_COUNT (.LUN, .BOARD);
12692 ;          6995                                    end;
12693 ;          6996
12o94 ;          6997                                end;                    !* 6H *
12695 ;          6998                            tes;
12696 ;          6999
12697 ;          7000                        WPTR = .WPTR + (.WRDCNT*2);
12698 ;          7001                        SECTOR = .SECTOR + (.WRDCNT/256);
12699 ;          7002                        end;                            !* 6 *  END OF SECTOR SELECTION LOOP
12700 ;          7003
12701 ;          7004                end;                                    !* 5 *  END OF TEST FOR AN ACTIVE UNIT
```

SEQ 0283

27-Mar-1982 19:24:42 TOPS-20 Bliss-16 V2(212)
27-Mar-1982 19:23:44 PA:<NEALE>MLX4.BLI.5 (44)

```
12703 ;MLX4
12704 ;                 TESTING RANDOM DATA
12705
12706 ;          7005
12707 ;          7006               !+
12708 ;          7007               ! Test to see if this uut's address space is
12709 ;          7008               ! to be read for soft errors.  This test is
12710 ;          7009               ! is intended for DMT purposes.
12711 ;          7010               !-
12712 ;          7011
12713 ;          7012               if .EFN$21                          !Is the background pattern to be read
12714 ;          7013               then
12715 ;          7014                   begin
12716 ;          7015
12717 ;          7016 !
12718 ;          7017 ! version czmlbb changed incr to incru
12719 ;          7018 !
12720 ;          7019                   incru SECTOR from LOWEST to HIGHEST do
12721 ;          7020
12722 ;          7021                       if read (.LUN, 256, RBUFF, .SECTOR) eql 5
12723 ;          7022                       then
12724 ;          7023                           begin
12725 ;          7024                           ISOLATE ();             !Find the failing bank and board no.
12726 ;          7025
12727 ;          7026                           if .ERROUT then PRINTB (FMT10B, .CHAN);
12728 ;          7027
12729 ;          7028                           ! Print where the error is
12730 ;          7029                           !
12731 ;          7030                           ! Save the contents of the ML error location
12732 ;          7031                           ! register so we can compare it to the new
12733 ;          7032                           ! contents of this register after the retry.
12734 ;          7033                           ! This is done to classify the error.
12735 ;          7034                           !
12736 ;          7035                           OLDSEC = .MLEL;
12737 ;          7036                           OLDCHN = .CHAN;
12738 ;          7037                           !
12739 ;          7038                           ! Do a classify retry call.  If the same error
12740 ;          7039                           ! occures then classify it as a hard error.  If
12741 ;          7040                           ! a different error occured or the error went away
12742 ;          7041                           ! then classify it as a soft error.
12743 ;          7042                           !
12744 ;          7043
12745 ;          7044                           if RETRY (ONE, .COMMAND, .LUN, 256, .PTR, .OLDSEC) eql 5
12746 ;          7045                           then
12747 ;          7046                           !
12748 ;          7047                           ! The same error occured so see if it is at the same
12749 ;          7048                           ! sector and channel number, if so then classify
12750 ;          7049                           ! it as a hard error else classiy it as a soft
12751 ;          7050                           ! error.
12752 ;          7051                           !
12753 ;          7052
12754 ;          7053                               if ((.MLEL eql .OLDSEC) and (.CHAN eql .OLDCHN))
12755 ;          7054                               then
12756 ;          7055                                   begin           !Same error occured 'hard'
12757 ;          7056
```

```
12759 ;MLX4
12760 ;                 TESTING RANDOM DATA
12761
12762 ;          7057                                   if .ERROUT       !Print error if enabled
12763 ;          7058                                   then
12764 ;          7059                                       begin
12765 ;        P 7060                                       ERRHRD (5112,        !Error number
12766 ;        P 7061                                           MSG4,    !Error message
12767 ;          7062                                           0);      !Additional message routine
12768 ;          7063                                       end;
12769 ;          7064
12770 ;          7065                                   UP_HARD_COUNT (.LUN, .BOARD);
12771 ;          7066                                   end
12772 ;          7067                               else
12773 ;          7068                                   begin            !Not the same error 'soft'
12774 ;          7069
12775 ;          7070                                   if .ERROUT       !Print error if enabled
12776 ;          7071                                   then
12777 ;          7072                                       begin
12778 ;        P 7073                                       ERRSOFT (5113,       !Error number
12779 ;        P 7074                                           MSG3,    !Error message
12780 ;          7075                                           0);      !Additional message routine
12781 ;          7076                                       end;
12782 ;          7077
12783 ;          7078                                   UP_SOFT_COUNT (.LUN, .BOARD);
12784 ;          7079                                   end
12785 ;          7080
12786 ;          7081                           else                     !Not the same error 'soft'
12787 ;          7082                               begin
12788 ;          7083
12789 ;          7084                               if .ERROUT           !Print error if enabled
12790 ;          7085                               then
12791 ;          7086                                   begin
12792 ;        P 7087                                   ERRSOFT (5114,   !Error number
12793 ;        P 7088                                       MSG3,        !Error message
12794 ;          7089                                       0);          !Additional message routine
12795 ;          7090                                   end;
12796 ;          7091
12797 ;          7092                               UP_SOFT_COUNT (.LUN, .BOARD);
12798 ;          7093                               end;
12799 ;          7094
12800 ;          7095                           end;
12801 ;          7096
12802 ;          7097                   end;
12803 ;          7098
12804 ;          7099               end;                                 !* 4 *  END OF LOOP THAT COMPLETELY TESTS 1 UNIT
12805 ;          7100               end;                                 !* 3 *  END OF LOGICAL UNIT SELECTION LOOP
12806 ;          7101
12807 ;          7102           end;                                     !* 2 *  END OF REPEAT LOOP FOR THIS ROUTINE
12808 ;          7103
12809 ;          7104       return;
12810 ;          7105       end;                                         !* 1 *  END OF ROUTINE
```

```
                                        :MLX4
                                        :                TESTING RANDOM DATA                                    27-Mar-1982 19:24:42   TOPS
                                                                                                                27-Mar-1982 19:23:44   PA:<

                                                 .SBTTL  RAND1 TESTING RANDOM DATA
070540  004167  114570          RAND1:  JSR     R1,$SAVE5                                                                              
070544  162706  000020                  SUB     #20,SP                  ;                                                              6746
070550  012746  007664                  MOV     #RTN5A,-(SP)                                                                           
070554  012746  007072                  MOV     #SAY1,-(SP)             ;                                                              6809
070560  012746  000002                  MOV     #2,-(SP)                                                                               
070564  010600                          MOV     SP,R0                   ; SP,*                                                         
070566  104414                          TRAP    14                                                                                     
070570  005066  000024                  CLR     24(SP)                  ; COUNT                                                        
070574  000167  002276                  JMP     42$                                                                                    6812
070600  004767  151014          1$:     JSR     PC,GEN5                 ;                                                              
070604  016766  111202  000010        MOV     L$UNIT,10(SP)           ;                                                              6814
070612  005066  000006                  CLR     6(SP)                   ;                                                              6816
070616  000167  002240                  JMP     41$                     ; LUN                                                          
070622  016600  000006          2$:     MOV     6(SP),R0                ; LUN,*                                                        
070626  006200                          ASR     R0                                                                                     6821
070630  006200                          ASR     R0                                                                                     
070632  006200                          ASR     R0                                                                                     
070634  062700  034442                  ADD     #DRIVE.STATUS,R0                                                                       
070640  010046                          MOV     R0,-(SP)                                                                               
070642  016646  000010                  MOV     10(SP),-(SP)            ; LUN,*                                                        
070646  042716  177770                  BIC     #177770,(SP)                                                                           
070652  012746  000001                  MOV     #1,-(SP)                                                                               
070656  005046                          CLR     -(SP)                                                                                  
070660  004767  113472                  JSR     PC,BL$GT2                                                                              
070664  062706  000010                  ADD     #10,SP                                                                                 
070670  005300                          DEC     R0                                                                                     
070672  001402                          BEQ     4$                                                                                     
070674  000167  001502          3$:     JMP     29$                                                                                    
070700  016667  000006  111166  4$:     MOV     6(SP),L$LUN             ; LUN,*                                                        
070706  012767  012670  141754          MOV     #WBUFF,WPTR             ;                                                              6824
070714  012767  022670  141750          MOV     #RBUFF,RPTR             ;                                                              6825
070722  016605  000006                  MOV     6(SP),R5                ;                                                              6826
070726  006305                          ASL     R5                      ; LUN,*                                                        6827
070730  016503  034460                  MOV     LOW.SECT(R5),R3         ; *,SECTOR                                                     
070734  020365  034500          5$:     CMP     R3,TOP.SECT(R5)         ; SECTOR,*                                                     
070740  101355                          BHI     3$                                                                                     6829
070742  010346                          MOV     R3,-(SP)                ; SECTOR,*                                                     
070744  016546  034500                  MOV     TOP.SECT(R5),-(SP)                                                                     6831
070750  004767  156222                  JSR     PC,GET.WRDCNT                                                                          
070754  010002                          MOV     R0,R2                   ; *,WRDCNT                                                     
070756  010216                          MOV     R2,(SP)                 ; WRDCNT,*                                                     
070760  004767  156146                  JSR     PC,SET.PTRS                                                                            6832
070764  016616  000012                  MOV     12(SP),(SP)             ; LUN,*                                                        
070770  010246                          MOV     R2,-(SP)                ; WRDCNT,*                                                     6833
070772  016746  141672                  MOV     WPTR,-(SP)                                                                             
070776  010346                          MOV     R3,-(SP)                ; SECTOR,*                                                     
071000  004767  154420                  JSR     PC,WRITE                                                                               
                                        :MLX4
                                        :                TESTING RANDOM DATA                                    27-Mar-1982 19:24:42   TOPS
                                                                                                                27-Mar-1982 19:23:44   PA:<

071004  010004                  MOV     R0,R4                   ; *,VALUE                                                      
071006  020427  000001          CMP     R4,#1                   ; VALUE,*                                                      
                                                                                                                                6839
```

```
12875 071012 001035                    BNE     6$
12876 071014 012746 000006             MOV     #6,-(SP)
12877 071020 012746 045424             MOV     #WRITE,-(SP)                                    ;                                       6845
12878 071024 016646 000024             MOV     24(SP),-(SP)
12879 071030 010246                    MOV     R2,-(SP)                                        ; LUN,*
12880 071032 016746 141632             MOV     WPTR,-(SP)                                      ; WRDCNT,*
12881 071036 010346                    MOV     R3,-(SP)
12882 071040 004767 155160             JSR     PC,RETRY                                        ; SECTOR,*
12883 071044 062706 000014             ADD     #14,SP
12884 071050 005700                    TST     R0
12885 071052 001456                    BEQ     9$
12886 071054 016601 000020             MOV     20(SP),R1
12887 071060 112761 000004 034446      MOVB    #4,WHY.DROPT(R1)                                ; LUN,*                                 6848
12888 071066 104455                    TRAP    55
12889 071070 011755                    .WORD   11755                                           ;                                       6849
12890 071072 011070                    .WORD   MSG1
12891 071074 000000                    .WORD   0
12892 071076 016600 000020             MOV     20(SP),R0
12893 071102 104451                    TRAP    51                                              ; LUN,*                                 6850
12894 071104 000436                    BR      8$
12895 071106 020427 000002      6$:    CMP     R4,#2                                           ;                                       6851
12896 071112 001015                    BNE     7$                                              ; VALUE,*                               6839
12897 071114 016601 000020             MOV     20(SP),R1
12898 071120 112761 000005 034446      MOVB    #5,WHY.DROPT(R1)                                ; LUN,*                                 6858
12899 071126 104455                    TRAP    55
12900 071130 011756                    .WORD   11756                                           ;                                       6859
12901 071132 011070                    .WORD   MSG1
12902 071134 000000                    .WORD   0
12903 071136 016600 000020             MOV     20(SP),R0
12904 071142 104451                    TRAP    51                                              ; LUN,*                                 6860
12905 071144 000416                    BR      8$
12906 071146 020427 000003      7$:    CMP     R4,#3                                           ;                                       6861
12907 071152 001016                    BNE     9$                                              ; VALUE,*                               6839
12908 071154 104455                    TRAP    55
12909 071156 011757                    .WORD   11757                                           ;                                       6866
12910 071160 011070                    .WORD   MSG1
12911 071162 000000                    .WORD   0
12912 071164 016601 000020             MOV     20(SP),R1
12913 071170 112761 000006 034446      MOVB    #6,WHY.DROPT(R1)                                ; LUN,*                                 6867
12914 071176 010100                    MOV     R1,R0
12915 071200 104451                    TRAP    51                                              ; LUN,*                                 6868
12916 071202 062706 000012      8$:    ADD     #12,SP
12917 071206 000543                    BR      14$                                             ;                                       6869
12918 071210 004767 154752      9$:    JSR     PC,CHOOSE                                       ;                                       6873
12919 071214 010066 000034             MOV     R0,34(SP)                                       ; *,COMMAND
12920 071220 005001                    CLR     R1                                              ;                                       6875
12921 071222 020027 045612             CMP     R0,#READ                                        ; COMMAND,*
12922 071226 001015                    BNE     10$
12923 071230 005201                    INC     R1
12924 071232 016766 141434 000032      MOV     RPTR,32(SP)                                     ; *,PTR                                 6878
12925                           ;MLX4                                                                                                  TOPS
12926                           ;       TESTING RANDOM DATA                                                     27-Mar-1982 19:24:42
12927                                                                                                           27-Mar-1982 19:23:44    PA:<
12928 071240 016646 000020             MOV     20(SP),-(SP)                                    ; LUN,*                                 6879
12929 071244 010246                    MOV     R2,-(SP)                                        ; WRDCNT,*
12930 071246 016746 141420             MOV     RPTR,-(SP)
12931 071252 010346                    MOV     R3,-(SP)                                        ; SECTOR,*
```

```
12932 071254 004767 154332                      JSR     PC,READ                                                 SEQ 0287
12933 071260 000413                             BR      11$
12934 071262 016766 141402 000032  10$:         MOV     WPTR,32(SP)             ; *,PTR                         6883
12935 071270 016646 000020                      MOV     20(SP),-(SP)            ; LUN,*                         6884
12936 071274 010246                             MOV     R2,-(SP)                ; WRDCNT,*
12937 071276 016746 141366                      MOV     WPTR,-(SP)
12938 071302 010346                             MOV     R3,-(SP)                ; SECTOR,*
12939 071304 004767 154470                      JSR     PC,CHECK
12940 071310 010004                 11$:        MOV     R0,R4                   ; *,VALUE
12941 071312 001107                             BNE     15$                     ;
12942 071314 006001                             ROR     R1                      ;                               6891
12943 071316 103402                             BLO     13$                                                     6896
12944 071320 000167 001014          12$:        JMP     28$
12945 071324 016746 141340          13$:        MOV     WPTR,-(SP)              ;                               6900
12946 071330 016746 141336                      MOV     RPTR,-(SP)
12947 071334 010246                             MOV     R2,-(SP)                ; WRDCNT,*
12948 071336 004767 152764                      JSR     PC,DOUBLE.CHECK
12949 071342 062706 000006                      ADD     #6,SP
12950 071346 010066 000036                      MOV     R0,36(SP)               ; *,DBL.VALUE
12951 071352 001762                             BEQ     12$
12952 071354 016646 000030                      MOV     30(SP),-(SP)            ; LUN,*                         6903
12953 071360 004767 144140                      JSR     PC,SAYWHO
12954 071364 012716 011216                      MOV     #MSG5,(SP)              ;                               6904
12955 071370 012746 007072                      MOV     #SAY1,-(SP)
12956 071374 012746 000002                      MOV     #2,-(SP)
12957 071400 010600                             MOV     SP,R0                   ; SP,*
12958 071402 104414                             TRAP    14
12959 071404 016616 000044                      MOV     44(SP),(SP)             ; DBL.VALUE,*                   6905
12960 071410 017646 000044                      MOV     @44(SP),-(SP)           ; DBL.VALUE,*
12961 071414 012746 006710                      MOV     #FMT12A,-(SP)
12962 071420 012746 000003                      MOV     #3,-(SP)
12963 071424 010600                             MOV     SP,R0                   ; SP,*
12964 071426 104414                             TRAP    14
12965 071430 062766 010000 000052               ADD     #10000,52(SP)           ; *,DBL.VALUE                   6907
12966 071436 016616 000052                      MOV     52(SP),(SP)             ; DBL.VALUE,*                   6908
12967 071442 017646 000052                      MOV     @52(SP),-(SP)           ; DBL.VALUE,*
12968 071446 012746 006756                      MOV     #FMT12B,-(SP)
12969 071452 012746 000003                      MOV     #3,-(SP)
12970 071456 010600                             MOV     SP,R0                   ; SP,*
12971 071460 104414                             TRAP    14
12972 071462 016601 000052                      MOV     52(SP),R1               ; LUN,*                         6910
12973 071466 112761 000010 034446               MOVB    #10,WHY.DROPT(R1)
12974 071474 104455                             TRAP    55                      ;                               6911
12975 071476 011760                             .WORD   11760
12976 071500 011070                             .WORD   MSG1
12977 071502 000000                             .WORD   0
12978 071504 016600 000052                      MOV     52(SP),R0               ; LUN,*
12979 071510 104451                             TRAP    51                                                      6912
12980                           ;MLX4
12981                           ;               TESTING RANDOM DATA                     27-Mar-1982 19:24:42    TOPS
12982                                                                                   27-Mar-1982 19:23:44    PA:<
12983 071512 062706 000044                      ADD     #44,SP
12984 071516 000521                 14$:        BR      20$                     ;                               6913
12985 071520 020427 000001          15$:        CMP     R4,#1                   ; VALUE,*                       6891
12986 071524 001035                             BNE     16$
12987 071526 012746 000006                      MOV     #6,-(SP)                ;
12988 071532 016646 000046                      MOV     46(SP),-(SP)            ; COMMAND,*                     6921
```

SEQ 0288

```
12989 071536  016646  000034                 MOV     34(SP),-(SP)          ; LUN,*
12990 071542  010246                         MOV     R2,-(SP)              ; WRDCNT,*
12991 071544  016646  000052                 MOV     52(SP),-(SP)          ; PTR,*
12992 071550  010346                         MOV     R3,-(SP)              ; SECTOR,*
12993 071552  004767  154446                 JSR     PC,RETRY
12994 071556  062706  000014                 ADD     #14,SP
12995 071562  005700                         TST     R0
12996 071564  001655                         BEQ     12$
12997 071566  016601  000030                 MOV     30(SP),R1
12998 071572  112761  000004  034446         MOVB    #4,WHY.DROPT(R1)      ; LUN,*                    6924
12999 071600  104455                         TRAP    55
13000 071602  011761                         .WORD   11761                 ;                          6925
13001 071604  011070                         .WORD   MSG1
13002 071606  000000                         .WORD   0
13003 071610  016600  000030                 MOV     30(SP),R0             ; LUN,*
13004 071614  104451                         TRAP    51                                               6926
13005 071616  000457                         BR      19$                   ;
13006 071620  020427  000002          16$:   CMP     R4,#2                 ; VALUE,*                  6927
13007 071624  001015                         BNE     17$                                              6891
13008 071626  016601  000030                 MOV     30(SP),R1
13009 071632  112761  000005  034446         MOVB    #5,WHY.DROPT(R1)      ; LUN,*                    6934
13010 071640  104455                         TRAP    55
13011 071642  011762                         .WORD   11762                 ;                          6935
13012 071644  011070                         .WORD   MSG1
13013 071646  000000                         .WORD   0
13014 071650  016600  000030                 MOV     30(SP),R0             ; LUN,*
13015 071654  104451                         TRAP    51                                               6936
13016 071656  000437                         BR      19$                   ;
13017 071660  020427  000003          17$:   CMP     R4,#3                 ; VALUE,*                  6937
13018 071664  001014                         BNE     18$                                              6891
13019 071666  104455                         TRAP    55
13020 071670  011763                         .WORD   11763                 ;                          6942
13021 071672  011070                         .WORD   MSG1
13022 071674  000000                         .WORD   0
13023 071676  016601  000030                 MOV     30(SP),R1
13024 071702  112761  000006  034446         MOVB    #6,WHY.DROPT(R1)      ; LUN,*                    6943
13025 071710  010100                         MOV     R1,R0
13026 071712  104451                         TRAP    51                    ; LUN,*                    6944
13027 071714  000420                         BR      19$                   ;
13028 071716  020427  000004          18$:   CMP     R4,#4                 ; VALUE,*                  6945
13029 071722  001021                         BNE     21$                                              6891
13030 071724  004767  145402                 JSR     PC,ISOLATE
13031 071730  104455                         TRAP    55                    ;                          6950
13032 071732  011764                         .WORD   11764                 ;                          6951
13033 071734  011120                         .WORD   MSG2
13034 071736  000000                         .WORD   0
13035                       ;MLX4
13036                       ;                                                    27-Mar-1982 19:24:42  TOPS
13037                                TESTING RANDOM DATA                         27-Mar-1982 19:23:44  PA:<
13038 071740  016601  000030                 MOV     30(SP),R1
13039 071744  112761  000007  034446         MOVB    #7,WHY.DROPT(R1)      ; LUN,*                    6952
13040 071752  010100                         MOV     R1,R0
13041 071754  104451                         TRAP    51                    ; LUN,*                    6953
13042 071756  062706  000022          19$:   ADD     #22,SP
13043 071762  000167  001070          20$:   JMP     40$                   ;                          6954
13044 071766  020427  000005          21$:   CMP     R4,#5                 ; VALUE,*
13045 071772  001162                         BNE     28$                                              6891
```

```
                                                                                                                    SEQ 0289
13046 071774  004767  145332                JSR     PC,ISOLATE
13047 072000  032767  000001  110252        BIT     #1,ERROUT       ;                                                 6959
13048 072006  001423                        BEQ     22$             ;                                                 6961
13049 072010  017701  142374                MOV     @ML.REG+42,R1
13050 072014  006201                        ASR     R1
13051 072016  006201                        ASR     R1
13052 072020  006201                        ASR     R1
13053 072022  006201                        ASR     R1
13054 072024  006201                        ASR     R1
13055 072026  006201                        ASR     R1
13056 072030  042701  177700                BIC     #177700,R1
13057 072034  010146                        MOV     R1,-(SP)
13058 072036  012746  006640                MOV     #FMT10B,-(SP)
13059 072042  012746  000002                MOV     #2,-(SP)
13060 072046  010600                        MOV     SP,R0
13061 072050  104414                        TRAP    14              ; SP,*
13062 072052  062706  000006                ADD     #6,SP
13063 072056  017766  142330  000040  22$:  MOV     @ML.REG+44,40(SP)  ; *,OLDSEC
13064 072064  017701  142320                MOV     @ML.REG+42,R1   ;                                                 6964
13065 072070  006201                        ASR     R1              ;                                                 6965
13066 072072  006201                        ASR     R1
13067 072074  006201                        ASR     R1
13068 072076  006201                        ASR     R1
13069 072100  006201                        ASR     R1
13070 072102  006201                        ASR     R1
13071 072104  042701  177700                BIC     #177700,R1
13072 072110  010166  000034                MOV     R1,34(SP)       ; *,OLDCHN
13073 072114  012746  000001                MOV     #1,-(SP)        ;
13074 072120  016646  000046                MOV     46(SP),-(SP)    ; COMMAND,*                                       6967
13075 072124  016646  000034                MOV     34(SP),-(SP)    ; LUN,*
13076 072130  010246                        MOV     R2,-(SP)        ; WRDCNT,*
13077 072132  016646  000052                MOV     52(SP),-(SP)    ; PTR,*
13078 072136  010346                        MOV     R3,-(SP)        ; SECTOR,*
13079 072140  004767  154060                JSR     PC,RETRY
13080 072144  062706  000014                ADD     #14,SP
13081 072150  020027  000005                CMP     R0,#5
13082 072154  001052                        BNE     25$
13083 072156  027766  142230  000040        CMP     @ML.REG+44,40(SP)  ; *,OLDSEC                                     6970
13084 072164  001035                        BNE     24$
13085 072166  016600  000034                MOV     34(SP),R0       ; OLDCHN,*
13086 072172  017701  142212                MOV     @ML.REG+42,R1
13087 072176  006201                        ASR     R1
13088 072200  006201                        ASR     R1
13089 072202  006201                        ASR     R1
13090
13091                               ;MLX4                                   27-Mar-1982 19:24:42    TOPS
13092                               ;       TESTING RANDOM DATA             27-Mar-1982 19:23:44    PA:<
13093 072204  006201                        ASR     R1
13094 072206  006201                        ASR     R1
13095 072210  006201                        ASR     R1
13096 072212  042701  177700                BIC     #177700,R1
13097 072216  020100                        CMP     R1,R0
13098 072220  001017                        BNE     24$
13099 072222  032767  000001  110030        BIT     #1,ERROUT       ;                                                 6974
13100 072230  001404                        BEQ     23$
13101 072232  104456                        TRAP    56
13102 072234  011765                        .WORD   11765
```

SEQ 0290

```
13103 072236  011202                        .WORD   MSG4
13104 072240  000000                        .WORD   0
13105 072242  016646  000030         23$:   MOV     30(SP),-(SP)          ; LUN,*                    6977
13106 072246  016746  142070               MOV     BOARD,-(SP)
13107 072252  004767  145144               JSR     PC,UP.HARD.COUNT
13108 072256  000427                        BR      27$
13109 072260  032767  000001  107772 24$:   BIT     #1,ERROUT             ;                          6970
13110 072266  001415                        BEQ     26$                   ;                          6982
13111 072270  104457                        TRAP    57
13112 072272  011766                        .WORD   11766
13113 072274  011166                        .WORD   MSG3
13114 072276  000000                        .WORD   0
13115 072300  000410                        BR      26$
13116 072302  032767  000001  107750 25$:   BIT     #1,ERROUT             ;                          6985
13117 072310  001404                        BEQ     26$                   ;                          6991
13118 072312  104457                        TRAP    57
13119 072314  011767                        .WORD   11767
13120 072316  011166                        .WORD   MSG3
13121 072320  000000                        .WORD   0
13122 072322  016646  000030         26$:   MOV     30(SP),-(SP)          ; LUN,*                    6994
13123 072326  016746  142010               MOV     BOARD,-(SP)
13124 072332  004767  145546               JSR     PC,UP.SOFT.COUNT
13125 072336  022626               27$:    CMP     (SP)+,(SP)+           ;                          6958
13126 072340  010200               28$:    MOV     R2,R0                 ; WRDCNT,*                 7000
13127 072342  006300                        ASL     R0
13128 072344  066700  140320               ADD     WPTR,R0
13129 072350  010067  140314               MOV     R0,WPTR
13130 072354  010216                        MOV     R2,(SP)               ; WRDCNT,*                 7001
13131 072356  012746  000400               MOV     #400,-(SP)
13132 072362  004767  112602               JSR     PC,BL$DIV
13133 072366  060300                        ADD     R3,R0                 ; SECTOR,*
13134 072370  010003                        MOV     R0,R3                 ; *,SECTOR
13135 072372  062706  000024               ADD     #24,SP                ;                          6830
13136 072376  000167  176332               JMP     5$                    ;                          6829
13137 072402  032767  000001  107652 29$:   BIT     #1,EFN$21             ;                          7012
13138 072410  001002                        BNE     30$
13139 072412  000167  000440               JMP     40$
13140 072416  016600  000006         30$:   MOV     6(SP),R0              ; LUN,*                    7019
13141 072422  006300                        ASL     R0
13142 072424  016001  034500               MOV     TOP.SECT(R0),R1
13143 072430  016005  034460               MOV     LOW.SECT(R0),R5       ; *,SECTOR
13144 072434  000167  000406               JMP     39$
13145                              ;MLX4
13146                              ;                                                27-Mar-1982 19:24:42  TOPS
13147                                      TESTING RANDOM DATA                      27-Mar-1982 19:23:44  PA:<
13148 072440  016646  000006         31$:   MOV     6(SP),-(SP)           ; LUN,*                    7021
13149 072444  012746  000400               MOV     #400,-(SP)
13150 072450  012746  022670               MOV     #RBUFF,-(SP)
13151 072454  010546                        MOV     R5,-(SP)              ; SECTOR,*
13152 072456  004767  153130               JSR     PC,READ
13153 072462  062706  000010               ADD     #10,SP
13154 072466  020027  000005               CMP     R0,#5
13155 072472  001164                        BNE     38$
13156 072474  004767  144632               JSR     PC,ISOLATE
13157 072500  032767  000001  107552       BIT     #1,ERROUT             ;                          7024
13158 072506  001423                        BEQ     32$                   ;                          7026
13159 072510  017700  141674               MOV     @ML.REG+42,R0
```

SEQ 0291

```
13160 072514  006200                    ASR     R0
13161 072516  006200                    ASR     R0
13162 072520  006200                    ASR     R0
13163 072522  006200                    ASR     R0
13164 072524  006200                    ASR     R0
13165 072526  006200                    ASR     R0
13166 072530  042700  177700            BIC     #177700,R0
13167 072534  010046                    MOV     R0,-(SP)
13168 072536  012746  006640            MOV     #FMT10B,-(SP)
13169 072542  012746  000002            MOV     #2,-(SP)
13170 072546  010600                    MOV     SP,R0                   ; SP,*
13171 072550  104414                    TRAP    14
13172 072552  062706  000006            ADD     #6,SP
13173 072556  017766  141630  000016 32$:MOV     @ML.REG+44,16(SP)       ; *,OLDSEC                      7035
13174 072564  017700  141620            MOV     @ML.REG+42,R0           ;                               7036
13175 072570  006200                    ASR     R0
13176 072572  006200                    ASR     R0
13177 072574  006200                    ASR     R0
13178 072576  006200                    ASR     R0
13179 072600  006200                    ASR     R0
13180 072602  006200                    ASR     R0
13181 072604  042700  177700            BIC     #177700,R0
13182 072610  010066  000012            MOV     R0,12(SP)               ; *,OLDCHN
13183 072614  012746  000001            MOV     #1,-(SP)                ;
13184 072620  016646  000024            MOV     24(SP),-(SP)            ;                               7044
13185 072624  016646  000012            MOV     12(SP),-(SP)            ; COMMAND,*
13186 072630  012746  000400            MOV     #400,-(SP)              ; LUN,*
13187 072634  016646  000030            MOV     30(SP),-(SP)
13188 072640  016646  000030            MOV     30(SP),-(SP)            ; PTR,*
13189 072644  004767  153354            JSR     PC,RETRY                ; OLDSEC,*
13190 072650  062706  000014            ADD     #14,SP
13191 072654  020027  000005            CMP     R0,#5
13192 072660  001052                    BNE     35$
13193 072662  027766  141524  000016    CMP     @ML.REG+44,16(SP)       ; *,OLDSEC                      7053
13194 072670  001035                    BNE     34$
13195 072672  016646  000012            MOV     12(SP),-(SP)            ; OLDCHN,*
13196 072676  017700  141506            MOV     @ML.REG+42,R0
13197 072702  006200                    ASR     R0
13198 072704  006200                    ASR     R0
13199 072706  006200                    ASR     R0
13200                           ;MLX4
13201                           ;                       TESTING RANDOM DATA     27-Mar-1982 19:24:42    TOPS
13202                                                                           27-Mar-1982 19:23:44    PA:<
13203 072710  006200                    ASR     R0
13204 072712  006200                    ASR     R0
13205 072714  006200                    ASR     R0
13206 072716  042700  177700            BIC     #177700,R0
13207 072722  020026                    CMP     R0,(SP)+
13208 072724  001017                    BNE     34$
13209 072726  032767  000001  107324    BIT     #1,ERROUT               ;                               7057
13210 072734  001404                    BEQ     33$
13211 072736  104456                    TRAP    56                      ;                               7062
13212 072740  011770                    .WORD   11770
13213 072742  011202                    .WORD   MSG4
13214 072744  000000                    .WORD   0
13215 072746  016646  000006      33$:  MOV     6(SP),-(SP)             ; LUN,*                         7065
13216 072752  016746  141364            MOV     BOARD,-(SP)
```

```
                                                                                                                                SEQ 0292
 13217 072756  004767  144440                      JSR     PC,UP.HARD.COUNT
 13218 072762  000427                              BR      37$
 13219 072764  032767  000001  107266  34$:        BIT     #1,ERROUT               ;                                       7053
 13220 072772  001415                              BEQ     36$                     ;                                       7070
 13221 072774  104457                              TRAP    57                      ;
 13222 072776  011771                              .WORD   11771                   ;                                       7075
 13223 073000  011166                              .WORD   MSG3
 13224 073002  000000                              .WORD   0
 13225 073004  000410                              BR      36$
 13226 073006  032767  000001  107244  35$:        BIT     #1,ERROUT               ;                                       7078
 13227 073014  001404                              BEQ     36$                     ;                                       7084
 13228 073016  104457                              TRAP    57                      ;
 13229 073020  011772                              .WORD   11772                   ;                                       7089
 13230 073022  011166                              .WORD   MSG3
 13231 073024  000000                              .WORD   0
 13232 073026  016646  000006        36$:          MOV     6(SP),-(SP)             ; LUN,*                                 7092
 13233 073032  016746  141304                      MOV     BOARD,-(SP)
 13234 073036  004767  145042                      JSR     PC,UP.SOFT.COUNT
 13235 073042  022626              37$:          CMP     (SP)+,(SP)+             ;
 13236 073044  005205              38$:          INC     R5                      ;                                       7023
 13237 073046  020501              39$:          CMP     R5,R1                   ; SECTOR                                7019
 13238 073050  101002                            BHI     40$                     ; SECTOR,*
 13239 073052  000167  177362                    JMP     31$
 13240 073056  005266  000006        40$:          INC     6(SP)
 13241 073062  026666  000006  000010  41$:      CMP     6(SP),10(SP)            ; LUN                                   6816
 13242 073070  002002                            BGE     42$                     ; LUN,*
 13243 073072  000167  175524                    JMP     2$
 13244 073076  005266  000024        42$:          INC     24(SP)
 13245 073102  026666  000024  000044            CMP     24(SP),44(SP)           ; COUNT                                 6812
 13246 073110  003002                            BGT     43$                     ; COUNT,REPEAT
 13247 073112  000167  175462                    JMP     1$
 13248 073116  062706  000026        43$:          ADD     #26,SP
 13249 073122  000207                            RTS     PC                      ;                                       6746
 13250
 13251                                  ; Routine Size:  634 words
 13252                                  ; Maximum stack depth per invocation:  35 words
 13260
 13261
```

```
13263 ;MLX4
13264 ;                 TESTING RANDOM DATA & WORD COUNTS                27-Mar-1982 19:24:42    TOPS-20 Bliss-16 V2(212)
13265 ;                                                                  27-Mar-1982 19:23:44    PA:<NEALE>MLX4.BLI.5 (45)
13266 ;          7106  %sbttl 'TESTING RANDOM DATA & WORD COUNTS'
13267 ;          7107  routine RAND2 (REPEAT) : novalue =
13268 ;          7108      begin                                          !* 1 * START OF ROUTINE
13269 ;          7109
13270 ;          7110  !++
13271 ;          7111  !   ROUTINE:      RAND2(REPEAT)
13272 ;          7112  !
13273 ;          7113  !   PURPOSE:      TO TEST ALL UNITS IN A SEQUENTIAL FASHION, BUT
13274 ;          7114  !                 USING RANDOM WORD COUNTS
13275 ;          7115  !
13276 ;          7116  !   ARGUMENT:     REPEAT = NUMBER OF TIMES TO EXECUTE THIS ROUTINE
13277 ;          7117  !                 BEFORE RETURNING TO THE CALLER (OPT5).
13278 ;          7118  !
13279 ;          7119  !      NOTE:      SINCE ONLY THE WORD COUNTS AND THE DATA ARE RANDOM,
13280 ;          7120  !                 THE OTHER TRANSFER VALUES MUST BE SET UP BY THIS
13281 ;          7121  !                 ROUTINE.  THESE VALUES INCLUDE:
13282 ;          7122  !                    THE LUN, SECTOR AND BUFFER POINTERS.
13283 ;          7123  !
13284 ;          7124  !   THE CODE FOR 'RAND2' IN BRIEF:
13285 ;          7125  !
13286 ;          7126  !   BEGIN 1 (START OF ROUTINE)
13287 ;          7127  !   SAY ROUTINE IS RUNNING
13288 ;          7128  !   INCR COUNT FROM 1 TO REPEAT
13289 ;          7129  !   : BEGIN 2 (START OF REPEAT LOOP FOR THE ROUTINE)
13290 ;          7130  !   : INCR LUN FROM 0 TO LAST
13291 ;          7131  !   : : BEGIN 3 (START OF LOGICAL UNIT SELECTION LOOP)
13292 ;          7132  !   : : GENERATE THE RANDOM PATTERN
13293 ;          7133  !   : : TESTLOOP:
13294 ;          7134  !   : : : BEGIN 4 (START OF LOOP THAT COMPLETELY TESTS 1 UNIT)
13295 ;          7135  !   : : : IF UNIT IS ACTIVE
13296 ;          7136  !   : : : THEN
13297 ;          7137  !   : : : : BEGIN 5 (START OF TEST FOR AN ACTIVE UNIT)
13298 ;          7138  !   : : : : SECTOR = LOWEST
13299 ;          7139  !   : : : : INITIALIZE THE WRITE AND READ BUFFER POINTERS
13300 ;          7140  !   : : : : WHILE SECTOR LEQ HIGHEST DO
13301 ;          7141  !   : : : : : BEGIN 6 (START OF A PASS THROUGH ALL SECTORS)
13302 ;          7142  !   : : : : : CHOOSE A RANDOM WORD COUNT
13303 ;          7143  !   : : : : : SET UP BUFFER POINTERS BEFORE TRANSFER
13304 ;          7144  !   : : : : : CALCULATE NEXT STARTING SECTOR (BASED ON WORD COUNT)
13305 ;          7145  !   : : : : : IF NEXT STARTING SECTOR GTR HIGHEST
13306 ;          7146  !   : : : : : : THEN ADJUST THE WORD COUNT & NEXT SECTOR SO THEY FIT
13307 ;          7147  !   : : : : : : :    WITHIN THE TESTABLE SECTOR LIMITS
13308 ;          7148  !   : : : : : WRITE
13309 ;          7149  !   : : : : : LOOK FOR ERRORS (IF DROP UNIT, LEAVE TESTLOOP)
13310 ;          7150  !   : : : : : CHOOSE WHETHER TO WRITE CHECK OR READ
13311 ;          7151  !   : : : : : DO THE WRITE CHECK OR READ
13312 ;          7152  !   : : : : : LOOK FOR ERRORS (IF DROP UNIT, LEAVE TESTLOOP)
13313 ;          7153  !   : : : : : SECTOR = THE CALCULATED NEXT STARTING SECTOR
13314 ;          7154  !   : : : : : CHANGE BUFFER POINTERS AFTER TRANSFER
13315 ;          7155  !   : : : : : END 6 (END OF A PASS THROUGH ALL SECTORS)
13316 ;          7156  !   : : : : END 5 (END OF TEST FOR AN ACTIVE UNIT)
13317 ;          7157  !   : : : END 4 (END TESTLOOP)
```

```
13319 ;MLX4
13320 ;                 TESTING RANDOM DATA & WORD COUNTS
13321 ;
13322 ;          7158  !  : : END 3 (END OF LOGICAL UNIT SELECTION LOOP)
13323 ;          7159  !  : END 2 (END OF REPEAT LOOP FOR THIS ROUTINE)
13324 ;          7160  !  RETURN
13325 ;          7161  !  END 1 (END OF ROUTINE)
13326 ;          7162  !--
13327 ;          7163
13328 ;          7164      local
13329 ;          7165          WRDCNT,
13330 ;          7166          SECTOR,
13331 ;          7167          NEXT_SECT,
13332 ;          7168          VALUE,
13333 ;          7169          OLDSEC,
13334 ;          7170          OLDCHN,
13335 ;          7171          PTR,
13336 ;          7172          COMMAND,
13337 ;          7173          DBL_VALUE;
13338 ;          7174
13339 ;          7175      label
13340 ;          7176          LOOP;
13341 ;          7177
13342 ;          7178      PRINTB (SAY1, RTN5B);
13343 ;          7179      !'RAND2'
13344 ;          7180
13345 ;          7181      incr COUNT from 1 to .REPEAT do
13346 ;          7182          begin                                   !* 2 *  START OF REPEAT LOOP FOR THIS ROUTINE
13347 ;          7183
13348 ;          7184      incr LUN from 0 to (.L$UNIT - 1) do
13349 ;          7185          begin
13350 ;          7186          GEN5 ();                                !* 3 *  START OF LOGICAL UNIT SELECTION LOOP
13351 ;          7187  LOOP :                                          !RANDOM DATA PATTERN
13352 ;          7188          begin                                   !* 4 *  START OF THE LOOP FOR COMPLETELY TESTING 1 UNIT
13353 ;          7189
13354 ;          7190          if .DRIVE_STATUS [.LUN] eql ACTIVE
13355 ;          7191          then
13356 ;          7192              begin                               !* 5 *  START OF TEST FOR AN ACTIVE UNIT
13357 ;          7193              L$LUN = .LUN;
13358 ;          7194              SECTOR = LOWEST;
13359 ;          7195              WPTR = WBUFF;
13360 ;          7196              RPTR = RBUFF;
13361 ;          7197
13362 ;          7198              while .SECTOR leqa HIGHEST do
13363 ;          7199                  begin                           !* 6 *  START OF A PASS THROUGH ALL SECTORS
13364 ;          7200                  WRDCNT = RNDWC ();              !EXPECT VALUE BETWEEN 1 AND BUFSIZ
13365 ;          7201                  SET_PTRS (.WRDCNT);
13366 ;          7202                  NEXT_SECT = .SECTOR + (.WRDCNT/256);
13367 ;          7203
13368 ;          7204                  if .SECTOR eql .NEXT_SECT then NEXT_SECT = .NEXT_SECT + 1;
13369 ;          7205
13370 ;          7206                  if .NEXT_SECT gtra HIGHEST
13371 ;          7207                  then
13372 ;          7208                      begin
13373 ;          7209                      WRDCNT = 256*(HIGHEST - .SECTOR + 1);
```

```
13375 ;ML X4
13376 ;                 TESTING RANDOM DATA & WORD COUNTS
13377
13378 ;          7210                           NEXT_SECT = HIGHEST + 1;
13379 ;          7211                           end;
13380 ;          7212
13381 ;          7213                       VALUE = write (.LUN, .WRDCNT, .WPTR, .SECTOR);
13382 ;          7214
13383 ;          7215                       !+
13384 ;          7216                       !  SEE HOW SUCCESSFUL THE WRITE WAS:
13385 ;          7217                       !-
13386 ;          7218
13387 ;          7219                       selectone .VALUE of              !SEE 'SYSFRR' FOR DEFINITION
13388 ;          7220                           set                          !OF ERROR # CONTAINED IN 'VALUE'
13389 ;          7221
13390 ;          7222                           [1] :
13391 ;          7223                               begin                    !* 6A * RETRY ALLOWED
13392 ;          7224
13393 ;          7225                               if RETRY (SIX, write, .LUN, .WRDCNT, .WPTR, .SECTOR) neq 0
13394 ;          7226                               then                     !THE RETRY FAILED -- SYSTEM FATAL ERROR
13395 ;          7227                                   begin
13396 ;          7228                                   WHY_DROPT [.LUN] = CODE_4;
13397 ;          7229                                   ERRDF (5201, MSG1, 0);  !**** OPTION 5, RAND2 ERROR 01 ****
13398 ;          7230                                   DODU (.LUN);
13399 ;          7231                                   leave LOOP;       !JUMP JUST BEYOND END OF BLOCK * 4 *
13400 ;          7232                                   end;
13401 ;          7233
13402 ;          7234                               end;                     !* 6A *
13403 ;          7235
13404 ;          7236                           [2] :
13405 ;          7237                               begin                    !* 6B * FATAL CONTROLLER ERROR -- NO RETRY ALLOWED
13406 ;          7238                               WHY_DROPT [.LUN] = CODE_5;
13407 ;          7239                               ERRDF (5202, MSG1, 0);       !**** OPTION 5, RAND2 ERROR 02 ****
13408 ;          7240                               DODU (.LUN);
13409 ;          7241                               leave LOOP;              !JUMP JUST BEYOND END OF BLOCK * 4 *
13410 ;          7242                               end;                     !* 6B *
13411 ;          7243
13412 ;          7244                           [3] :
13413 ;          7245                               begin                    !* 6C * FATAL DRIVE ERROR -- NO RETRY ALLOWED
13414 ;          7246                               ERRDF (5203, MSG1, 0);       !**** OPTION 5, RAND2 ERROR 03 ****
13415 ;          7247                               WHY_DROPT [.LUN] = CODE_6;
13416 ;          7248                               DODU (.LUN);
13417 ;          7249                               leave LOOP;              !JUMP JUST BEYOND END OF BLOCK * 4 *
13418 ;          7250                               end;                     !* 6C *
13419 ;          7251                           tes;
13420 ;          7252
13421 ;          7253                       COMMAND = CHOOSE ();
13422 ;          7254
13423 ;          7255                       if .COMMAND eql read
13424 ;          7256                       then
13425 ;          7257                           begin
13426 ;          7258                           PTR = .RPTR;
13427 ;          7259                           VALUE = read (.LUN, .WRDCNT, .RPTR, .SECTOR);
13428 ;          7260                           end
13429 ;          7261                       else
```

```
;MLX4
;
;
;          7262                    begin
;          7263                    PTR = .WPTR;
;          7264                    VALUE = CHECK (.LUN, .WRDCNT, .WPTR, .SECTOR);
;          7265                    end;
;          7266
;          7267                !+
;          7268                !  SEE HOW SUCCESSFUL THE OPERATION WAS:
;          7269                !-
;          7270
;          7271                selectone .VALUE of         !SEE 'SYSERR' FOR DEFINITION
;          7272                    set                     !OF ERROR # CONTAINED IN 'VALUE'
;          7273
;          7274                    [0] :
;          7275
;          7276                        if .COMMAND eql read
;          7277                        then
;          7278                            begin
;          7279
;          7280                            if (DBL_VALUE = DOUBLE_CHECK (.WPTR, .RPTR, .WRDCNT)) neq 0
;          7281                            then
;          7282                                begin
;          7283                                SAYWHO (.LUN);
;          7284                                PRINTB (SAY1, MSG5);        !'ECC LOGIC FAILED TO DETECT DATA ERROR'
;          7285                                PRINTB (FMT12A, ..DBL_VALUE, .DBL_VALUE);
;          7286                                                            !'GOOD DATA: XXXXXX AT LOCATION YYYYYY'
;          7287                                DBL_VALUE = .DBL_VALUE + BUFSIZ*2;
;          7288                                PRINTB (FMT12B, ..DBL_VALUE, .DBL_VALUE);
;          7289                                                            !'BAD DATA:  PPPPPP AT LOCATION QQQQQQ'
;          7290                                WHY_DROPT [.LUN] = CODE_8;
;          7291                                ERRDF (5204, MSG1, 0);      !**** OPTION 5, RAND2 ERROR 04 ****
;          7292                                DODU (.LUN);
;          7293                                leave LOOP; !JUMP JUST BEYOND END OF BLOCK * 4 *
;          7294                                end;
;          7295
;          7296                            end;
;          7297
;          7298                    [1] :
;          7299                        begin                   !* 6D * RETRY ALLOWED
;          7300
;          7301                        if RETRY (SIX, .COMMAND, .LUN, .WRDCNT,  PTR, .SECTOR) neq 0
;          7302                        then                    !THE RETRY FAILED -- SYSTEM FATAL ERROR
;          7303                            begin
;          7304                            WHY_DROPT [.LUN] = CODE_4;
;          7305                            ERRDF (5205, MSG1, 0);  !**** OPTION 5, RAND2 ERROR 05 ****
;          7306                            DODU (.LUN);
;          7307                            leave LOOP;     !JUMP JUST BEYOND END OF BLOCK * 4 *
;          7308                            end;
;          7309
;          7310                        end;                    !* 6D *
;          7311
;          7312                    [2] :
;          7313                        begin                   !* 6E * FATAL CONTROLLER ERROR -- NO RETRY ALLOWED
```

```
;        7314                    WHY_DROPT [.LUN] = CODE_5;
;        7315                    ERRDF (5206, MSG1, 0);      !**** OPTION 5, RAND2 ERROR 06 ****
;        7316                    DODU (.LUN);
;        7317                    leave LOOP;                 !JUMP JUST BEYOND END OF BLOCK * 4 *
;        7318                    end;                        !* 6E *
;        7319
;        7320                [3] :
;        7321                    begin                       !* 6F * FATAL DRIVE ERROR -- NO RETRY ALLOWED
;        7322                    ERRDF (5207, MSG1, 0);      !**** OPTION 5, RAND2 ERROR 07 ****
;        7323                    WHY_DROPT [.LUN] = CODE_6;
;        7324                    DODU (.LUN);
;        7325                    leave LOOP;                 !JUMP JUST BEYOND END OF BLOCK * 4 *
;        7326                    end;                        !* 6F *
;        7327
;        7328                [4] :
;        7329                    begin                       !* 6G * UNRECOVERABLE DATA ERROR
;        7330                    ISOLATE ();
;        7331                    ERRDF (5208, MSG2, 0);      !**** OPTION 5, RAND2 ERROR 08 ****
;        7332                    WHY_DROPT [.LUN] = CODE_7;
;        7333                    DODU (.LUN);
;        7334                    leave LOOP;                 !JUMP JUST BEYOND END OF BLOCK * 4 *
;        7335                    end;                        !* 6G *
;        7336
;        7337                [5] :
;        7338                    begin                       !* 6H * RECOVERABLE DATA ERROR
;        7339                    ISOLATE ();
;        7340
;        7341                    if .ERROUT then PRINTB (FMT10B, .CHAN);
;        7342
;        7343                    !'    BIT QQ'
;        7344                    OLDSEC = .MLEL;
;        7345                    OLDCHN = .CHAN;
;        7346
;        7347                    if RETRY (ONE, .COMMAND, .LUN, .WRDCNT, .PTR, .SECTOR) eql 5
;        7348                    then
;        7349
;        7350                        if ((.MLEL eql .OLDSEC) and (.CHAN eql .OLDCHN))
;        7351                        then
;        7352                            begin
;        7353
;        7354                            if .ERROUT then ERRHRD (5209, MSG4, 0);
;        7355
;        7356                                        !**** OPTION 5, RAND2 ERROR 09 ****
;        7357                            UP_HARD_COUNT (.LUN, .BOARD);
;        7358                            end
;        7359                        else
;        7360                            begin
;        7361
;        7362                            if .ERROUT then ERRSOFT (5210, MSG3, 0);
;        7363
;        7364                                        !**** OPTION 5, RAND2 ERROR 10 ****
;        7365                            UP_SOFT_COUNT (.LUN, .BOARD);
```

```
13543 ;MLX4
13544 ;
13545
13546 ;          7366                                    end
13547 ;          7367
13548 ;          7368                                else
13549 ;          7369                                    begin
13550 ;          7370
13551 ;          7371                                    if .ERROUT then ERRSOFT (5211, MSG3, 0);
13552 ;          7372
13553 ;          7373                                                    !**** OPTION 5, RAND2 ERROR 11 ****
13554 ;          7374                                    UP_SOFT_COUNT (.LUN, .BOARD);
13555 ;          7375                                    end;
13556 ;          7376
13557 ;          7377                                end;                !* 6H *
13558 ;          7378                            tes;
13559 ;          7379
13560 ;          7380                        SECTOR = .NEXT_SECT;
13561 ;          7381                        WPTR = .WPTR + 2;
13562 ;          7382                        end;                        !* 6 *  END OF A PASS THROUGH ALL SECTORS
13563 ;          7383
13564 ;          7384                    end;                            !* 5 *  END OF TEST FOR AN ACTIVE UNIT
13565 ;          7385
13566 ;          7386                !+
13567 ;          7387                ! Test to see if this uut's address space is
13568 ;          7388                ! to be read for soft errors.  This test is
13569 ;          7389                ! is intended for DMT purposes.
13570 ;          7390                !-
13571 ;          7391
13572 ;          7392                if .EFN$21                          !Is the background pattern to be read
13573 ;          7393                then
13574 ;          7394                    begin
13575 ;          7395
13576 ;          7396 !
13577 ;          7397 ! version czmlbb changed incr to incru
13578 ;          7398 !
13579 ;          7399                    incru SECTOR from LOWEST to HIGHEST do
13580 ;          7400
13581 ;          7401                        if read (.LUN, 256, RBUFF, .SECTOR) eql 5
13582 ;          7402                        then
13583 ;          7403                            begin
13584 ;          7404                            ISOLATE ();             !Find the failing bank and board no.
13585 ;          7405
13586 ;          7406                            if .ERROUT then PRINTB (FMT10B, .CHAN);
13587 ;          7407
13588 ;          7408                            ! Print where the error is
13589 ;          7409                            !
13590 ;          7410                            ! Save the contents of the ML error location
13591 ;          7411                            ! register so we can compare it to the new
13592 ;          7412                            ! contents of this register after the retry.
13593 ;          7413                            ! This is done to classify the error.
13594 ;          7414                            !
13595 ;          7415                            OLDSEC = .MLEL;
13596 ;          7416                            OLDCHN = .CHAN;
13597 ;          7417                            !
```

```
;        7418                    ! Do a classify retry call.  If the same error
;        7419                    ! occures then classify it as a hard error.  If
;        7420                    ! a different error occured or the error went away
;        7421                    ! then classify it as a soft error.
;        7422                    !
;        7423
;        7424                    if RETRY (ONE, .COMMAND, .LUN, 256, .PTR, .OLDSEC) eql 5
;        7425                    then
;        7426                    !
;        7427                    ! The same error occured so see if it is at the same
;        7428                    ! sector and channel number, if so then classify
;        7429                    ! it as a hard error else classiy it as a soft
;        7430                    ! error.
;        7431                    !
;        7432
;        7433                        if ((.MLEL eql .OLDSEC) and (.CHAN eql .OLDCHN))
;        7434                        then
;        7435                            begin           !Same error occured 'hard'
;        7436
;        7437                            if .ERROUT      !Print error if enabled
;        7438                            then
;        7439                                begin
;      P 7440                                ERRHRD (5212,       !Error number
;      P 7441                                    MSG4,   !Error message
;        7442                                    0);     !Additional message routine
;        7443                                end;
;        7444
;        7445                            UP_HARD_COUNT (.LUN, .BOARD);
;        7446                            end
;        7447                        else
;        7448                            begin           !Not the same error 'soft'
;        7449
;        7450                            if .ERROUT      !Print error if enabled
;        7451                            then
;        7452                                begin
;      P 7453                                ERRSOFT (5213,      !Error number
;      P 7454                                    MSG3,   !Error message
;        7455                                    0);     !Additional message routine
;        7456                                end;
;        7457
;        7458                            UP_SOFT_COUNT (.LUN, .BOARD);
;        7459                            end
;        7460
;        7461                    else                    !Not the same error 'soft'
;        7462                        begin
;        7463
;        7464                        if .ERROUT          !Print error if enabled
;        7465                        then
;        7466                            begin
;      P 7467                            ERRSOFT (5214,  !Error number
;      P 7468                                MSG3,       !Error message
;        7469                                0);         !Additional message routine
```

SEQ 0300


```
13655 ;MLX4
13656 ;                 TESTING RANDOM DATA & WORD COUNTS
13657
13658 ;          7470                                   end;
13659 ;          7471
13660 ;          7472                               UP_SOFT_COUNT (.LUN, .BOARD);
13661 ;          7473                               end;
13662 ;          7474
13663 ;          7475                           end;
13664 ;          7476
13665 ;          7477                       end;
13666 ;          7478
13667 ;          7479                   end;                        !* 4 *  END OF LOOP THAT COMPLETELY TESTS 1 UNIT
13668 ;          7480                   end;                        !* 3 * END OF LOGICAL UNIT SELECTION LOOP
13669 ;          7481
13670 ;          7482               end;                            !* 2 *  END OF REPEAT LOOP FOR THIS ROUTINE
13671 ;          7483
13672 ;          7484           return;
13673 ;          7485           end;                                !* 1 *  END OF ROUTINE
13677
13678                                          .SBTTL  RAND2 TESTING RANDOM DATA & WORD COUNTS
13682 073124  004167  112204          RAND2:  JSR     R1,$SAVE5                       ;                       7107
13683 073130  162706  000022                  SUB     #22,SP
13684 073134  012746  007672                  MOV     #RTN5B,-(SP)                    ;                       7178
13685 073140  012746  007072                  MOV     #SAY1,-(SP)
13686 073144  012746  000002                  MOV     #2,-(SP)
13687 073150  010600                          MOV     SP,R0                           ; SP,*
13688 073152  104414                          TRAP    14
13689 073154  005066  000022                  CLR     22(SP)                          ; COUNT
13690 073160  000167  002232                  JMP     44$                                             7181
13691 073164  016766  106622  000026  1$:     MOV     L$UNIT,26(SP)                   ;                       7184
13692 073172  005001                          CLR     R1                              ; LUN
13693 073174  000167  002204                  JMP     43$
13694 073200  004767  146414          2$:     JSR     PC,GEN5                         ;                       7186
13695 073204  010100                          MOV     R1,R0                           ; LUN,*                 7190
13696 073206  006200                          ASR     R0
13697 073210  006200                          ASR     R0
13698 073212  006200                          ASR     R0
13699 073214  062700  034442                  ADD     #DRIVE.STATUS,R0
13700 073220  010046                          MOV     R0,-(SP)
13701 073222  010146                          MOV     R1,-(SP)                        ; LUN,*
13702 073224  042716  177770                  BIC     #177770,(SP)
13703 073230  012746  000001                  MOV     #1,-(SP)
13704 073234  005046                          CLR     -(SP)
13705 073236  004767  111114                  JSR     PC,BL$GT2
13706 073242  062706  000010                  ADD     #10,SP
13707 073246  005300                          DEC     R0
13708 073250  001402                          BEQ     4$
13709 073252  000167  001460          3$:     JMP     31$
```

```
13711                                          ;MLX4
13712                                          ;                                                    27-Mar-1982 19:24:42    TOPS
13713                                          ;       TESTING RANDOM DATA & WORD COUNTS            27-Mar-1982 19:23:44    PA:<
13714 073256  010167  106612          4$:      MOV     R1,L$LUN                    ; LUN,*                                  7193
13715 073262  010100                           MOV     R1,R0                       ; LUN,*                                  7194
13716 073264  006300                           ASL     R0
13717 073266  016003  034460                   MOV     LOW.SECT(R0),R3             ; *,SECTOR
13718 073272  012767  012670  137370           MOV     #WBUFF,WPTR                 ;                                        7195
13719 073300  012767  022670  137364           MOV     #RBUFF,RPTR                 ;                                        7196
13720 073306  012766  034500  000024           MOV     #TOP.SECT,24(SP)            ;                                        7198
13721 073314  060066  000024                   ADD     R0,24(SP)
13722 073320  020376  000024          5$:      CMP     R3,@24(SP)                  ; SECTOR,*
13723 073324  101352                           BHI     3$
13724 073326  004767  174652                   JSR     PC,RNDWC                    ;                                        7200
13725 073332  010002                           MOV     R0,R2                       ; *,WRDCNT
13726 073334  010246                           MOV     R2,-(SP)                    ; WRDCNT,*                               7201
13727 073336  004767  153570                   JSR     PC,SET.PTRS
13728 073342  010216                           MOV     R2,(SP)                     ; WRDCNT,*                               7202
13729 073344  012746  000400                   MOV     #400,-(SP)
13730 073350  004767  111614                   JSR     PC,BL$DIV
13731 073354  060300                           ADD     R3,R0                       ; SECTOR,*
13732 073356  010066  000012                   MOV     R0,12(SP)                   ; *,NEXT.SECT
13733 073362  020300                           CMP     R3,R0                       ; SECTOR,NEXT.SECT                       7204
13734 073364  001002                           BNE     6$
13735 073366  005266  000012                   INC     12(SP)                      ; NEXT.SECT
13736 073372  026676  000012  000030  6$:      CMP     12(SP),@30(SP)              ; NEXT.SECT,*                            7206
13737 073400  101415                           BLOS    7$
13738 073402  017600  000030                   MOV     @30(SP),R0                  ;                                        7209
13739 073406  160300                           SUB     R3,R0                       ; SECTOR,*
13740 073410  000300                           SWAB    R0                          ;                                        7208
13741 073412  105000                           CLRB    R0
13742 073414  010002                           MOV     R0,R2
13743 073416  062702  000400                   ADD     #400,R2                     ; *,WRDCNT                               7209
13744 073422  017666  000030  000012           MOV     @30(SP),12(SP)              ; *,WRDCNT
13745 073430  005266  000012                   INC     12(SP)                      ; *,NEXT.SECT                            7210
13746 073434  010116                           MOV     R1,(SP)                     ; NEXT.SECT
13747 073436  010246                   7$:     MOV     R2,-(SP)                    ; LUN,*                                  7213
13748 073440  016746  137224                   MOV     WPTR,-(SP)                  ; WRDCNT,*
13749 073444  010346                           MOV     R3,-(SP)
13750 073446  004767  151752                   JSR     PC,WRITE                    ; SECTOR,*
13751 073452  010004                           MOV     R0,R4
13752 073454  020427  000001                   CMP     R4,#1                       ; *,VALUE
13753 073460  001031                           BNE     8$                          ; VALUE,*                                7219
13754 073462  012746  000006                   MOV     #6,-(SP)
13755 073466  012746  045424                   MOV     #WRITE,-(SP)                ;                                        7225
13756 073472  010146                           MOV     R1,-(SP)
13757 073474  010246                           MOV     R2,-(SP)                    ; LUN,*
13758 073476  016746  137166                   MOV     WPTR,-(SP)                  ; WRDCNT,*
13759 073502  010346                           MOV     R3,-(SP)
13760 073504  004767  152514                   JSR     PC,RETRY                    ; SECTOR,*
13761 073510  062706  000014                   ADD     #14,SP
13762 073514  005700                           TST     R0
13763 073516  001446                           BEQ     11$
13764 073520  112761  000004  034446          MOVB    #4,WHY.DROPT(R1)            ; *,*(LUN)                               7228
13765 073526  104455                           TRAP    55                          ;                                        7229
```

```
                                        ;MLX4
                                        ;                    TESTING RANDOM DATA & WORD COUNTS          27-Mar-1982 19:24:42    TOPS
                                                                                                        27-Mar-1982 19:23:44    PA:<
013770 073530 012121                            .WORD   12121
013771 073532 011070                            .WORD   MSG1
013772 073534 000000                            .WORD   0
013773 073536 010100                            MOV     R1,R0
013774 073540 104451                            TRAP    51                      ; LUN,*                         7230
013775 073542 000431                            BR      10$                     ;                               7231
013776 073544 020427 000002             8$:     CMP     R4,#2                   ; VALUE,*                       7219
013777 073550 001012                            BNE     9$
013778 073552 112761 000005 034446              MOVB    #5,WHY.DROPT(R1)        ; *,*(LUN)                      7238
013779 073560 104455                            TRAP    55                      ;                               7239
013780 073562 012122                            .WORD   12122
013781 073564 011070                            .WORD   MSG1
013782 073566 000000                            .WORD   0
013783 073570 010100                            MOV     R1,R0
013784 073572 104451                            TRAP    51                      ; LUN,*                         7240
013785 073574 000414                            BR      10$                     ;                               7241
013786 073576 020427 000003             9$:     CMP     R4,#3                   ; VALUE,*                       7219
013787 073602 001014                            BNE     11$
013788 073604 104455                            TRAP    55                      ;                               7246
013789 073606 012123                            .WORD   12123
013790 073610 011070                            .WORD   MSG1
013791 073612 000000                            .WORD   0
013792 073614 112761 000006 034446              MOVB    #6,WHY.DROPT(R1)        ; *,*(LUN)                      7247
013793 073622 010100                            MOV     R1,R0                   ; LUN,*                         7248
013794 073624 104451                            TRAP    51
013795 073626 062706 000012             10$:    ADD     #12,SP                  ;                               7249
013796 073632 000535                            BR      16$
013797 073634 004767 152326             11$:    JSR     PC,CHOOSE               ;                               7253
013798 073640 010066 000022                     MOV     R0,22(SP)               ; *,COMMAND
013799 073644 005005                            CLR     R5                      ;                               7255
013800 073646 020027 045612                     CMP     R0,#READ                ; COMMAND,*
013801 073652 001014                            BNE     12$
013802 073654 005205                            INC     R5
013803 073656 016766 137010 000024              MOV     RPTR,24(SP)             ; *,PTR                         7258
013804 073664 010146                            MOV     R1,-(SP)                ; LUN,*                         7259
013805 073666 010246                            MOV     R2,-(SP)                ; WRDCNT,*
013806 073670 016746 136776                     MOV     RPTR,-(SP)
013807 073674 010346                            MOV     R3,-(SP)                ; SECTOR,*
013808 073676 004767 151710                     JSR     PC,READ
013809 073702 000412                            BR      13$
013810 073704 016766 136760 000024      12$:    MOV     WPTR,24(SP)             ; *,PTR                         7263
013811 073712 010146                            MOV     R1,-(SP)                ; LUN,*                         7264
013812 073714 010246                            MOV     R2,-(SP)                ; WRDCNT,*
013813 073716 016746 136746                     MOV     WPTR,-(SP)
013814 073722 010346                            MOV     R3,-(SP)                ; SECTOR,*
013815 073724 004767 152050                     JSR     PC,CHECK
013816 073730 010004                    13$:    MOV     R0,R4                   ; *,VALUE
013817 073732 001073                            BNE     17$                     ;                               7271
013818 073734 006005                            ROR     R5                      ;                               7276
013819 073736 103402                            BLO     15$
013820 073740 000167 000750             14$:    JMP     30$
013821 073744 016746 136720             15$:    MOV     WPTR,-(SP)              ;                               7280
```

```
13823                                   ;MLX4
13824                                   ;
13825                                   ;       TESTING RANDOM DATA & WORD COUNTS                27-Mar-1982 19:24:42    TOPS
13826                                   ;                                                        27-Mar-1982 19:23:44    PA:<
13827 073750  016746  136716                    MOV     RPTR,-(SP)
13828 073754  010246                            MOV     R2,-(SP)
13829 073756  004767  150344                    JSR     PC,DOUBLE.CHECK         ; WRDCNT,*
13830 073762  062706  000006                    ADD     #6,SP
13831 073766  010066  000040                    MOV     R0,40(SP)
13832 073772  001762                            BEQ     14$                     ; *,DBL.VALUE
13833 073774  010146                            MOV     R1,-(SP)
13834 074002  004767  141522                    JSR     PC,SAYWHO               ; LUN,*                                  7283
13835 074006  012716  011216                    MOV     #MSG5,(SP)              ;                                        7284
13836 074012  012746  007072                    MOV     #SAY1,-(SP)
13837 074016  012746  000002                    MOV     #2,-(SP)
13838 074020  010600                            MOV     SP,R0                   ; SP,*
13839 074022  104414                            TRAP    14
13840 074026  016616  000046                    MOV     46(SP),(SP)             ; DBL.VALUE,*                            7285
13841 074032  017646  000046                    MOV     @46(SP),-(SP)           ; DBL.VALUE,*
13842 074036  012746  006710                    MOV     #FMT12A,-(SP)
13843 074042  012746  000003                    MOV     #3,-(SP)
13844 074044  010600                            MOV     SP,R0                   ; SP,*
13845 074046  104414                            TRAP    14
13846 074054  062766  010000  000054            ADD     #10000,54(SP)           ; *,DBL.VALUE                            7287
13847 074060  016616  000054                    MOV     54(SP),(SP)             ; DBL.VALUE,*                            7288
13848 074064  017646  000054                    MOV     @54(SP),-(SP)           ; DBL.VALUE,*
13849 074070  012746  006756                    MOV     #FMT12B,-(SP)
13850 074074  012746  000003                    MOV     #3,-(SP)
13851 074076  010600                            MOV     SP,R0                   ; SP,*
13852 074100  104414                            TRAP    14
13853 074106  112761  000010  034446            MOVB    #10,WHY.DROPT(R1)       ; *,*(LUN)                               7290
13854 074110  104455                            TRAP    55                      ;                                        7291
13855 074112  012124                            .WORD   12124
13856 074114  011070                            .WORD   MSG1
13857 074116  000000                            .WORD   0
13858 074120  010100                            MOV     R1,R0                   ; LUN,*                                  7292
13859 074122  104451                            TRAP    51
13860 074126  062706  000044                    ADD     #44,SP                  ;                                        7293
13861 074130  000506                    16$:    BR      22$
13862 074134  020427  000001            17$:    CMP     R4,#1                   ; VALUE,*                                7271
13863 074136  001031                            BNE     18$
13864 074142  012746  000006                    MOV     #6,-(SP)                ;                                        7301
13865 074146  016646  000034                    MOV     34(SP),-(SP)            ; COMMAND,*
13866 074150  010146                            MOV     R1,-(SP)                ; LUN,*
13867 074152  010246                            MOV     R2,-(SP)                ; WRDCNT,*
13868 074156  016646  000044                    MOV     44(SP),-(SP)            ; PTR,*
13869 074160  010346                            MOV     R3,-(SP)                ; SECTOR,*
13870 074164  004767  152040                    JSR     PC,RETRY
13871 074170  062706  000014                    ADD     #14,SP
13872 074172  005700                            TST     R0
13873 074174  001662                            BEQ     14$
13874 074202  112761  000004  034446            MOVB    #4,WHY.DROPT(R1)        ; *,*(LUN)                               7304
13875 074204  104455                            TRAP    55                      ;                                        7305
13876 074206  012125                            .WORD   12125
13877 074210  011070                            .WORD   MSG1
13878 074212  000000                            .WORD   0
```

SEQ 0304

```
13879
13880                                   ;MLX4
13881                                   ;                TESTING RANDOM DATA & WORD COUNTS                27-Mar-1982 19:24:42   TOPS
                                                                                                         27-Mar-1982 19:23:44   PA:<
13882 074212  010100                             MOV     R1,R0                           ; LUN,*                                 7306
13883 074214  104451                             TRAP    51
13884 074216  000450                             BR      21$                             ;
13885 074220  020427  000002              18$:    CMP     R4,#2                           ; VALUE,*                               7307
13886 074224  001012                             BNE     19$                                                                     7271
13887 074226  112761  000005  034446             MOVB    #5,WHY.DROPT(R1)                ; *,*(LUN)                              7314
13888 074234  104455                             TRAP    55                              ;                                       7315
13889 074236  012126                             .WORD   12126
13890 074240  011070                             .WORD   MSG1
13891 074242  000000                             .WORD   0
13892 074244  010100                             MOV     R1,R0                           ; LUN,*                                 7316
13893 074246  104451                             TRAP    51
13894 074250  000433                             BR      21$                             ;
13895 074252  020427  000003              19$:    CMP     R4,#3                           ; VALUE,*                               7317
13896 074256  001012                             BNE     20$                                                                     7271
13897 074260  104455                             TRAP    55                              ;                                       7322
13898 074262  012127                             .WORD   12127
13899 074264  011070                             .WORD   MSG1
13900 074266  000000                             .WORD   0
13901 074270  112761  000006  034446             MOVB    #6,WHY.DROPT(R1)                ; *,*(LUN)                              7323
13902 074276  010100                             MOV     R1,R0                           ; LUN,*                                 7324
13903 074300  104451                             TRAP    51
13904 074302  000416                             BR      21$                             ;
13905 074304  020427  000004              20$:    CMP     R4,#4                           ; VALUE,*                               7325
13906 074310  001017                             BNE     23$                                                                     7271
13907 074312  004767  143014                       JSR     PC,ISOLATE
13908 074316  104455                             TRAP    55                              ;                                       7330
13909 074320  012130                             .WORD   12130                           ;                                       7331
13910 074322  011120                             .WORD   MSG2
13911 074324  000000                             .WORD   0
13912 074326  112761  000007  034446             MOVB    #7,WHY.DROPT(R1)                ; *,*(LUN)                              7332
13913 074334  010100                             MOV     R1,R0                           ; LUN,*                                 7333
13914 074336  104451                             TRAP    51
13915 074340  062706  000022              21$:    ADD     #22,SP                          ;                                       7334
13916 074344  000167  001032              22$:    JMP     42$
13917 074350  020427  000005              23$:    CMP     R4,#5                           ; VALUE,*                               7271
13918 074354  001157                             BNE     30$
13919 074356  004767  142750                       JSR     PC,ISOLATE
13920 074362  032767  000001  105670             BIT     #1,ERROUT                       ;                                       7339
13921 074370  001423                             BEQ     24$                             ;                                       7341
13922 074372  017705  140012                       MOV     @ML.REG+42,R5
13923 074376  006205                             ASR     R5
13924 074400  006205                             ASR     R5
13925 074402  006205                             ASR     R5
13926 074404  006205                             ASR     R5
13927 074406  006205                             ASR     R5
13928 074410  006205                             ASR     R5
13929 074412  042705  177700                       BIC     #177700,R5
13930 074416  010546                             MOV     R5,-(SP)
13931 074420  012746  006640                       MOV     #FMT10B,-(SP)
13932 074424  012746  000002                       MOV     #2,-(SP)
13933 074430  010600                             MOV     SP,R0                           ; SP,*
```

```
13935                                          ;MLX4
13936                                          ;
13937                                          ;       TESTING RANDOM DATA & WORD COUNTS                27-Mar-1982 19:24:42    TOPS
13938                                                                                                   27-Mar-1982 19:23:44    PA:<
13939 074432  104414                                   TRAP    14
13940 074434  062706  000006                           ADD     #6,SP
13941 074440  017766  137746  000036  24$:             MOV     @ML.REG+44,36(SP)       ; *,OLDSEC                      7344
13942 074446  017705  137736                           MOV     @ML.REG+42,R5           ;                               7345
13943 074452  006205                                   ASR     R5
13944 074454  006205                                   ASR     R5
13945 074456  006205                                   ASR     R5
13946 074460  006205                                   ASR     R5
13947 074462  006205                                   ASR     R5
13948 074464  006205                                   ASR     R5
13949 074466  042705  177700                           BIC     #177700,R5
13950 074472  010566  000042                           MOV     R5,42(SP)               ; *,OLDCHN
13951 074476  012746  000001                           MOV     #1,-(SP)                ;                               7347
13952 074502  016646  000034                           MOV     34(SP),-(SP)            ; COMMAND,*
13953 074506  010146                                   MOV     R1,-(SP)                ; LUN,*
13954 074510  010246                                   MOV     R2,-(SP)                ; WRDCNT,*
13955 074512  016646  000044                           MOV     44(SP),-(SP)            ; PTR,*
13956 074516  010346                                   MOV     R3,-(SP)                ; SECTOR,*
13957 074520  004767  151500                           JSR     PC,RETRY
13958 074524  062706  000014                           ADD     #14,SP
13959 074530  020027  000005                           CMP     R0,#5
13960 074534  001051                                   BNE     27$
13961 074536  027766  137650  000036                   CMP     @ML.REG+44,36(SP)       ; *,OLDSEC                      7350
13962 074544  001034                                   BNE     26$
13963 074546  016600  000042                           MOV     42(SP),R0               ; OLDCHN,*
13964 074552  017705  137632                           MOV     @ML.REG+42,R5
13965 074556  006205                                   ASR     R5
13966 074560  006205                                   ASR     R5
13967 074562  006205                                   ASR     R5
13968 074564  006205                                   ASR     R5
13969 074566  006205                                   ASR     R5
13970 074570  006205                                   ASR     R5
13971 074572  042705  177700                           BIC     #177700,R5
13972 074576  020500                                   CMP     R5,R0
13973 074600  001016                                   BNE     26$
13974 074602  032767  000001  105450                   BIT     #1,ERROUT               ;                               7354
13975 074610  001404                                   BEQ     25$
13976 074612  104456                                   TRAP    56
13977 074614  012131                                   .WORD   12131
13978 074616  011202                                   .WORD   MSG4
13979 074620  000000                                   .WORD   0
13980 074622  010146                          25$:     MOV     R1,-(SP)                ; LUN,*                         7357
13981 074624  016746  137512                           MOV     BOARD,-(SP)
13982 074630  004767  142566                           JSR     PC,UP.HARD.COUNT
13983 074634  000426                                   BR      29$
13984 074636  032767  000001  105414  26$:             BIT     #1,ERROUT               ;                               7350
13985 074644  001415                                   BEQ     28$                     ;                               7362
13986 074646  104457                                   TRAP    57
13987 074650  012132                                   .WORD   12132
13988 074652  011166                                   .WORD   MSG3
13989 074654  000000                                   .WORD   0
13990 074656  000410                                   BR      28$                     ;                               7365
```

SEQ 0306

```
13991
13992                                   ;MLX4
13993                                   ;                TESTING RANDOM DATA & WORD COUNTS
13994 074660  032767  000001  105372  27$:    BIT     #1,ERROUT                       ;
13995 074666  001404                          BEQ     28$
13996 074670  104457                          TRAP    57
13997 074672  012133                          .WORD   12133
13998 074674  011166                          .WORD   MSG3
13999 074676  000000                          .WORD   0
14000 074700  010146                  28$:    MOV     R1,-(SP)
14001 074702  016746  137434                  MOV     BOARD,-(SP)                     ; LUN,*
14002 074706  004767  143172                  JSR     PC,UP.SOFT.COUNT
14003 074712  022626                  29$:    CMP     (SP)+,(SP)+
14004 074714  016603  000030          30$:    MOV     30(SP),R3                       ;
14005 074720  062767  000002  135742          ADD     #2,WPTR                         ; NEXT.SECT,SECTOR
14006 074726  062706  000022                  ADD     #22,SP                          ;
14007 074732  000167  176362                  JMP     5$                              ;
14008 074736  032767  000001  105316  31$:    BIT     #1,EFN$21                       ;
14009 074744  001002                          BNE     32$                             ;
14010 074746  000167  000430                  JMP     42$
14011 074752  010100                  32$:    MOV     R1,R0
14012 074754  006300                          ASL     R0                              ; LUN,*
14013 074756  016066  034500  000024          MOV     TOP.SECT(R0),24(SP)
14014 074764  016005  034460                  MOV     LOW.SECT(R0),R5                 ; *,SECTOR
14015 074770  000577                          BR      41$
14016 074772  010146                  33$:    MOV     R1,-(SP)                        ; LUN,*
14017 074774  012746  000400                  MOV     #400,-(SP)
14018 075000  012746  022670                  MOV     #RBUFF,-(SP)
14019 075004  010546                          MOV     R5,-(SP)                        ; SECTOR,*
14020 075006  004767  150600                  JSR     PC,READ
14021 075012  062706  000010                  ADD     #10,SP
14022 075016  020027  000005                  CMP     R0,#5
14023 075022  001161                          BNE     40$
14024 075024  004767  142302                  JSR     PC,ISOLATE
14025 075030  032767  000001  105222          BIT     #1,ERROUT                       ;
14026 075036  001423                          BEQ     34$
14027 075040  017700  137344                  MOV     @ML.REG+42,R0
14028 075044  006200                          ASR     R0
14029 075046  006200                          ASR     R0
14030 075050  006200                          ASR     R0
14031 075052  006200                          ASR     R0
14032 075054  006200                          ASR     R0
14033 075056  006200                          ASR     R0
14034 075060  042700  177700                  BIC     #177700,R0
14035 075064  010046                          MOV     R0,-(SP)
14036 075066  012746  006640                  MOV     #FMT10B,-(SP)
14037 075072  012746  000002                  MOV     #2,-(SP)
14038 075076  010600                          MOV     SP,R0                           ; SP,*
14039 075100  104414                          TRAP    14
14040 075102  062706  000006                  ADD     #6,SP
14041 075106  017766  137300  000014  34$:    MOV     @ML.REG+44,14(SP)               ; *,OLDSEC
14042 075114  017700  137270                  MOV     @ML.REG+42,R0                   ;
14043 075120  006200                          ASR     R0
14044 075122  006200                          ASR     R0
14045 075124  006200                          ASR     R0
```

```
14047                                   ;MLX4
14048                                   ;
14049                                   ;       TESTING RANDOM DATA & WORD COUNTS                27-Mar-1982 19:24:42    TOPS
14050 075126  006200                                                                             27-Mar-1982 19:23:44    PA:<
14051 075130  006200                            ASR     R0
14052 075132  006200                            ASR     R0
14053 075134  042700  177700                    ASR     R0
14054 075140  010066  000020                    BIC     #177700,R0
14055 075144  012746  000001                    MOV     R0,20(SP)                       ; *,OLDCHN
14056 075150  016646  000012                    MOV     #1,-(SP)                        ;                       7424
14057 075154  010146                            MOV     12(SP),-(SP)                    ; COMMAND,*
14058 075156  012746  000400                    MOV     R1,-(SP)                        ; LUN,*
14059 075162  016646  000022                    MOV     #400,-(SP)
14060 075166  016646  000026                    MOV     22(SP),-(SP)                    ; PTR,*
14061 075172  004767  151026                    MOV     26(SP),-(SP)                    ; OLDSEC,*
14062 075176  062706  000014                    JSR     PC,RETRY
14063 075202  020027  000005                    ADD     #14,SP
14064 075206  001051                            CMP     R0,#5
14065 075210  027766  137176  000014            BNE     37$
14066 075216  001034                            CMP     @ML.REG+44,14(SP)               ; *,OLDSEC              7433
14067 075220  016646  000020                    BNE     36$
14068 075224  017700  137160                    MOV     20(SP),-(SP)                    ; OLDCHN,*
14069 075230  006200                            MOV     @ML.REG+42,R0
14070 075232  006200                            ASR     R0
14071 075234  006200                            ASR     R0
14072 075236  006200                            ASR     R0
14073 075240  006200                            ASR     R0
14074 075242  006200                            ASR     R0
14075 075244  042700  177700                    ASR     R0
14076 075250  020026                            BIC     #177700,R0
14077 075252  001016                            CMP     R0,(SP)+
14078 075254  032767  000001  104776            BNE     36$
14079 075262  001404                            BIT     #1,ERROUT                       ;                       7437
14080 075264  104456                            BEQ     35$
14081 075266  012134                            TRAP    56                              ;                       7442
14082 075270   11202                            .WORD   12134
14083 075272  000000                            .WORD   MSG4
14084 075274  010146                            .WORD   0
14085 075276  016746  137040          35$:      MOV     R1,-(SP)                        ; LUN,*                 7445
14086 075302  004767  142114                    MOV     BOARD,-(SP)
14087 075306  000426                            JSR     PC,UP.HARD.COUNT
14088 075310  032767  000001  104742            BR      39$
14089 075316  001415                    36$:    BIT     #1,ERROUT                       ;                       7433
14090 075320  104457                            BEQ     38$                             ;                       7450
14091 075322  012135                            TRAP    57                              ;                       7455
14092 075324  011166                            .WORD   12135
14093 075326  000000                            .WORD   MSG3
14094 075330  000410                            .WORD   0
14095 075332  032767  000001  104720            BR      38$
14096 075340  001404                    37$:    BIT     #1,ERROUT                       ;                       7458
14097 075342  104457                            BEQ     38$                             ;                       7464
14098 075344  012136                            TRAP    57                              ;                       7469
14099 075346  011166                            .WORD   12136
14100 075350  000000                            .WORD   MSG3
14101 075352  010146                            .WORD   0
                                        38$:    MOV     R1,-(SP)                        ; LUN,*                 7472
```

SEQ 0308

```
14103                                   ;MLX4
14104                                   ;
14105                                   ;               TESTING RANDOM DATA & WORD COUNTS                27-Mar-1982 19:24:42    TOPS
14106                                                                                                            27-Mar-1982 19:23:44    PA:<
14107 075354  016746  136762                    MOV     BOARD,-(SP)
14108 075360  004767  142520                    JSR     PC,UP.SOFT.COUNT
14109 075364  022626                    39$:    CMP     (SP)+,(SP)+
14110 075366  005205                    40$:    INC     R5                              ;
14111 075370  020566  000024            41$:    CMP     R5,24(SP)                       ; SECTOR                                7403
14112 075374  101002                            BHI     42$                             ; SECTOR,*                              7399
14113 075376  000167  177370                    JMP     33$
14114 075402  005201                    42$:    INC     R1
14115 075404  020166  000026            43$:    CMP     R1,26(SP)                       ; LUN
14116 075410  002002                            BGE     44$                             ; LUN,*                                 7184
14117 075412  000167  175562                    JMP     2$
14118 075416  005266  000022            44$:    INC     22(SP)
14119 075422  026666  000022  000046            CMP     22(SP),46(SP)                   ; COUNT                                 7181
14120 075430  003002                            BGT     45$                             ; COUNT,REPEAT
14121 075432  000167  175526                    JMP     1$
14122 075436  062706  000030            45$:    ADD     #30,SP                          ;                                       7107
14123 075442  000207                            RTS     PC
14124
14125                                   ; Routine Size:  616 words
14130                                   ; Maximum stack depth per invocation:  36 words
14131
```

SEQ 0309


```
14133 ;MLX4
14134 ;
14135
14136 ;          7486  %sbttl 'TESTING RANDOM SECTORS, DATA, WORD COUNTS'
14137 ;          7487  routine RAND3 (REPEAT) : novalue =
14138 ;          7488      begin                                          !* 1 *  START OF ROUTINE
14139 ;          7489
14140 ;          7490  !++
14141 ;          7491  !  ROUTINE:     RAND3(REPEAT)
14142 ;          7492  !
14143 ;          7493  !  PURPOSE:     TO CHECK RANDOM SECTORS
14144 ;          7494  !
14145 ;          7495  !  ARGUMENT:    REPEAT = NUMBER OF TIMES TO EXECUTE ENTIRE ROUTINE
14146 ;          7496  !
14147 ;          7497  !  THE CODE FOR 'RAND3' IN BRIEF:
14148 ;          7498  !
14149 ;          7499  !  BEGIN 1 (START OF ROUTINE)
14150 ;          7500  !  SAY ROUTINE IS RUNNING
14151 ;          7501  !  INCR COUNT FROM 1 TO REPEAT
14152 ;          7502  !  : BEGIN 2 (START OF REPEAT LOOP FOR THIS ROUTINE)
14153 ;          7503  !  : GENERATE THE RANDOM PATTERN
14154 ;          7504  !  : INCR LUN FROM 0 TO LAST
14155 ;          7505  !  : : BEGIN 3 (START OF LOGICAL UNIT SELECTION LOOP)
14156 ;          7506  !  : : INITIALIZE BUFFER POINTERS
14157 ;          7507  !  : : TESTLOOP:
14158 ;          7508  !  : : : BEGIN 4 (START OF LOOP THAT COMPLETELY TESTS 1 UNIT)
14159 ;          7509  !  : : : IF UNIT IS ACTIVE
14160 ;          7510  !  : : : THEN
14161 ;          7511  !  : : : : BEGIN 5 (START OF TEST FOR AN ACTIVE UNIT)
14162 ;          7512  !  : : : : TIMES = (HIGHEST - LOWEST)/2 + 1
14163 ;          7513  !  : : : : INCR KOUNT FROM 1 TO TIMES
14164 ;          7514  !  : : : : : BEGIN 6 (START OF COUNTING LOOP FOR SECTOR SELECTION)
14165 ;          7515  !  : : : : : CHOOSE A RANDOM WORD COUNT
14166 ;          7516  !  : : : : : SET UP BUFFER POINTERS BEFORE TRANSFER
14167 ;          7517  !  : : : : : CHOOSE A RANDOM SECTOR
14168 ;          7518  !  : : : : : CALCULATE WHERE TRANSFER WILL END (BASED ON WORD COUNT)
14169 ;          7519  !  : : : : : IF CALCULATED VALUE GTR HIGHEST
14170 ;          7520  !  : : : : : : THEN ADJUST THE WORD COUNT SO IT FITS
14171 ;          7521  !  : : : : : : :    WITHIN THE TESTABLE SECTOR LIMITS
14172 ;          7522  !  : : : : : WRITE
14173 ;          7523  !  : : : : : LOOK FOR ERRORS (IF DROP UNIT, LEAVE LABEL)
14174 ;          7524  !  : : : : : CHOOSE WHETHER TO WRITE CHECK OR READ
14175 ;          7525  !  : : : : : DO THE WRITE CHECK OR READ
14176 ;          7526  !  : : : : : LOOK FOR ERRORS (IF DROP UNIT, LEAVE LABEL)
14177 ;          7527  !  : : : : : CHANGE BUFFER POINTERS AFTER TRANSFER
14178 ;          7528  !  : : : : : END 6 (END OF COUNTING LOOP FOR SECTOR SELECTION)
14179 ;          7529  !  : : : : END 5 (END OF TEST FOR AN ACTIVE UNIT)
14180 ;          7530  !  : : : END 4 (END OF TESTLOOP)
14181 ;          7531  !  : : END 3 (END OF LOGICAL UNIT SELECTION LOOP)
14182 ;          7532  !  : END 2 (END OF REPEAT LOOP FOR THIS ROUTINE)
14183 ;          7533  !  RETURN
14184 ;          7534  !  END 1 (END OF ROUTINE)
14185 ;          7535  !--
14186 ;          7536
14187 ;          7537      local
```

```
14189 ;MLX4
14190 ;                 TESTING RANDOM SECTORS, DATA, WORD COUNTS        27-Mar-1982 19:24:42    TOPS-20 Bliss-16 V2(212)
14191                                                                    27-Mar-1982 19:23:44    PA:<NEALE>MLX4.BLI.5 (46)
14192 ;          7538          WRDCNT,
14193 ;          7539          SECTOR,
14194 ;          7540          TIMES,
14195 ;          7541          VALUE,
14196 ;          7542          OLDSEC,
14197 ;          7543          OLDCHN,
14198 ;          7544          PTR,
14199 ;          7545          COMMAND,
14200 ;          7546          DBL_VALUE;
14201 ;          7547
14202 ;          7548      label
14203 ;          7549          LOOP;
14204 ;          7550
14205 ;          7551      PRINTB (SAY1, RTN5C);
14206 ;          7552      !'RAND3'
14207 ;          7553
14208 ;          7554      incr COUNT from 1 to .REPEAT do
14209 ;          7555          begin                                        !* 2 *  START OF REPEAT LOOP FOR THIS ROUTINE
14210 ;          7556
14211 ;          7557          incr LUN from 0 to (.L$UNIT - 1) do
14212 ;          7558              begin                                    !* 3 *  START OF LOGICAL UNIT SELECTION LOOP
14213 ;          7559              GEN5 ();
14214 ;          7560  LOOP :
14215 ;          7561              begin                                    !* 4 *  START OF THE LOOP THAT COMPLETELY TESTS 1 UNIT
14216 ;          7562
14217 ;          7563          if .DRIVE_STATUS [.LUN] eql ACTIVE
14218 ;          7564          then
14219 ;          7565              begin                                    !* 5 *  START OF TEST FOR AN ACTIVE UNIT
14220 ;          7566              L$LUN = .LUN;
14221 ;          7567              WPTR = WBUFF;
14222 ;          7568              RPTR = RBUFF;
14223 ;          7569              TIMES = (HIGHEST - LOWEST)/2 + 1;
14224 ;          7570
14225 ;          7571              incr KOUNT from 1 to .TIMES do
14226 ;          7572                  begin                                !* 6 *  START OF COUNTING LOOP FOR SECTOR SELECTION
14227 ;          7573                  WRDCNT = RNDWC ();                   !VALUE BETWEEN 1 AND BUFSIZ
14228 ;          7574                  SET_PTRS (.WRDCNT);
14229 ;          7575                  SECTOR = RNDSEC (.LUN);              !VALUE BETWEEN LOWEST AND HIGHEST
14230 ;          7576
14231 ;          7577                  if (.SECTOR + .WRDCNT/256) gtra HIGHEST then WRDCNT = 256*(HIGHEST - .SECTOR + 1);
14232 ;          7578
14233 ;          7579                  VALUE = write (.LUN, .WRDCNT, .WPTR, .SECTOR);
14234 ;          7580
14235 ;          7581                  !+
14236 ;          7582                  !  SEE HOW SUCCESSFUL THE WRITE WAS:
14237 ;          7583                  !-
14238 ;          7584
14239 ;          7585                  selectone .VALUE of                  !SEE 'SYSERR' FOR DEFINITION
14240 ;          7586                      set                              !OF ERROR # CONTAINED IN 'VALUE'
14241 ;          7587
14242 ;          7588                      [1] :
14243 ;          7589                          begin                        !* 6A * RETRY ALLOWED
```

```
14245 ;MLX4
14246 ;                 TESTING RANDOM SECTORS, DATA, WORD COUNTS
14247
14248 ;          7590
14249 ;          7591                        if RETRY (SIX, write, .LUN, .WRDCNT, .WPTR, .SECTOR) neq 0
14250 ;          7592                        then                    !THE RETRY FAILED -- SYSTEM FATAL ERROR
14251 ;          7593                            begin
14252 ;          7594                            WHY_DROPT [.LUN] = CODE_4;
14253 ;          7595                            ERRDF (5301, MSG1, 0);  !**** OPTION 5, RAND3 ERROR 01 ****
14254 ;          7596                            DODU (.LUN);
14255 ;          7597                            leave LOOP;             !JUMP JUST BEYOND END OF BLOCK * 4 *
14256 ;          7598                            end;
14257 ;          7599
14258 ;          7600                        end;                        !* 6A *
14259 ;          7601
14260 ;          7602                    [2] :
14261 ;          7603                        begin                       !* 6B * FATAL CONTROLLER ERROR -- NO RETRY ALLOWED
14262 ;          7604                        WHY_DROPT [.LUN] = CODE_5;
14263 ;          7605                        ERRDF (5302, MSG1, 0);      !**** OPTION 5, RAND3 ERROR 02 ****
14264 ;          7606                        DODU (.LUN);
14265 ;          7607                        leave LOOP;                 !JUMP JUST BEYOND END OF BLOCK * 4 *
14266 ;          7608                        end;                        !* 6B *
14267 ;          7609
14268 ;          7610                    [3] :
14269 ;          7611                        begin                       !* 6C * FATAL DRIVE ERROR -- NO RETRY ALLOWED
14270 ;          7612                        ERRDF (5303, MSG1, 0);      !**** OPTION 5, RAND3 ERROR 03 ****
14271 ;          7613                        WHY_DROPT [.LUN] = CODE_6;
14272 ;          7614                        DODU (.LUN);
14273 ;          7615                        leave LOOP;                 !JUMP JUST BEYOND END OF BLOCK * 4 *
14274 ;          7616                        end;                        !* 6C *
14275 ;          7617                    tes;
14276 ;          7618
14277 ;          7619                COMMAND = CHOOSE ();
14278 ;          7620
14279 ;          7621                if .COMMAND eql read
14280 ;          7622                then
14281 ;          7623                    begin
14282 ;          7624                    PTR = .RPTR;
14283 ;          7625                    VALUE = read (.LUN, .WRDCNT, .RPTR, .SECTOR);
14284 ;          7626                    end
14285 ;          7627                else
14286 ;          7628                    begin
14287 ;          7629                    PTR = .WPTR;
14288 ;          7630                    VALUE = CHECK (.LUN, .WRDCNT, .WPTR, .SECTOR);
14289 ;          7631                    end;
14290 ;          7632
14291 ;          7633                !+
14292 ;          7634                !  SEE HOW SUCCESSFUL THE OPERATION WAS:
14293 ;          7635                !-
14294 ;          7636
14295 ;          7637                selectone .VALUE of             !SEE 'SYSERR' FOR DEFINITION
14296 ;          7638                    set                         !OF ERROR # CONTAINED IN 'VALUE'
14297 ;          7639
14298 ;          7640                    [0] :
14299 ;          7641
```

```
  14301 ;MLX4
  14302 ;                 TESTING RANDOM SECTORS, DATA, WORD COUNTS        27-Mar-1982 19:24:42    TOPS-20 Bliss-16 V2(212)
  14303 ;                                                                  27-Mar-1982 19:23:44    PA:<NEALE>MLX4.BLI.5 (46)
  14304 ;          7642                                    if .COMMAND eql read
  14305 ;          7643                                    then
  14306 ;          7644                                        begin
  14307 ;          7645
  14308 ;          7646                                        if (DBL_VALUE = DOUBLE_CHECK (.WPTR, .RPTR, .WRDCNT)) neq 0
  14309 ;          7647                                        then
  14310 ;          7648                                            begin
  14311 ;          7649                                            SAYWHO (.LUN);
  14312 ;          7650                                            PRINTB (SAY1, MSG5);          !'ECC LOGIC FAILED TO DETECT DATA ERROR'
  14313 ;          7651                                            PRINTB (FMT12A, ..DBL_VALUE, .DBL_VALUE);
  14314 ;          7652                                                          !'GOOD DATA: XXXXXX AT LOCATION YYYYYY'
  14315 ;          7653                                            DBL_VALUE = .DBL_VALUE + BUFSIZ*2;
  14316 ;          7654                                            PRINTB (FMT12B, ..DBL_VALUE, .DBL_VALUE);
  14317 ;          7655                                                          !'BAD DATA:  PPPPPP AT LOCATION QQQQQQ'
  14318 ;          7656                                            WHY_DROPT [.LUN] = CODE_8;
  14319 ;          7657                                            ERRDF (5304, MSG1, 0);        !**** OPTION 5, RAND3 ERROR 04 ****
  14320 ;          7658                                            DODU (.LUN);
  14321 ;          7659                                            leave LOOP; !JUMP JUST BEYOND END OF BLOCK * 4 *
  14322 ;          7660                                            end;
  14323 ;          7661
  14324 ;          7662                                        end;
  14325 ;          7663
  14326 ;          7664                                [1] :
  14327 ;          7665                                    begin                 !* 6D * RETRY ALLOWED
  14328 ;          7666
  14329 ;          7667                                    if RETRY (SIX, .COMMAND, .LUN, .WRDCNT, .PTR, .SECTOR) neq 0
  14330 ;          7668                                    then                  !THE RETRY FAILED -- SYSTEM FATAL ERROR
  14331 ;          7669                                        begin
  14332 ;          7670                                        WHY_DROPT [.LUN] = CODE_4;
  14333 ;          7671                                        ERRDF (5305, MSG1, 0);  !**** OPTION 5, RAND3 ERROR 05 ****
  14334 ;          7672                                        DODU (.LUN);
  14335 ;          7673                                        leave LOOP;     !JUMP JUST BEYOND END OF BLOCK * 4 *
  14336 ;          7674                                        end;
  14337 ;          7675
  14338 ;          7676                                    end;                  !* 6D *
  14339 ;          7677
  14340 ;          7678                                [2] :
  14341 ;          7679                                    begin                 !* 6E * FATAL CONTROLLER ERROR -- NO RETRY ALLOWED
  14342 ;          7680                                    WHY_DROPT [.LUN] = CODE_5;
  14343 ;          7681                                    ERRDF (5306, MSG1, 0);      !**** OPTION 5, RAND3 ERROR 06 ****
  14344 ;          7682                                    DODU (.LUN);
  14345 ;          7683                                    leave LOOP;           !JUMP JUST BEYOND END OF BLOCK * 4 *
  14346 ;          7684                                    end;
  14347 ;          7685
  14348 ;          7686                                [3] :
  14349 ;          7687                                    begin                 !* 6F * FATAL DRIVE ERROR -- NO RETRY ALLOWED
  14350 ;          7688                                    ERRDF (5307, MSG1, 0);      !**** OPTION 5, RAND3 ERROR 07 ****
  14351 ;          7689                                    WHY_DROPT [.LUN] = CODE_6;
  14352 ;          7690                                    DODU (.LUN);
  14353 ;          7691                                    leave LOOP;           !JUMP JUST BEYOND END OF BLOCK * 4 *
  14354 ;          7692                                    end;                  !* 6F *
  14355 ;          7693
```

```
14357 ;MLX4
14358 ;                 TESTING RANDOM SECTORS, DATA, WORD COUNTS        27-Mar-1982 19:24:42    TOPS-20 Bliss-16 V2(212)
14359 ;                                                                  27-Mar-1982 19:23:44    PA:<NEALE>MLX4.BLI.5 (46)
14360 ;          7694                                [4] :
14361 ;          7695                                    begin                   !* 6G * UNRECOVERABLE DATA ERROR
14362 ;          7696                                    ISOLATE ();
14363 ;          7697                                    ERRDF (5308, MSG2, 0);      :**** OPTION 5, RAND3 ERROR 08 ****
14364 ;          7698                                    WHY_DROPT [.LUN] = CODE_7;
14365 ;          7699                                    DODC (.LUN);
14366 ;          7700                                    leave LOOP;             !JUMP JUST BEYOND END OF BLOCK * 4 *
14367 ;          7701                                    end;                    !* 6G *
14368 ;          7702
14369 ;          7703                                [5] :
14370 ;          7704                                    begin                   !* 6H * RECOVERABLE DATA ERROR
14371 ;          7705                                    ISOLATE ();
14372 ;          7706
14373 ;          7707                                    if .ERROUT then PRINTB (FMT10B, .CHAN);
14374 ;          7708
14375 ;          7709                                    !'    BIT QQ'
14376 ;          7710                                    OLDSEC = .MLEL;
14377 ;          7711                                    OLDCHN = .CHAN;
14378 ;          7712
14379 ;          7713                                    if RETRY (ONE, CHECK, .LUN, .WRDCNT, .WPTR, .SECTOR) eql 5
14380 ;          7714                                    then
14381 ;          7715
14382 ;          7716                                        if ((.MLEL eql .OLDSEC) and (.CHAN eql .OLDCHN))
14383 ;          7717                                        then
14384 ;          7718                                            begin
14385 ;          7719
14386 ;          7720                                            if .ERROUT then ERRHRD (5309, MSG4, 0);
14387 ;          7721
14388 ;          7722                                                            !**** OPTION 5, RAND3 ERROR 09 ****
14389 ;          7723                                            UP_HARD_COUNT (.LUN, .BOARD);
14390 ;          7724                                            end
14391 ;          7725                                        else
14392 ;          7726                                            begin
14393 ;          7727
14394 ;          7728                                            if .ERROUT then ERRSOFT (5310, MSG3, 0);
14395 ;          7729
14396 ;          7730                                                            !**** OPTION 5, RAND3 ERROR 10 ****
14397 ;          7731                                            UP_SOFT_COUNT (.LUN, .BOARD);
14398 ;          7732                                            end
14399 ;          7733
14400 ;          7734                                    else
14401 ;          7735                                        begin
14402 ;          7736
14403 ;          7737                                        if .ERROUT then ERRSOFT (5311, MSG3, 0);
14404 ;          7738
14405 ;          7739                                                            !**** OPTION 5, RAND3 ERROR 11 ****
14406 ;          7740                                        UP_SOFT_COUNT (.LUN, .BOARD);
14407 ;          7741                                        end;
14408 ;          7742
14409 ;          7743                                    end;                    !* 6H *
14410 ;          7744                            tes;
14411 ;          7745
```

```
14413 ;MLX4
14414 ;
14415
14416 ;          7746                    WPTR = .WPTR + 2;
14417 ;          7747                    end;                            !* 6 *  END OF COUNTING LOOP FOR SECTOR SELECTION
14418 ;          7748
14419 ;          7749                end;                                !* 5 *  END OF TEST FOR AN ACTIVE UNIT
14420 ;          7750
14421 ;          7751            !+
14422 ;          7752            ! Test to see if this uut's address space is
14423 ;          7753            ! to be read for soft errors.  This test is
14424 ;          7754            ! is intended for DMT purposes.
14425 ;          7755            !-
14426 ;          7756
14427 ;          7757            if .EFN$21                              !Is the background pattern to be read
14428 ;          7758            then
14429 ;          7759                begin
14430 ;          7760
14431 ;          7761    !
14432 ;          7762    ! version czmlbb changed incr to incru
14433 ;          7763    !
14434 ;          7764                incru SECTOR from LOWEST to HIGHEST do
14435 ;          7765
14436 ;          7766                    if read (.LUN, 256, RBUFF, .SECTOR) eql 5
14437 ;          7767                    then
14438 ;          7768                        begin
14439 ;          7769                        ISOLATE ();                 !Find the failing bank and board no.
14440 ;          7770
14441 ;          7771                        if .ERROUT then PRINTB (FMT10B, .CHAN);
14442 ;          7772
14443 ;          7773                        ! Print where the error is
14444 ;          7774                        !
14445 ;          7775                        ! Save the contents of the ML error location
14446 ;          7776                        ! register so we can compare it to the new
14447 ;          7777                        ! contents of this register after the retry.
14448 ;          7778                        ! This is done to classify the error.
14449 ;          7779                        !
14450 ;          7780                        OLDSEC = .MLEL;
14451 ;          7781                        OLDCHN = .CHAN;
14452 ;          7782                        !
14453 ;          7783                        ! Do a classify retry call.  If the same error
14454 ;          7784                        ! occures then classify it as a hard error.  If
14455 ;          7785                        ! a different error occured or the error went away
14456 ;          7786                        ! then classify it as a soft error.
14457 ;          7787                        !
14458 ;          7788
14459 ;          7789                        if RETRY (ONE, .COMMAND, .LUN, 256, .PTR, .OLDSEC) eql 5
14460 ;          7790                        then
14461 ;          7791                        !
14462 ;          7792                        ! The same error occured so see if it is at the same
14463 ;          7793                        ! sector and channel number, if so then classify
14464 ;          7794                        ! it as a hard error else classiy it as a soft
14465 ;          7795                        ! error.
14466 ;          7796                        !
14467 ;          7797
```

```
14469 ;MLX4
14470 ;
14471
14472 ;          7798                                   if ((.MLEL eql .OLDSEC) and (.CHAN eql .OLDCHN))
14473 ;          7799                                   then
14474 ;          7800                                       begin            !Same error occured 'hard'
14475 ;          7801
14476 ;          7802                                       if .ERROUT       !Print error if enabled
14477 ;          7803                                       then
14478 ;          7804                                           begin
14479 ;        P 7805                                           ERRHRD (5312,       !Error number
14480 ;        P 7806                                               MSG4,   !Error message
14481 ;          7807                                               0);     !Additional message routine
14482 ;          7808                                           end;
14483 ;          7809
14484 ;          7810                                       UP_HARD_COUNT (.LUN, .BOARD);
14485 ;          7811                                       end
14486 ;          7812                                   else
14487 ;          7813                                       begin            !Not the same error 'soft'
14488 ;          7814
14489 ;          7815                                       if .ERROUT       !Print error if enabled
14490 ;          7816                                       then
14491 ;          7817                                           begin
14492 ;        P 7818                                           ERRSOFT (5313,      !Error number
14493 ;        P 7819                                               MSG3,   !Error message
14494 ;          7820                                               0);     !Additional message routine
14495 ;          7821                                           end;
14496 ;          7822
14497 ;          7823                                       UP_SOFT_COUNT (.LUN, .BOARD);
14498 ;          7824                                       end
14499 ;          7825
14500 ;          7826                               else                     !Not the same error 'soft'
14501 ;          7827                                   begin
14502 ;          7828
14503 ;          7829                                   if .ERROUT           !Print error if enabled
14504 ;          7830                                   then
14505 ;          7831                                       begin
14506 ;        P 7832                                       ERRSOFT (5314,  !Error number
14507 ;        P 7833                                           MSG3,       !Error message
14508 ;          7834                                           0);         !Additional message routine
14509 ;          7835                                       end;
14510 ;          7836
14511 ;          7837                                   UP_SOFT_COUNT (.LUN, .BOARD);
14512 ;          7838                                   end;
14513 ;          7839
14514 ;          7840                               end;
14515 ;          7841
14516 ;          7842                       end;
14517 ;          7843
14518 ;          7844                   end;                                 !* 4 *  END OF LOOP THAT COMPLETELY TESTS 1 UNIT
14519 ;          7845                   end;                                 !* 3 *  END OF LOGICAL UNIT SELECTION LOOP
14520 ;          7846
14521 ;          7847               end;                                     !* 2 *  END OF REPEAT LOOP FOR THIS ROUTINE
14522 ;          7848
14523 ;          7849           return;
```

```
14525 ;MLX4
14526 ;                 TESTING RANDOM SECTORS, DATA, WORD COUNTS        27-Mar-1982 19:24:42    TOPS-20 Bliss-16 V2(212)
14527                                                                    27-Mar-1982 19:23:44    PA:<NEALE>MLX4.BLI.5 (46)
14528 ;         7850        end;                                         !* 1 *  END OF ROUTINE
14532
14533
14537 075444  004167  107664            RAND3:  .SBTTL  RAND3 TESTING RANDOM SECTORS, DATA, WORD COUNTS
14538 075450  162706  000024                    JSR     R1,$SAVE5                                                    ;                                7487
14539 075454  012746  007700                    SUB     #24,SP
14540 075460  012746  007072                    MOV     #RTN5C,-(SP)                                                 ;                                7551
14541 075464  012746  000002                    MOV     #SAY1,-(SP)
14542 075470  010600                            MOV     #2,-(SP)
14543 075472  104414                            MOV     SP,R0                                                        ; SP,*
14544 075474  005066  000030                    TRAP    14
14545 075500  000167  002254                    CLR     30(SP)                                                       ; COUNT
14546 075504  016766  104302  000012    1$:     JMP     44$                                                                                           7554
14547 075512  005003                            MOV     L$UNIT,12(SP)                                                ;                                7557
14548 075514  000167  002226                    CLR     R3                                                           ; LUN
14549 075520  004767  144074            2$:     JMP     43$
14550 075524  010300                            JSR     PC,GEN5                                                      ;                                7559
14551 075526  006200                            MOV     R3,R0                                                        ; LUN,*                          7563
14552 075530  006200                            ASR     R0
14553 075532  006200                            ASR     R0
14554 075534  062700  034442                    ASR     R0
14555 075540  010046                            ADD     #DRIVE.STATUS,R0
14556 075542  010346                            MOV     R0,-(SP)
14557 075544  042716  177770                    MOV     R3,-(SP)                                                     ; LUN,*
14558 075550  012746  000001                    BIC     #177770,(SP)
14559 075554  005046                            MOV     #1,-(SP)
14560 075556  004767  106574                    CLR     -(SP)
14561 075562  062706  000010                    JSR     PC,BL$GT2
14562 075566  005300                            ADD     #10,SP
14563 075570  001402                            DEC     R0
14564 075572  000167  001502                    BEQ     3$
14565 075576  010367  104272                    JMP     31$
14566 075602  012767  012670  135060    3$:     MOV     R3,L$LUN                                                     ; LUN,*                          7566
14567 075610  012767  022670  135054            MOV     #WBUFF,WPTR                                                  ;                                7567
14568 075616  010300                            MOV     #RBUFF,RPTR                                                  ;                                7568
14569 075620  006300                            MOV     R3,R0                                                        ; LUN,*                          7569
14570 075622  012766  034500  000010            ASL     R0
14571 075630  060066  000010                    MOV     #TOP.SECT,10(SP)
14572 075634  017646  000010                    ADD     R0,10(SP)
14573 075640  166016  034460                    MOV     @10(SP),-(SP)
14574 075644  012746  000002                    SUB     LOW.SECT(R0),(SP)
14575 075650  004767  107314                    MOV     #2,-(SP)
14576 075654  010066  000020                    JSR     PC,BL$DIV
14577 075660  005266  000020                    MOV     R0,20(SP)                                                    ; *,TIMES
14578 075664  005066  000012                    INC     20(SP)                                                       ; TIMES
14579 075670  000167  001362                    CLR     12(SP)                                                       ; KOUNT                          7571
                                                JMP     29$
```

```
14581                                   ;MLX4
14582                                   :                                                       27-Mar-1982 19:24:42    TOPS
14583                                   :       TESTING RANDOM SECTORS, DATA, WORD COUNTS       27-Mar-1982 19:23:44    PA:<
14584 075674  004767  172304          4$:     JSR     PC,RNDWC
14585 075700  010002                          MOV     R0,R2                   ; *,WRDCNT              7573
14586 075702  010246                          MOV     R2,-(SP)                ; WRDCNT,*
14587 075704  004767  151222                  JSR     PC,SET.PTRS             ;                       7574
14588 075710  010316                          MOV     R3,(SP)
14589 075712  004767  172322                  JSR     PC,RNDSEC               ; LUN,*                 7575
14590 075716  010001                          MOV     R0,R1
14591 075720  010246                          MOV     R2,-(SP)                ; *,SECTOR
14592 075722  012746  000400                  MOV     #400,-(SP)              ; WRDCNT,*              7577
14593 075726  004767  107236                  JSR     PC,BL$DIV
14594 075732  022626                          CMP     (SP)+,(SP)+
14595 075734  060100                          ADD     R1,R0
14596 075736  020076  000016                  CMP     R0,@16(SP)              ; SECTOR,*
14597 075742  101410                          BLOS    5$
14598 075744  017600  000016                  MOV     @16(SP),R0
14599 075750  160100                          SUB     R1,R0                   ; SECTOR,*
14600 075752  000300                          SWAB    R0
14601 075754  105000                          CLRB    R0
14602 075756  010002                          MOV     R0,R2                   ; *,WRDCNT
14603 075760  062702  000400                  ADD     #400,R2                 ; *,WRDCNT
14604 075764  010316                  5$:     MOV     R3,(SP)                 ; LUN,*                 7579
14605 075766  010246                          MOV     R2,-(SP)                ; WRDCNT,*
14606 075770  016746  134674                  MOV     WPTR,-(SP)
14607 075774  010146                          MOV     R1,-(SP)                ; SECTOR,*
14608 075776  004767  147422                  JSR     PC,WRITE
14609 076002  010004                          MOV     R0,R4                   ; *,VALUE
14610 076004  020427  000001                  CMP     R4,#1                   ; VALUE,*               7585
14611 076010  001031                          BNE     6$
14612 076012  012746  000006                  MOV     #6,-(SP)                ;                       7591
14613 076016  012746  045424                  MOV     #WRITE,-(SP)
14614 076022  010346                          MOV     R3,-(SP)                ; LUN,*
14615 076024  010246                          MOV     R2,-(SP)                ; WRDCNT,*
14616 076026  016746  134636                  MOV     WPTR,-(SP)
14617 076032  010146                          MOV     R1,-(SP)                ; SECTOR,*
14618 076034  004767  150164                  JSR     PC,RETRY
14619 076040  062706  000014                  ADD     #14,SP
14620 076044  005700                          TST     R0
14621 076046  001446                          BEQ     9$
14622 076050  112763  000004  034446          MOVB    #4,WHY.DROPT(R3)        ; *,*(LUN)              7594
14623 076056  104455                          TRAP    55                      ;                       7595
14624 076060  012265                          .WORD   12265
14625 076062  011070                          .WORD   MSG1
14626 076064  000000                          .WORD   0
14627 076066  010300                          MOV     R3,R0                   ; LUN,*                 7596
14628 076070  104451                          TRAP    51
14629 076072  000431                          BR      8$                      ;                       7597
14630 076074  020427  000002          6$:     CMP     R4,#2                   ; VALUE,*               7585
14631 076100  001012                          BNE     7$
14632 076102  112763  000005  034446          MOVB    #5,WHY.DROPT(R3)        ; *,*(LUN)              7604
14633 076110  104455                          TRAP    55                      ;                       7605
14634 076112  012266                          .WORD   12266
14635 076114  011070                          .WORD   MSG1
```

```
                                        ;MLX4
                                        ;                TESTING RANDOM SECTORS, DATA, WORD COUNTS      27-Mar-1982 19:24:42   TOPS
                                                                                                       27-Mar-1982 19:23:44   PA:
076116 000000                                    .WORD   0
076120 010300                                    MOV     R3,R0
076122 104451                                    TRAP    51                      ; LUN,*                                      7606
076124 000414                                    BR      8$
076126 020427 000003             7$:             CMP     R4,#3                   ;                                            7607
076132 001014                                    BNE     9$                      ; VALUE,*                                    7585
076134 104455                                    TRAP    55
076136 012267                                    .WORD   12267                   ;                                            7612
076140 011070                                    .WORD   MSG1
076142 000000                                    .WORD   0
076144 112763 000006 034446                      MOVB    #6,WHY.DROPT(R3)        ; *,*(LUN)                                   7613
076152 010300                                    MOV     R3,R0                   ; LUN,*                                      7614
076154 104451                                    TRAP    51
076156 062706 000014             8$:             ADD     #14,SP                  ;                                            7615
076162 000535                                    BR      14$
076164 004767 147776             9$:             JSR     PC,CHOOSE               ;                                            7619
076170 010066 000036                             MOV     R0,36(SP)               ; *,COMMAND
076174 005005                                    CLR     R5                      ;                                            7621
076176 020027 045612                             CMP     R0,#READ                ; COMMAND,*
076202 001014                                    BNE     10$
076204 005205                                    INC     R5
076206 016766 134460 000034                      MOV     RPTR,34(SP)             ; *,PTR                                      7624
076214 010346                                    MOV     R3,-(SP)                ; LUN,*                                      7625
076216 010246                                    MOV     R2,-(SP)                ; WRDCNT,*
076220 016746 134446                             MOV     RPTR,-(SP)
076224 010146                                    MOV     R1,-(SP)                ; SECTOR,*
076226 004767 147360                             JSR     PC,READ
076232 000412                                    BR      11$
076234 016766 134430 000034      10$:            MOV     WPTR,34(SP)             ; *,PTR                                      7629
076242 010346                                    MOV     R3,-(SP)                ; LUN,*                                      7630
076244 010246                                    MOV     R2,-(SP)                ; WRDCNT,*
076246 016746 134416                             MOV     WPTR,-(SP)
076252 010146                                    MOV     R1,-(SP)                ; SECTOR,*
076254 004767 147520                             JSR     PC,CHECK
076260 010004                    11$:            MOV     R0,R4                   ; *,VALUE
076262 001076                                    BNE     15$                     ;
076264 006005                                    ROR     R5                      ;                                            7637
076266 103402                                    BLO     13$                     ;                                            7642
076270 000167 000750             12$:            JMP     28$
076274 016746 134370             13$:            MOV     WPTR,-(SP)              ;                                            7646
076300 016746 134366                             MOV     RPTR,-(SP)
076304 010246                                    MOV     R2,-(SP)                ; WRDCNT,*
076306 004767 146014                             JSR     PC,DOUBLE.CHECK
076312 062706 000006                             ADD     #6,SP
076316 010066 000050                             MOV     R0,50(SP)               ; *,DBL.VALUE
076322 001762                                    BEQ     12$
076324 010346                                    MOV     R3,-(SP)                ; LUN,*                                      7649
076326 004767 137172                             JSR     PC,SAYWHO
076332 012716 011216                             MOV     #MSG5,(SP)              ;                                            7650
076336 012746 007072                             MOV     #SAY1,-(SP)
076342 012746 000002                             MOV     #2,-(SP)
076346 010600                                    MOV     SP,R0                   ; SP,*
```

```
                                        ;MLX4
                                        ;                TESTING RANDOM SECTORS, DATA, WORD COUNTS        27-Mar-1982 19:24:42   TOPS
                                                                                                         27-Mar-1982 19:23:44   PA:<
076350  104414                                  TRAP    14
076352  016616  000056                          MOV     56(SP),(SP)
076356  017646  000056                          MOV     @56(SP),-(SP)           ; DBL.VALUE,*                          7651
076362  012746  006710                          MOV     #FMT12A,-(SP)           ; DBL.VALUE,*
076366  012746  000003                          MOV     #3,-(SP)
076372  010600                                  MOV     SP,R0
076374  104414                                  TRAP    14                      ; SP,*
076376  062766  010000  000064                  ADD     #10000,64(SP)           ; *,DBL.VALUE                          7653
076404  016616  000064                          MOV     64(SP),(SP)             ; DBL.VALUE,*                          7654
076410  017646  000064                          MOV     @64(SP),-(SP)           ; DBL.VALUE,*
076414  012746  006756                          MOV     #FMT12B,-(SP)
076420  012746  000003                          MOV     #3,-(SP)
076424  010600                                  MOV     SP,R0
076426  104414                                  TRAP    14                      ; SP,*
076430  112763  000010  034446                  MOVB    #10,WHY.DROPT(R3)       ; *,*(LUN)                             7656
076436  104455                                  TRAP    55                      ;                                      7657
076440  012270                                  .WORD   12270
076442  011070                                  .WORD   MSG1
076444  000000                                  .WORD   0
076446  010300                                  MOV     R3,R0                   ; LUN,*                                7658
076450  104451                                  TRAP    51
076452  062706  000046                          ADD     #46,SP                  ;                                      7659
076456  000506                          14$:    BR      20$
076460  020427  000001                  15$:    CMP     R4,#1                   ; VALUE,*                              7637
076464  001031                                  BNE     16$
076466  012746  000006                          MOV     #6,-(SP)                ;                                      7667
076472  016646  000050                          MOV     50(SP),-(SP)            ; COMMAND,*
076476  010346                                  MOV     R3,-(SP)                ; LUN,*
076500  010246                                  MOV     R2,-(SP)                ; WRDCNT,*
076502  016646  000054                          MOV     54(SP),-(SP)            ; PTR,*
076506  010146                                  MOV     R1,-(SP)                ; SECTOR,*
076510  004767  147510                          JSR     PC,RETRY
076514  062706  000014                          ADD     #14,SP
076520  005700                                  TST     R0
076522  001662                                  BEQ     12$
076524  112763  000004  034446                  MOVB    #4,WHY.DROPT(R3)        ; *,*(LUN)                             7670
076532  104455                                  TRAP    55                      ;                                      7671
076534  012271                                  .WORD   12271
076536  011070                                  .WORD   MSG1
076540  000000                                  .WORD   0
076542  010300                                  MOV     R3,R0                   ; LUN,*                                7672
076544  104451                                  TRAP    51
076546  000450                                  BR      19$                     ;                                      7673
076550  020427  000002                  16$:    CMP     R4,#2                   ; VALUE,*                              7637
076554  001012                                  BNE     17$
076556  112763  000005  034446                  MOVB    #5,WHY.DROPT(R3)        ; *,*(LUN)                             7680
076564  104455                                  TRAP    55                      ;                                      7681
076566  012272                                  .WORD   12272
076570  011070                                  .WORD   MSG1
076572  000000                                  .WORD   0
076574  010300                                  MOV     R3,R0                   ; LUN,*                                7682
076576  104451                                  TRAP    51
```

```
14749                                   ;MLX4
14750                                   ;                TESTING RANDOM SECTORS, DATA, WORD COUNTS      27-Mar-1982 19:24:42   TOPS
14751                                                                                                  27-Mar-1982 19:23:44   PA:<
14752 076600 000433                             BR      19$
14753 076602 020427 000003              17$:    CMP     R4,#3                   ;                                              7683
14754 076606 001012                             BNE     18$                     ; VALUE,*                                      7637
14755 076610 104455                             TRAP    55                      ;
14756 076612 012273                             .WORD   12273                   ;                                              7688
14757 076614 011070                             .WORD   MSG1
14758 076616 000000                             .WORD   0
14759 076620 112763 000006 034446               MOVB    #6,WHY.DROPT(R3)        ; *,*(LUN)
14760 076626 010300                             MOV     R3,R0                   ; LUN,*                                        7689
14761 076630 104451                             TRAP    51                                                                     7690
14762 076632 000416                             BR      19$                     ;
14763 076634 020427 000004              18$:    CMP     R4,#4                   ;                                              7691
14764 076640 001017                             BNE     21$                     ; VALUE,*                                      7637
14765 076642 004767 140464                      JSR     PC,ISOLATE              ;
14766 076646 104455                             TRAP    55                      ;                                              7696
14767 076650 012274                             .WORD   12274                   ;                                              7697
14768 076652 011120                             .WORD   MSG2
14769 076654 000000                             .WORD   0
14770 076656 112763 000007 034446               MOVB    #7,WHY.DROPT(R3)        ; *,*(LUN)
14771 076664 010300                             MOV     R3,R0                   ; LUN,*                                        7698
14772 076666 104451                             TRAP    51                                                                     7699
14773 076670 062706 000024              19$:    ADD     #24,SP                  ;                                              7700
14774 076674 000167 001044              20$:    JMP     42$
14775 076700 020427 000005              21$:    CMP     R4,#5                   ; VALUE,*                                      7637
14776 076704 001157                             BNE     28$
14777 076706 004767 140420                      JSR     PC,ISOLATE              ;
14778 076712 032767 000001 103340               BIT     #1,ERROUT               ;                                              7705
14779 076720 001423                             BEQ     22$                     ;                                              7707
14780 076722 017705 135462                      MOV     @ML.REG+42,R5
14781 076726 006205                             ASR     R5
14782 076730 006205                             ASR     R5
14783 076732 006205                             ASR     R5
14784 076734 006205                             ASR     R5
14785 076736 006205                             ASR     R5
14786 076740 006205                             ASR     R5
14787 076742 042705 177700                      BIC     #177700,R5
14788 076746 010546                             MOV     R5,-(SP)
14789 076750 012746 006640                      MOV     #FMT10B,-(SP)
14790 076754 012746 000002                      MOV     #2,-(SP)
14791 076760 010600                             MOV     SP,R0                   ; SP,*
14792 076762 104414                             TRAP    14
14793 076764 062706 000006                      ADD     #6,SP
14794 076770 017766 135416 000052       22$:    MOV     @ML.REG+44,52(SP)       ; *,OLDSEC                                     7710
14795 076776 017705 135406                      MOV     @ML.REG+42,R5           ;                                              7711
14796 077002 006205                             ASR     R5
14797 077004 006205                             ASR     R5
14798 077006 006205                             ASR     R5
14799 077010 006205                             ASR     R5
14800 077012 006205                             ASR     R5
14801 077014 006205                             ASR     R5
14802 077016 042705 177700                      BIC     #177700,R5
14803 077022 010566 000042                      MOV     R5,42(SP)               ; *,OLDCHN
```

```
14805                                   ;MLX4
14806                                   ;                TESTING RANDOM SECTORS, DATA, WORD COUNTS      27-Mar-1982 19:24:42   TOPS
14807                                                                                                   27-Mar-1982 19:23:44   PA:<
14808 077026  012746  000001                      MOV     #1,-(SP)
14809 077032  012746  046000                      MOV     #CHECK,-(SP)            ;                                      7713
14810 077036  010346                              MOV     R3,-(SP)
14811 077040  010246                              MOV     R2,-(SP)                ; LUN,*
14812 077042  016746  133622                      MOV     WPTR,-(SP)              ; WRDCNT,*
14813 077046  010146                              MOV     R1,-(SP)
14814 077050  004767  147150                      JSR     PC,RETRY                ; SECTOR,*
14815 077054  062706  000014                      ADD     #14,SP
14816 077060  020027  000005                      CMP     R0,#5
14817 077064  001051                              BNE     25$
14818 077066  027766  135320  000052              CMP     @ML.REG+44,52(SP)       ; *,OLDSEC                             7716
14819 077074  001034                              BNE     24$
14820 077076  016600  000042                      MOV     42(SP),R0               ; OLDCHN,*
14821 077102  017705  135302                      MOV     @ML.REG+42,R5
14822 077106  006205                              ASR     R5
14823 077110  006205                              ASR     R5
14824 077112  006205                              ASR     R5
14825 077114  006205                              ASR     R5
14826 077116  006205                              ASR     R5
14827 077120  006205                              ASR     R5
14828 077122  042705  177700                      BIC     #177700,R5
14829 077126  020500                              CMP     R5,R0
14830 077130  001016                              BNE     24$
14831 077132  032767  000001  103120              BIT     #1,ERROUT               ;                                      7720
14832 077140  001404                              BEQ     23$
14833 077142  104456                              TRAP    56
14834 077144  012275                              .WORD   12275
14835 077146  011202                              .WORD   MSG4
14836 077150  000000                              .WORD   0
14837 077152  010346                  23$:        MOV     R3,-(SP)
14838 077154  016746  135162                      MOV     BOARD,-(SP)             ; LUN,*                                7723
14839 077160  004767  140236                      JSR     PC,UP.HARD.COUNT
14840 077164  000426                              BR      27$
14841 077166  032767  000001  103064  24$:        BIT     #1,ERROUT               ;                                      7716
14842 077174  001415                              BEQ     26$                     ;                                      7728
14843 077176  104457                              TRAP    57
14844 077200  012276                              .WORD   12276
14845 077202  011166                              .WORD   MSG3
14846 077204  000000                              .WORD   0
14847 077206  000410                              BR      26$
14848 077210  032767  000001  103042  25$:        BIT     #1,ERROUT               ;                                      7731
14849 077216  001404                              BEQ     26$                     ;                                      7737
14850 077220  104457                              TRAP    57
14851 077222  012277                              .WORD   12277
14852 077224  011166                              .WORD   MSG3
14853 077226  000000                              .WORD   0
14854 077230  010346                  26$:        MOV     R3,-(SP)
14855 077232  016746  135104                      MOV     BOARD,-(SP)             ; LUN,*                                7740
14856 077236  004767  140642                      JSR     PC,UP.SOFT.COUNT
14857 077242  022626                  27$:        CMP     (SP)+,(SP)+
14858 077244  062767  000002  133416  28$:        ADD     #2,WPTR                 ;                                      7704
14859 077252  062706  000020                      ADD     #20,SP                  ;                                      7746
                                                                                                                        7572
```

```
                                        ;MLX4
                                        ;                TESTING RANDOM SECTORS, DATA, WORD COUNTS        27-Mar-1982 19:24:42   TOPS
                                                                                                         27-Mar-1982 19:23:44   PA:<
077256 005266 000012                29$:  INC   12(SP)                                                                          7571
077262 026666 000012 000020               CMP   12(SP),20(SP)          ; KOUNT,TIMES
077270 003002                             BGT   30$
077272 000167 176376                      JMP   4$
077274 022626                       30$:  CMP   (SP)+,(SP)+
077300 032767 000001 102754         31$:  BIT   #1,EFN$21              ;                                                        7565
077306 001002                             BNE   32$                    ;                                                        7757
077310 000167 000430                      JMP   42$
077314 010300                       32$:  MOV   R3,R0                  ; LUN,*                                                  7764
077316 006300                             ASL   R0
077320 016066 034500 000010               MOV   TOP.SECT(R0),10(SP)
077326 016005 034460                      MOV   LOW.SECT(R0),R5        ; *,SECTOR
077332 000577                             BR    41$
077334 010346                       33$:  MOV   R3,-(SP)               ; LUN,*                                                  7766
077336 012746 000400                      MOV   #400,-(SP)
077342 012746 022670                      MOV   #RBUFF,-(SP)
077346 010546                             MOV   R5,-(SP)               ; SECTOR,*
077350 004767 146236                      JSR   PC,READ
077354 062706 000010                      ADD   #10,SP
077360 020027 000005                      CMP   R0,#5
077364 001161                             BNE   40$
077366 004767 137740                      JSR   PC,ISOLATE
077372 032767 000001 102660               BIT   #1,ERROUT              ;                                                        7769
077400 001423                             BEQ   34$                    ;                                                        7771
077402 017700 135002                      MOV   @ML.REG+42,R0
077406 006200                             ASR   R0
077410 006200                             ASR   R0
077412 006200                             ASR   R0
077414 006200                             ASR   R0
077416 006200                             ASR   R0
077420 006200                             ASR   R0
077422 042700 177700                      BIC   #177700,R0
077426 010046                             MOV   R0,-(SP)
077430 012746 006640                      MOV   #FMT10B,-(SP)
077434 012746 000002                      MOV   #2,-(SP)
077440 010600                             MOV   SP,R0                  ; SP,*
077442 104414                             TRAP  14
077444 062706 000006                      ADD   #6,SP
077450 017766 134736 000026 34$:  MOV   @ML.REG+44,26(SP)              ; *,OLDSEC                                               7780
077456 017700 134726                      MOV   @ML.REG+42,R0          ;                                                        7781
077462 006200                             ASR   R0
077464 006200                             ASR   R0
077466 006200                             ASR   R0
077470 006200                             ASR   R0
077472 006200                             ASR   R0
077474 006200                             ASR   R0
077476 042700 177700                      BIC   #177700,R0
077502 010066 000016                      MOV   R0,16(SP)              ; *,OLDCHN
077506 012746 000001                      MOV   #1,-(SP)               ;
077512 016646 000024                      MOV   24(SP),-(SP)           ; COMMAND,*                                              7789
077516 010346                             MOV   R3,-(SP)               ; LUN,*
077520 012746 000400                      MOV   #400,-(SP)
```

SEQ 0323

```
14917                                   ;MLX4
14918                                   :                                                    27-Mar-1982 19:24:42    TOPS
14919                                                TESTING RANDOM SECTORS, DATA, WORD COUNTS   27-Mar-1982 19:23:44    PA:<
14920 077524  016646  000030                        MOV     30(SP),-(SP)                    ; PTR,*
14921 077530  016646  000040                        MOV     40(SP),-(SP)                    ; OLDSEC,*
14922 077534  004767  146464                        JSR     PC,RETRY
14923 077540  062706  000014                        ADD     #14,SP
14924 077544  020027  000005                        CMP     R0,#5
14925 077550  001051                                BNE     37$
14926 077552  027766  134634  000026                CMP     @ML.REG+44,26(SP)               ; *,OLDSEC                      7798
14927 077560  001034                                BNE     36$
14928 077562  016646  000016                        MOV     16(SP),-(SP)                    ; OLDCHN,*
14929 077566  017700  134616                        MOV     @ML.REG+42,R0
14930 077572  006200                                ASR     R0
14931 077574  006200                                ASR     R0
14932 077576  006200                                ASR     R0
14933 077600  006200                                ASR     R0
14934 077602  006200                                ASR     R0
14935 077604  006200                                ASR     R0
14936 077606  042700  177700                        BIC     #177700,R0
14937 077612  020026                                CMP     R0,(SP)+
14938 077614  001016                                BNE     36$
14939 077616  032767  000001  102434                BIT     #1,ERROUT                       ;                               7802
14940 077624  001404                                BEQ     35$
14941 077626  104456                                TRAP    56                              ;                               7807
14942 077630  012300                                .WORD   12300
14943 077632  011202                                .WORD   MSG4
14944 077634  000000                                .WORD   0
14945 077636  010346                     35$:       MOV     R3,-(SP)                        ; LUN,*                         7810
14946 077640  016746  134476                        MOV     BOARD,-(SP)
14947 077644  004767  137552                        JSR     PC,UP.HARD.COUNT
14948 077650  000426                                BR      39$
14949 077652  032767  000001  102400     36$:       BIT     #1,ERROUT                       ;                               7798
14950 077660  001415                                BEQ     38$                             ;                               7815
14951 077662  104457                                TRAP    57
14952 077664  012301                                .WORD   12301                           ;                               7820
14953 077666  011166                                .WORD   MSG3
14954 077670  000000                                .WORD   0
14955 077672  000410                                BR      38$
14956 077674  032767  000001  102356     37$:       BIT     #1,ERROUT                       ;                               7823
14957 077702  001404                                BEQ     38$                             ;                               7829
14958 077704  104457                                TRAP    57
14959 077706  012302                                .WORD   12302                           ;                               7834
14960 077710  011166                                .WORD   MSG3
14961 077712  000000                                .WORD   0
14962 077714  010346                     38$:       MOV     R3,-(SP)                        ; LUN,*                         7837
14963 077716  016746  134420                        MOV     BOARD,-(SP)
14964 077722  004767  140156                        JSR     PC,UP.SOFT.COUNT
14965 077726  022626                     39$:       CMP     (SP)+,(SP)+
14966 077730  005205                     40$:       INC     R5                              ;                               7768
14967 077732  020566  000010            41$:       CMP     R5,10(SP)                       ; SECTOR                        7764
14968 077736  101002                                BHI     42$                             ; SECTOR,*
14969 077740  000167  177370                        JMP     33$
14970 077744  005203                     42$:       INC     R3                              ; LUN
14971 077746  020366  000012            43$:       CMP     R3,12(SP)                       ; LUN,*                         7557
```

SEQ 0324

```
                                        ;MLX4
                                        ;                                                    27-Mar-1982 19:24:42    TOPS
                                        ;               TESTING RANDOM SECTORS. DATA, WORD COUNTS    27-Mar-1982 19:23:44    PA:<
077752  002002                                BGE     44$
077754  000167  175540                      JMP     2$
077760  005266  000030                44$:    INC     30(SP)                               ; COUNT                          7554
077764  026666  000030  000050              CMP     30(SP),50(SP)                        ; COUNT,REPEAT
077772  003002                                BGT     45$
077774  000167  175504                      JMP     1$
100000  062706  000032                45$:    ADD     #32,SP                               ;                                7487
100004  000207                                RTS     PC

                                        ; Routine Size:  625 words
                                        ; Maximum stack depth per invocation:  38 words
```

SEQ 0325

27-Mar-1982 19:24:42 TOPS-20 Bliss-16 V2(212)
27-Mar-1982 19:23:44 PA:<NEALE>MLX4.BLI.5 (47)

```
14994 ;MLX4
14995 ;                 TESTING RANDOM UNITS, SECTORS, DATA, WORD COUNT
14996
14997 ;          7851  %sbttl 'TESTING RANDOM UNITS, SECTORS, DATA, WORD COUNTS'
14998 ;          7852  routine RAND4 (REPEAT) : novalue =
14999 ;          7853      begin                                       !* 1 *  START OF ROUTINE
15000 ;          7854
15001 ;          7855  !++
15002 ;          7856  !   ROUTINE:      RAND4(REPEAT)
15003 ;          7857  !
15004 ;          7858  !   PURPOSE:      TO CHECK RANDOM UNITS
15005 ;          7859  !
15006 ;          7860  !   ARGUMENT:     REPEAT = NUMBER OF TIMES TO EXECUTE ENTIRE ROUTINE
15007 ;          7861  !
15008 ;          7862  !   THE CODE FOR 'RAND4' IN BRIEF:
15009 ;          7863  !
15010 ;          7864  !   BEGIN 1 (START OF ROUTINE)
15011 ;          7865  !   SAY ROUTINE IS RUNNING
15012 ;          7866  !   INCR COUNT FROM 1 TO REPEAT
15013 ;          7867  !   : BEGIN 2 (START OF REPEAT LOOP FOR THIS ROUTINE)
15014 ;          7868  !   : GENERATE THE RANDOM PATTERN
15015 ;          7869  !   : TIMES = NUMBER OF UNITS * 4
15016 ;          7870  !   : INCR KOUNT FROM 1 TO TIMES
15017 ;          7871  !   : : BEGIN 3 (START OF COUNTING LOOP FOR UNIT SELECTION)
15018 ;          7872  !   : : CHOOSE A RANDOM LOGICAL UNIT WHICH IS ACTIVE
15019 ;          7873  !   : : INITIALIZE BUFFER POINTERS
15020 ;          7874  !   : : TESTLOOP:
15021 ;          7875  !   : : : BEGIN 4 (START OF LOOP THAT COMPLETELY TESTS 1 UNIT)
15022 ;          7876  !   : : : INCR KOUNT2 FROM 1 TO 10
15023 ;          7877  !   : : : : BEGIN 5 (START OF COUNTING LOOP FOR SECTOR SELECTION)
15024 ;          7878  !   : : : : CHOOSE A RANDOM WORD COUNT
15025 ;          7879  !   : : : : SET UP BUFFER POINTERS BEFORE TRANSFER
15026 ;          7880  !   : : : : CHOOSE A RANDOM SECTOR
15027 ;          7881  !   : : : : CALCULATE WHERE TRANSFER WILL END (BASED ON WORD COUNT)
15028 ;          7882  !   : : : : IF CALCULATED VALUE GTR HIGHEST
15029 ;          7883  !   : : : : : THEN ADJUST THE WORD COUNT SO IT FITS
15030 ;          7884  !   : : : : : :    WITHIN THE TESTABLE SECTOR LIMITS
15031 ;          7885  !   : : : : WRITE
15032 ;          7886  !   : : : : LOOK FOR ERRORS (IF DROP UNIT, LEAVE TESTLOOP)
15033 ;          7887  !   : : : : CHOOSE WHETHER TO WRITE CHECK OR READ
15034 ;          7888  !   : : : : DO THE WRITE CHECK OR READ
15035 ;          7889  !   : : : : LOOK FOR ERRORS (IF DROP UNIT, LEAVE TESTLOOP)
15036 ;          7890  !   : : : : CHANGE BUFFER POINTERS AFTER TRANSFER
15037 ;          7891  !   : : : : END 5 (END OF COUNTING LOOP FOR SECTOR SELECTION)
15038 ;          7892  !   : : : END 4 (END OF TESTLOOP)
15039 ;          7893  !   : : END 3 (END OF COUNTING LOOP FOR UNIT SELECTION)
15040 ;          7894  !   : END 2 (END OF REPEAT LOOP FOR THIS ROUTINE
15041 ;          7895  !   RETURN
15042 ;          7896  !   END 1 (END OF ROUTINE)
15043 ;          7897  !--
15044 ;          7898
15045 ;          7899      local
15046 ;          7900          LUN,
15047 ;          7901          WRDCNT,
15048 ;          7902          SECTOR,
```

```
15053 ;          7903          VALUE,
15054 ;          7904          OLDSEC,
15055 ;          7905          OLDCHN,
15056 ;          7906          PTR,
15057 ;          7907          COMMAND,
15058 ;          7908          DBL_VALUE;
15059 ;          7909
15060 ;          7910      label
15061 ;          7911          LOOP;
15062 ;          7912
15063 ;          7913      PRINTB (SAY1, RTN5D);
15064 ;          7914      !'RAND4'
15065 ;          7915
15066 ;          7916      incr COUNT from 1 to .REPEAT do
15067 ;          7917          begin                                   !* 2 *  START OF REPEAT LOOP FOR THIS ROUTINE
15068 ;          7918          GEN5 ();
15069 ;          7919
15070 ;          7920          incr KOUNT from 1 to (.NUM_DRIVES*4) do
15071 ;          7921              begin                               !* 3 *  START OF COUNTING LOOP FOR UNIT SELECTION
15072 ;          7922              LUN = RNDU ();
15073 ;          7923              L$LUN = .LUN;
15074 ;          7924              WPTR = WBUFF;
15075 ;          7925              RPTR = RBUFF;
15076 ;          7926  LOOP :
15077 ;          7927              begin                               !* 4 *  START OF LOOP THAT COMPLETELY TESTS 1 UNIT
15078 ;          7928
15079 ;          7929  incr KOUNT2 from 1 to 10 do
15080 ;          7930      begin                                       !* 5 *  START OF COUNTING LOOP FOR SECTOR SELECTION
15081 ;          7931      WRDCNT = RNDWC ();                          !EXPECT VALUE BETWEEN 1 AND BUFSIZ
15082 ;          7932      SET_PTRS (.WRDCNT);
15083 ;          7933      SECTOR = RNDSEC (.LUN);                     !EXPECT VALUE BETWEEN LOWEST AND HIGHEST
15084 ;          7934
15085 ;          7935      if (.SECTOR + .WRDCNT/256) gtra HIGHEST then WRDCNT = 256*(HIGHEST - .SECTOR + 1);
15086 ;          7936
15087 ;          7937      VALUE = write (.LUN, .WRDCNT, .WPTR, .SECTOR);
15088 ;          7938
15089 ;          7939      !+
15090 ;          7940      !  SEE HOW SUCCESSFUL THE WRITE WAS:
15091 ;          7941      !-
15092 ;          7942
15093 ;          7943      selectone .VALUE of                         !SEE 'SYSERR' FOR DEFINITION
15094 ;          7944          set                                     !OF ERROR # CONTAINED IN 'VALUE'
15095 ;          7945
15096 ;          7946          [1] :
15097 ;          7947              begin                               !* 5A * RETRY ALLOWED
15098 ;          7948
15099 ;          7949              if RETRY (SIX, write, .LUN, .WRDCNT, .WPTR, .SECTOR) neq 0
15100 ;          7950              then                                !THE RETRY FAILED -- SYSTEM FATAL ERROR
15101 ;          7951                  begin
15102 ;          7952                  WHY_DROPT [.LUN] = CODE_4;
15103 ;          7953                  ERRDF (5401, MSG1, 0);          !**** OPTION 5, RAND4 ERROR 01 ****
15104 ;          7954                  DODU (.LUN);
```

```
15106 ;MLX4
15107 ;
15108 ;
15109 ;          7955                                leave LOOP;             !JUMP JUST BEYOND END OF BLOCK * 4 *
15110 ;          7956                                end;
15111 ;          7957
15112 ;          7958                            end;                        !* 5A *
15113 ;          7959
15114 ;          7960                        [2] :
15115 ;          7961                            begin                       !* 5B * FATAL CONTROLLER ERROR -- NO RETRY ALLOWED
15116 ;          7962                            WHY_DROPT [.LUN] = CODE_5;
15117 ;          7963                            ERRDF (5402, MSG1, 0);      !**** OPTION 5, RAND4 ERROR 02 ****
15118 ;          7964                            DODU (.LUN);
15119 ;          7965                            leave LOOP;                 !JUMP JUST BEYOND END OF BLOCK * 4 *
15120 ;          7966                            end;                        !* 5B *
15121 ;          7967
15122 ;          7968                        [3] :
15123 ;          7969                            begin                       !* 5C * FATAL DRIVE ERROR -- NO RETRY ALLOWED
15124 ;          7970                            ERRDF (5403, MSG1, 0);      !**** OPTION 5, RAND4 ERROR 03 ****
15125 ;          7971                            WHY_DROPT [.LUN] = CODE_6;
15126 ;          7972                            DODU (.LUN);
15127 ;          7973                            leave LOOP;                 !JUMP JUST BEYOND END OF BLOCK * 4 *
15128 ;          7974                            end;                        !* 5C *
15129 ;          7975                        tes;
15130 ;          7976
15131 ;          7977                    COMMAND = CHOOSE ();
15132 ;          7978
15133 ;          7979                    if .COMMAND eql read
15134 ;          7980                    then
15135 ;          7981                        begin
15136 ;          7982                        PTR = .RPTR;
15137 ;          7983                        VALUE = read (.LUN, .WRDCNT, .RPTR, .SECTOR);
15138 ;          7984                        end
15139 ;          7985                    else
15140 ;          7986                        begin
15141 ;          7987                        PTR = .WPTR;
15142 ;          7988                        VALUE = CHECK (.LUN, .WRDCNT, .WPTR, .SECTOR);
15143 ;          7989                        end;
15144 ;          7990
15145 ;          7991                    !+
15146 ;          7992                    ! SEE HOW SUCCESSFUL THE OPERATION WAS:
15147 ;          7993                    !-
15148 ;          7994
15149 ;          7995                    selectone .VALUE of             !SEE 'SYSERR' FOR DEFINITION
15150 ;          7996                        set                         !OF ERROR # CONTAINED IN 'VALUE'
15151 ;          7997
15152 ;          7998                        [0] :
15153 ;          7999
15154 ;          8000                            if .COMMAND eql read
15155 ;          8001                            then
15156 ;          8002                                begin
15157 ;          8003
15158 ;          8004                                if (DBL_VALUE = DOUBLE_CHECK (.WPTR, .RPTR, .WRDCNT)) neq 0
15159 ;          8005                                then
15160 ;          8006                                    begin
```

```
15162 ;MLX4
15163 ;
15164
15165 ;          8007                                    SAYWHO (.LUN);
15166 ;          8008                                    PRINTB (SAY1, MSG5);
15167 ;          8009                                    !'ECC LOGIC FAILED TO DETECT DATA ERROR'
15168 ;          8010                                    PRINTB (FMT12A, ..DBL_VALUE, .DBL_VALUE);
15169 ;          8011                                    !'GOOD DATA: XXXXXX AT LOCATION YYYYYY'
15170 ;          8012                                    DBL_VALUE = .DBL_VALUE + BUFSIZ*2;
15171 ;          8013                                    PRINTB (FMT12B, ..DBL_VALUE, .DBL_VALUE);
15172 ;          8014                                    !'BAD DATA:  PPPPPP AT LOCATION QQQQQQ'
15173 ;          8015                                    WHY_DROPT [.LUN] = CODE_8;
15174 ;          8016                                    ERRDF (5404, MSG1, 0);  !**** OPTION 5, RAND4 ERROR 04 ****
15175 ;          8017                                    DODU (.LUN);
15176 ;          8018                                    leave LOOP;     !JUMP JUST BEYOND END OF BLOCK * 4 *
15177 ;          8019                                    end;
15178 ;          8020
15179 ;          8021                                end;
15180 ;          8022
15181 ;          8023                            [1] :
15182 ;          8024                                begin                   !* 5D * RETRY ALLOWED
15183 ;          8025
15184 ;          8026                                if RETRY (SIX, .COMMAND, .LUN, .WRDCNT, .PTR, .SECTOR) neq 0
15185 ;          8027                                then                    !THE RETRY FAILED -- SYSTEM FATAL ERROR
15186 ;          8028                                    begin
15187 ;          8029                                    WHY_DROPT [.LUN] = CODE_4;
15188 ;          8030                                    ERRDF (5405, MSG1, 0);      !**** OPTION 5, RAND4 ERROR 05 ****
15189 ;          8031                                    DODU (.LUN);
15190 ;          8032                                    leave LOOP;         !JUMP JUST BEYOND END OF BLOCK * 4 *
15191 ;          8033                                    end;
15192 ;          8034
15193 ;          8035                                end;                    !* 5D *
15194 ;          8036
15195 ;          8037                            [2] :
15196 ;          8038                                begin                   !* 5E * FATAL CONTROLLER ERROR -- NO RETRY ALLOWED
15197 ;          8039                                WHY_DROPT [.LUN] = CODE_5;
15198 ;          8040                                ERRDF (5406, MSG1, 0);  !**** OPTION 5, RAND4 ERROR 06 ****
15199 ;          8041                                DODU (.LUN);
15200 ;          8042                                leave LOOP;             !JUMP JUST BEYOND END OF BLOCK * 4 *
15201 ;          8043                                end;                    !* 5E *
15202 ;          8044
15203 ;          8045                            [3] :
15204 ;          8046                                begin                   !* 5F * FATAL DRIVE ERROR -- NO RETRY ALLOWED
15205 ;          8047                                ERRDF (5407, MSG1, 0);  !**** OPTION 5, RAND4 ERROR 07 ****
15206 ;          8048                                WHY_DROPT [.LUN] = CODE_6;
15207 ;          8049                                DODU (.LUN);
15208 ;          8050                                leave LOOP;             !JUMP JUST BEYOND END OF BLOCK * 4 *
15209 ;          8051                                end;                    !* 5F *
15210 ;          8052
15211 ;          8053                            [4] :
15212 ;          8054                                begin                   !* 5G * UNRECOVERABLE DATA ERROR
15213 ;          8055                                ISOLATE ();
15214 ;          8056                                ERRDF (5408, MSG2, 0);  !**** OPTION 5, RAND4 ERROR 08 ****
15215 ;          8057                                WHY_DROPT [.LUN] = CODE_7;
15216 ;          8058                                DODU (.LUN);
```

```
15218 ;MLX4
15219 ;
15220 ;
15221 ;          8059                        leave LOOP;                 !JUMP JUST BEYOND END OF BLOCK * 4 *
15222 ;          8060                        end;                        !* 5G *
15223 ;          8061
15224 ;          8062                    [5] :
15225 ;          8063                        begin                       !* 5H * RECOVERABLE DATA ERROR
15226 ;          8064                        ISOLATE ();
15227 ;          8065
15228 ;          8066                        if .ERROUT then PRINTB (FMT10B, .CHAN);
15229 ;          8067
15230 ;          8068                        !'    BIT 00'
15231 ;          8069                        OLDSEC = .MLEL;
15232 ;          8070                        OLDCHN = .CHAN;
15233 ;          8071
15234 ;          8072                        if RETRY (ONE, .COMMAND, .LUN, .WRDCNT, .PTR, .SECTOR) eql 5
15235 ;          8073                        then
15236 ;          8074
15237 ;          8075                            if ((.MLEL eql .OLDSEC) and (.CHAN eql .OLDCHN))
15238 ;          8076                            then
15239 ;          8077                                begin
15240 ;          8078
15241 ;          8079                                if .ERROUT then ERRHRD (5409, MSG4, 0); !**** OPTION 5, RAND4 ERROR 09 ****
15242 ;          8080
15243 ;          8081                                UP_HARD_COUNT (.LUN, .BOARD);
15244 ;          8082                                end
15245 ;          8083                            else
15246 ;          8084                                begin
15247 ;          8085
15248 ;          8086                                if .ERROUT then ERRSOFT (5410, MSG3, 0);
15249 ;          8087
15250 ;          8088                                                    !**** OPTION 5, RAND4 ERROR 10 ****
15251 ;          8089                                UP_SOFT_COUNT (.LUN, .BOARD);
15252 ;          8090                                end
15253 ;          8091
15254 ;          8092                        else
15255 ;          8093                            begin
15256 ;          8094
15257 ;          8095                            if .ERROUT then ERRSOFT (5411, MSG3, 0);    !**** OPTION 5, RAND4 ERROR 11 ****
15258 ;          8096
15259 ;          8097                            UP_SOFT_COUNT (.LUN, .BOARD);
15260 ;          8098                            end;
15261 ;          8099
15262 ;          8100                        end;                        !* 5H *
15263 ;          8101                    tes;
15264 ;          8102
15265 ;          8103                WPTR = .WPTR + 2;
15266 ;          8104                end;                                !* 5 *  END OF COUNTING LOOP FOR SECTOR SELECTION
15267 ;          8105
15268 ;          8106            !+
15269 ;          8107            ! Test to see if this uut's address space is
15270 ;          8108            ! to be read for soft errors.  This test is
15271 ;          8109            ! is intended for DMT purposes.
15272 ;          8110            !-
```

```
                8111
                8112            if .EFN$21                              !Is the background pattern to be read
                8113            then
                8114                begin
                8115
                8116    !
                8117    ! version czmlbb changed incr to incru
                8118    !
                8119                    incru SECTOR from LOWEST to HIGHEST do
                8120
                8121                        if read (.LUN, 256, RBUFF, .SECTOR) eql 5
                8122                        then
                8123                            begin
                8124                            ISOLATE ();             !Find the failing bank and board no.
                8125
                8126                            if .ERROUT then PRINTB (FMT10B, .CHAN);
                8127
                8128                            ! Print where the error is
                8129                            !
                8130                            ! Save the contents of the ML error location
                8131                            ! register so we can compare it to the new
                8132                            ! contents of this register after the retry.
                8133                            ! This is done to classify the error.
                8134                            !
                8135                            OLDSEC = .MLEL;
                8136                            OLDCHN = .CHAN;
                8137                            !
                8138                            ! Do a classify retry call.  If the same error
                8139                            ! occures then classify it as a hard error.  If
                8140                            ! a different error occured or the error went away
                8141                            ! then classify it as a soft error.
                8142                            !
                8143
                8144                            if RETRY (ONE, .COMMAND, .LUN, 256, .PTR, .OLDSEC) eql 5
                8145                            then
                8146                            !
                8147                            ! The same error occured so see if it is at the same
                8148                            ! sector and channel number, if so then classify
                8149                            ! it as a hard error else classiy it as a soft
                8150                            ! error.
                8151                            !
                8152
                8153                                if ((.MLEL eql .OLDSEC) and (.CHAN eql .OLDCHN))
                8154                                then
                8155                                    begin           !Same error occured 'hard'
                8156
                8157                                    if .ERROUT      !Print error if enabled
                8158                                    then
                8159                                        begin
              P 8160                                        ERRHRD (5412,       !Error number
              P 8161                                            MSG4,   !Error message
                8162                                            0);     !Additional message routine
```

```
15330 ;MLX4
15331 ;                 TESTING RANDOM UNITS, SECTORS, DATA, WORD COUNT
15332
15333 ;          8163                                    end;
15334 ;          8164
15335 ;          8165                                    UP_HARD_COUNT (.LUN, .BOARD);
15336 ;          8166                                    end
15337 ;          8167                                else
15338 ;          8168                                    begin               !Not the same error 'soft'
15339 ;          8169
15340 ;          8170                                    if .ERROUT          !Print error if enabled
15341 ;          8171                                    then
15342 ;          8172                                        begin
15343 ;        P 8173                                        ERRSOFT (5413,      !Error number
15344 ;        P 8174                                            MSG3,   !Error message
15345 ;          8175                                            0);     !Additional message routine
15346 ;          8176                                        end;
15347 ;          8177
15348 ;          8178                                    UP_SOFT_COUNT (.LUN, .BOARD);
15349 ;          8179                                    end
15350 ;          8180
15351 ;          8181                            else                        !Not the same error 'soft'
15352 ;          8182                                begin
15353 ;          8183
15354 ;          8184                                if .ERROUT              !Print error if enabled
15355 ;          8185                                then
15356 ;          8186                                    begin
15357 ;        P 8187                                    ERRSOFT (5414,  !Error number
15358 ;        P 8188                                        MSG3,       !Error message
15359 ;          8189                                        0);         !Additional message routine
15360 ;          8190                                    end;
15361 ;          8191
15362 ;          8192                                UP_SOFT_COUNT (.LUN, .BOARD);
15363 ;          8193                                end;
15364 ;          8194
15365 ;          8195                            end;
15366 ;          8196
15367 ;          8197                    end;
15368 ;          8198
15369 ;          8199                end;                                    !* 4 *  END OF LOOP THAT COMPLETELY TESTS 1 UNIT
15370 ;          8200                end;                                    !* 3 *  END OF COUNTING LOOP FOR UNIT SELECTION
15371 ;          8201
15372 ;          8202            end;                                        !* 2 *  END OF REPEAT LOOP FOR THIS ROUTINE
15373 ;          8203
15374 ;          8204        return;
15375 ;          8205        end;                                            !* 1 *  END OF ROUTINE
15379
15380
15384 100006  004167  105322          RAND4:  JSR     R1,$SAVE5                                       ;
```

.SBTTL RAND4 TESTING RANDOM UNITS, SECTORS, DATA, WORD COUNT

```
15386                                   ;MLX4
15387                                   ;                                                                  27-Mar-1982 19:24:42    TOPS
15388                                   ;       TESTING RANDOM UNITS, SECTORS, DATA, WORD COUNT            27-Mar-1982 19:23:44    PA:<
15389 100012  162706  000026                    SUB     #26,SP
15390 100016  012746  007706                    MOV     #RTN5D,-(SP)
15391 100022  012746  007072                    MOV     #SAV1,-(SP)             ;                                                  7913
15392 100026  012746  000002                    MOV     #2,-(SP)
15393 100032  010600                            MOV     SP,R0
15394 100034  104414                            TRAP    14                      ; SP,*
15395 100036  005066  000032                    CLR     32(SP)
15396 100042  000167  002176                    JMP     41$                     ; COUNT                                            7916
15397 100046  004767  141546            1$:     JSR     PC,GEN5
15398 100052  016766  134400  000016            MOV     NUM.DRIVES,16(SP)       ;                                                  7918
15399 100060  006366  000016                    ASL     16(SP)                  ;                                                  7920
15400 100064  006366  000016                    ASL     16(SP)
15401 100070  005066  000014                    CLR     14(SP)
15402 100074  000563                            BR      8$                      ; KOUNT
15403 100076  004767  170234            2$:     JSR     PC,RNDU
15404 100102  010002                            MOV     R0,R2                   ;                                                  7922
15405 100104  010267  101764                    MOV     R2,L$LUN                ; *,LUN
15406 100110  012767  012670  132552            MOV     #WBUFF,WPTR             ; LUN,*                                            7923
15407 100116  012767  022670  132546            MOV     #RBUFF,RPTR             ;                                                  7924
15408 100124  010266  000012                    MOV     R2,12(SP)               ;                                                  7925
15409 100130  006366  000012                    ASL     12(SP)                  ; LUN,*                                            7935
15410 100134  012766  034500  000010            MOV     #TOP.SECT,10(SP)
15411 100142  066666  000012  000010            ADD     12(SP),10(SP)
15412 100150  012766  000001  000006            MOV     #1,6(SP)
15413 100156  004767  170022            3$:     JSR     PC,RNDWC                ; *,KOUNT2                                         7929
15414 100162  010003                            MOV     R0,R3                   ;                                                  7931
15415 100164  010346                            MOV     R3,-(SP)                ; *,WRDCNT
15416 100166  004767  146740                    JSR     PC,SET.PTRS             ; WRDCNT,*                                         7932
15417 100172  010216                            MOV     R2,(SP)
15418 100174  004767  170040                    JSR     PC,RNDSEC               ; LUN,*                                            7933
15419 100200  010001                            MOV     R0,R1
15420 100202  010346                            MOV     R3,-(SP)                ; *,SECTOR
15421 100204  012746  000400                    MOV     #400,-(SP)              ; WRDCNT,*                                         7935
15422 100210  004767  104754                    JSR     PC,BL$DIV
15423 100214  022626                            CMP     (SP)+,(SP)+
15424 100216  060100                            ADD     R1,R0
15425 100220  020076  000012                    CMP     R0,@12(SP)              ; SECTOR,*
15426 100224  101410                            BLOS    4$
15427 100226  017600  000012                    MOV     @12(SP),R0
15428 100232  160100                            SUB     R1,R0                   ; SECTOR,*
15429 100234  000300                            SWAB    R0
15430 100236  105000                            CLRB    R0
15431 100240  010003                            MOV     R0,R3                   ; *,WRDCNT
15432 100242  062703  000400                    ADD     #400,R3                 ; *,WRDCNT
15433 100246  010216                    4$:     MOV     R2,(SP)                 ; LUN,*                                            7937
15434 100250  010346                            MOV     R3,-(SP)                ; WRDCNT,*
15435 100252  016746  132412                    MOV     WPTR,-(SP)
15436 100256  010146                            MOV     R1,-(SP)                ; SECTOR,*
15437 100260  004767  145140                    JSR     PC,WRITE
15438 100264  010004                            MOV     R0,R4
15439 100266  020427  000001                    CMP     R4,#1                   ; *,VALUE
15440 100272  001031                            BNE     5$                      ; VALUE,*                                          7943
```

```
                                        ;MLX4
                                        ;                                                    27-Mar-1982 19:24:42    TOPS
                                                TESTING RANDOM UNITS, SECTORS, DATA, WORD COUNT 27-Mar-1982 19:23:44    PA:<
15445 100274 012746 000006              MOV     #6,-(SP)                                    ;                           7949
15446 100300 012746 045424              MOV     #WRITE,-(SP)
15447 100304 010246                     MOV     R2,-(SP)                                    ; LUN,*
15448 100306 010346                     MOV     R3,-(SP)                                    ; WRDCNT,*
15449 100310 016746 132354              MOV     WPTR,-(SP)
15450 100314 010146                     MOV     R1,-(SP)                                    ; SECTOR,*
15451 100316 004767 145702              JSR     PC,RETRY
15452 100322 062706 000014              ADD     #14,SP
15453 100326 005700                     TST     R0
15454 100330 001446                     BEQ     9$
15455 100332 112762 000004 034446       MOVB    #4,WHY.DROPT(R2)                            ; *,*(LUN)                  7952
15456 100340 104455                     TRAP    55                                          ;                           7953
15457 100342 012431                     .WORD   12431
15458 100344 011070                     .WORD   MSG1
15459 100346 000000                     .WORD   0
15460 100350 010200                     MOV     R2,R0                                       ; LUN,*                     7954
15461 100352 104451                     TRAP    51
15462 100354 000431                     BR      7$                                          ;                           7955
15463 100356 020427 000002      5$:     CMP     R4,#2                                       ; VALUE,*                   7943
15464 100362 001012                     BNE     6$
15465 100364 112762 000005 034446       MOVB    #5,WHY.DROPT(R2)                            ; *,*(LUN)                  7962
15466 100372 104455                     TRAP    55                                          ;                           7963
15467 100374 012432                     .WORD   12432
15468 100376 011070                     .WORD   MSG1
15469 100400 000000                     .WORD   0
15470 100402 010200                     MOV     R2,R0                                       ; LUN,*                     7964
15471 100404 104451                     TRAP    51
15472 100406 000414                     BR      7$                                          ;                           7965
15473 100410 020427 000003      6$:     CMP     R4,#3                                       ; VALUE,*                   7943
15474 100414 001014                     BNE     9$
15475 100416 104455                     TRAP    55                                          ;                           7970
15476 100420 012433                     .WORD   12433
15477 100422 011070                     .WORD   MSG1
15478 100424 000000                     .WORD   0
15479 100426 112762 000006 034446       MOVB    #6,WHY.DROPT(R2)                            ; *,*(LUN)                  7971
15480 100434 010200                     MOV     R2,R0                                       ; LUN,*                     7972
15481 100436 104451                     TRAP    51
15482 100440 062706 000010      7$:     ADD     #10,SP                                      ;                           7973
15483 100444 000535             8$:     BR      14$
15484 100446 004767 145514      9$:     JSR     PC,CHOOSE                                   ;                           7977
15485 100452 010066 000040              MOV     R0,40(SP)                                   ; *,COMMAND
15486 100456 005005                     CLR     R5                                          ;                           7979
15487 100460 020027 045612              CMP     R0,#READ                                    ; COMMAND,*
15488 100464 001014                     BNE     10$
15489 100466 005205                     INC     R5
15490 100470 016766 132176 000036       MOV     RPTR,36(SP)                                 ; *,PTR                     7982
15491 100476 010246                     MOV     R2,-(SP)                                    ; LUN,*                     7983
15492 100500 010346                     MOV     R3,-(SP)                                    ; WRDCNT,*
15493 100502 016746 132164              MOV     RPTR,-(SP)
15494 100506 010146                     MOV     R1,-(SP)                                    ; SECTOR,*
15495 100510 004767 145076              JSR     PC,READ
15496 100514 000412                     BR      11$
```

SEQ 0334

```
15498                                          ;MLX4
15499                                          ;
15500                                                  TESTING RANDOM UNITS, SECTORS, DATA, WORD COUNT 27-Mar-1982 19:24:42    TOPS
15501 100516 016766 132146 000036 10$:   MOV   WPTR,36(SP)                                            27-Mar-1982 19:23:44    PA:<
15502 100524 010246                      MOV   R2,-(SP)          ; *,PTR                                                        7987
15503 100526 010346                      MOV   R3,-(SP)          ; LUN,*                                                        7988
15504 100530 016746 132134               MOV   WPTR,-(SP)        ; WRDCNT,*
15505 100534 010176                      MOV   R1,-(SP)
15506 100536 004767 145236               JSR   PC,CHECK          ; SECTOR,*
15507 100542 010004              11$:    MOV   R0,R4
15508 100544 001076                      BNE   15$               ; *,VALUE
15509 100546 006005                      ROR   R5                ;                                                              7995
15510 100550 103402                      BLO   13$               ;                                                              8000
15511 100552 000167 000750       12$:    JMP   28$
15512 100556 016746 132106       13$:    MOV   WPTR,-(SP)
15513 100562 016746 132104               MOV   RPTR,-(SP)        ;                                                              8004
15514 100566 010346                      MOV   R3,-(SP)          ; WRDCNT,*
15515 100570 004767 143532               JSR   PC,DOUBLE.CHECK
15516 100574 062706 000006               ADD   #6,SP
15517 100600 010066 000042               MOV   R0,42(SP)         ; *,DBL.VALUE
15518 100604 001762                      BEQ   12$
15519 100606 010246                      MOV   R2,-(SP)          ; LUN,*                                                        8007
15520 100610 004767 134710               JSR   PC,SAYWHO
15521 100614 012716 011216               MOV   #MSG5,(SP)        ;                                                              8008
15522 100620 012746 007072               MOV   #SAY1,-(SP)
15523 100624 012746 000002               MOV   #2,-(SP)
15524 100630 010600                      MOV   SP,R0             ; SP,*
15525 100632 104414                      TRAP  14
15526 100634 016616 000050               MOV   50(SP),(SP)       ; DBL.VALUE,*                                                  8010
15527 100640 017646 000050               MOV   @50(SP),-(SP)     ; DBL.VALUE,*
15528 100644 012746 006710               MOV   #FMT12A,-(SP)
15529 100650 012746 000003               MOV   #3,-(SP)
15530 100654 010600                      MOV   SP,R0             ; SP,*
15531 100656 104414                      TRAP  14
15532 100660 062766 010000 000056        ADD   #10000,56(SP)     ; *,DBL.VALUE                                                  8012
15533 100666 016616 000056               MOV   56(SP),(SP)       ; DBL.VALUE,*                                                  8013
15534 100672 017646 000056               MOV   @56(SP),-(SP)     ; DBL.VALUE,*
15535 100676 012746 006756               MOV   #FMT12B,-(SP)
15536 100702 012746 000003               MOV   #3,-(SP)
15537 100706 010000                      MOV   SP,R0             ; SP,*
15538 100710 104414                      TRAP  14
15539 100712 112762 000010 034446        MOVB  #10,WHY.DROPT(R2) ; *,*(LUN)                                                     8015
15540 100720 104455                      TRAP  55                ;                                                              8016
15541 100722 012434                      .WORD 12434
15542 100724 011070                      .WORD MSG1
15543 100726 000000                      .WORD 0
15544 100730 010200                      MOV   R2,R0             ; LUN,*                                                        8017
15545 100732 104451                      TRAP  51
15546 100734 062706 000042               ADD   #42,SP            ;                                                              8018
15547 100740 000506              14$:    BR    20$
15548 100742 020427 000001       15$:    CMP   R4,#1             ; VALUE,*                                                      7995
15549 100746 001031                      BNE   16$
15550 100750 012746 000006               MOV   #6,-(SP)          ;                                                              8026
15551 100754 016646 000052               MOV   52(SP),-(SP)      ; COMMAND,*
15552 100760 010246                      MOV   R2,-(SP)          ; LUN,*
```

```
                                        ;MLX4
                                        ;                TESTING RANDOM UNITS, SECTORS, DATA, WORD COUNT 27-Mar-1982 19:24:42  TOPS
                                                                                                         27-Mar-1982 19:23:44  PA:<
100762 010346                                   MOV     R3,-(SP)                ; WRDCNT,*
100764 016646 000056                            MOV     56(SP),-(SP)            ; PTR,*
100770 010146                                   MOV     R1,-(SP)                ; SECTOR,*
100772 004767 145226                            JSR     PC,RETRY
100776 062706 000014                            ADD     #14,SP
101002 005700                                   TST     R0
101004 001662                                   BEQ     12$
101006 112762 000004 034446                     MOVB    #4,WHY.DROPT(R2)        ; *,*(LUN)                                  8029
101014 104455                                   TRAP    55                      ;                                           8030
101016 012435                                   .WORD   12435
101020 011070                                   .WORD   MSG1
101022 000000                                   .WORD   0
101024 010200                                   MOV     R2,R0
101026 104451                                   TRAP    51                      ; LUN,*                                     8031
101030 000450                                   BR      19$                     ;                                           8032
101032 020427 000002            16$:            CMP     R4,#2                   ; VALUE,*                                   7995
101036 001012                                   BNE     17$
101040 112762 000005 034446                     MOVB    #5,WHY.DROPT(R2)        ; *,*(LUN)                                  8039
101046 104455                                   TRAP    55                      ;                                           8040
101050 012436                                   .WORD   12436
101052 011070                                   .WORD   MSG1
101054 000000                                   .WORD   0
101056 010200                                   MOV     R2,R0
101060 104451                                   TRAP    51                      ; LUN,*                                     8041
101062 000433                                   BR      19$                     ;                                           8042
101064 020427 000003            17$:            CMP     R4,#3                   ; VALUE,*                                   7995
101070 001012                                   BNE     18$
101072 104455                                   TRAP    55                      ;                                           8047
101074 012437                                   .WORD   12437
101076 011070                                   .WORD   MSG1
101100 000000                                   .WORD   0
101102 112762 000006 034446                     MOVB    #6,WHY.DROPT(R2)        ; *,*(LUN)                                  8048
101110 010200                                   MOV     R2,R0                   ; LUN,*                                     8049
101112 104451                                   TRAP    51                      ;                                           8050
101114 000416                                   BR      19$                     ; VALUE,*                                   7995
101116 020427 000004            18$:            CMP     R4,#4                   ;                                           8055
101122 001017                                   BNE     21$                     ;                                           8056
101124 004767 136202                            JSR     PC,ISOLATE
101130 104455                                   TRAP    55
101132 012440                                   .WORD   12440
101134 011120                                   .WORD   MSG2
101136 000000                                   .WORD   0
101140 112762 000007 034446                     MOVB    #7,WHY.DROPT(R2)        ; *,*(LUN)                                  8057
101146 010200                                   MOV     R2,R0                   ; LUN,*                                     8058
101150 104451                                   TRAP    51
101152 062706 000020            19$:            ADD     #20,SP                  ;                                           8059
101156 000167 001042            20$:            JMP     40$
101162 020427 000005            21$:            CMP     R4,#5                   ; VALUE,*                                   7995
101166 001157                                   BNE     28$
101170 004767 136136                            JSR     PC,ISOLATE              ;                                           8064
101174 032767 000001 101056                     BIT     #1,ERROUT               ;                                           8066
101202 001423                                   BEQ     22$
```

```
                                        ;MLX4
                                        ;                                                                    27-Mar-1982 19:24:42    TOPS
                                                 TESTING RANDOM UNITS, SECTORS, DATA, WORD COUNT             27-Mar-1982 19:23:44    PA:<
 15610 101204  017705  133200                    MOV     @ML.REG+42,R5
 15611 101210  006205                            ASR     R5
 15612 101212  006205                            ASR     R5
 15613 101214  006205                            ASR     R5
 15614 101216  006205                            ASR     R5
 15615 101220  006205                            ASR     R5
 15616 101222  006205                            ASR     R5
 15617 101224  042705  177700                    BIC     #177700,R5
 15618 101230  010546                            MOV     R5,-(SP)
 15619 101232  012746  006640                    MOV     #FMT10B,-(SP)
 15620 101236  012746  000002                    MOV     #2,-(SP)
 15621 101242  010600                            MOV     SP,R0
 15622 101244  104414                            TRAP    14                              ; SP,*
 15623 101246  062706  000006                    ADD     #6,SP
 15624 101252  017766  133134  000044  22$:      MOV     @ML.REG+44,44(SP)               ; *,OLDSEC                              8069
 15625 101260  017705  133124                    MOV     @ML.REG+42,R5                   ;                                       8070
 15626 101264  006205                            ASR     R5
 15627 101266  006205                            ASR     R5
 15630 101270  006205                            ASR     R5
 15631 101272  006205                            ASR     R5
 15632 101274  006205                            ASR     R5
 15633 101276  006205                            ASR     R5
 15634 101300  042705  177700                    BIC     #177700,R5
 15635 101304  010566  000040                    MOV     R5,40(SP)
 15636 101310  012746  000001                    MOV     #1,-(SP)                        ; *,OLDCHN
 15637 101314  016646  000052                    MOV     52(SP),-(SP)                    ;                                       8072
 15638 101320  010246                            MOV     R2,-(SP)                        ; COMMAND,*
 15639 101322  010346                            MOV     R3,-(SP)                        ; LUN,*
 15640 101324  016646  000056                    MOV     56(SP),-(SP)                    ; WRDCNT,*
 15641 101330  010146                            MOV     R1,-(SP)                        ; PTR,*
 15642 101332  004767  144666                    JSR     PC,RETRY                        ; SECTOR,*
 15643 101336  062706  000014                    ADD     #14,SP
 15644 101342  020027  000005                    CMP     R0,#5
 15645 101346  001051                            BNE     25$
 15646 101350  027766  133036  000044            CMP     @ML.REG+44,44(SP)               ; *,OLDSEC                              8075
 15647 101356  001034                            BNE     24$
 15650 101360  016600  000040                    MOV     40(SP),R0                       ; OLDCHN,*
 15651 101364  017705  133020                    MOV     @ML.REG+42,R5
 15652 101370  006205                            ASR     R5
 15653 101372  006205                            ASR     R5
 15654 101374  006205                            ASR     R5
 15655 101376  006205                            ASR     R5
 15656 101400  006205                            ASR     R5
 15657 101402  006205                            ASR     R5
 15660 101404  042705  177700                    BIC     #177700,R5
 15661 101410  020500                            CMP     R5,R0
 15662 101412  001016                            BNE     24$
 15663 101414  032767  000001  100636            BIT     #1,ERROUT                       ;                                       8079
 15664 101422  001404                            BEQ     23$
 15665 101424  104456                            TRAP    56
 15666 101426  012441                            .WORD   12441
 15667 101430  011202                            .WORD   MSG4
```

```
                                        ;M' X4
                                        ;                                                    27-Mar-1982 19:24:42    TOPS
                                        ;       TESTING RANDOM UNITS, SECTORS, DATA, WORD COUNT 27-Mar-1982 19:23:44    PA:<
000000                                          .WORD   0
010246                          23$:            MOV     R2,-(SP)                    ; LUN,*                             8081
016746  132700                                  MOV     BOARD,-(SP)
004767  135754                                  JSR     PC,UP.HARD.COUNT
000426                                          BR      27$
032767  000001  100602          24$:            BIT     #1,ERROUT                   ;                                   8075
001415                                          BEQ     26$                         ;                                   8086
104457                                          TRAP    57
012442                                          .WORD   12442
011166                                          .WORD   MSG3
000000                                          .WORD   0
000410                                          BR      26$
032767  000001  100560          25$:            BIT     #1,ERROUT                   ;                                   8089
001404                                          BEQ     26$                         ;                                   8095
104457                                          TRAP    57
012443                                          .WORD   12443
011166                                          .WORD   MSG3
000000                                          .WORD   0
010246                          26$:            MOV     R2,-(SP)                    ; LUN,*                             8097
016746  132622                                  MOV     BOARD,-(SP)
004767  136360                                  JSR     PC,UP.SOFT.COUNT
022626                          27$:            CMP     (SP)+,(SP)+
062767  000002  131134          28$:            ADD     #2,WPTR                     ;                                   8063
062706  000020                                  ADD     #20,SP                      ;                                   8103
005266  000006                                  INC     6(SP)                       ;                                   7930
026627  000006  000012                          CMP     6(SP),#12                   ; KOUNT2                            7929
003002                                          BGT     29$                         ; KOUNT2,*
000167  176376                                  JMP     3$
032767  000001  100474          29$:            BIT     #1,EFN$21                   ;                                   8112
001002                                          BNE     30$
000167  000430                                  JMP     40$
017666  000010  000010          30$:            MOV     @10(SP),10(SP)              ;                                   8119
016600  000012                                  MOV     12(SP),R0
016005  034460                                  MOV     LOW.SECT(R0),R5             ; *,SECTOR
000577                                          BR      39$
010246                          31$:            MOV     R2,-(SP)                    ; LUN,*                             8121
012746  000400                                  MOV     #400,-(SP)
012746  022670                                  MOV     #RBUFF,-(SP)
010546                                          MOV     R5,-(SP)                    ; SECTOR,*
004767  143756                                  JSR     PC,READ
062706  000010                                  ADD     #10,SP
020027  000005                                  CMP     R0,#5
001161                                          BNE     38$
004767  135460                                  JSR     PC,ISOLATE
032767  000001  100400                          BIT     #1,ERROUT                   ;                                   8124
001423                                          BEQ     32$                         ;                                   8126
017700  132522                                  MOV     @ML.REG+42,R0
006200                                          ASR     R0
006200                                          ASR     R0
006200                                          ASR     R0
006200                                          ASR     R0
006200                                          ASR     R0
```

```
                                        ;MLX4
                                        ;                                                    27-Mar-1982 19:24:42    TOPS
                                        ;       TESTING RANDOM UNITS, SECTORS, DATA, WORD COUNT 27-Mar-1982 19:23:44    PA:<
                                
15725 101700 006200                             ASR     R0
15726 101702 042700 177700                      BIC     #177700,R0
15727 101706 010046                             MOV     R0,-(SP)
15728 101710 012746 006640                      MOV     #FMT10B,-(SP)
15729 101714 012746 000002                      MOV     #2,-(SP)
15730 101720 010600                             MOV     SP,R0                   ; SP,*
15731 101722 104411                             TRAP    14
15732 101724 062706 000006                      ADD     #6,SP
15733 101730 017766 132456 000024  32$:         MOV     @ML.REG+44,24(SP)       ; *,OLDSEC                              8135
15734 101736 017700 132446                      MOV     @ML.REG+42,R0           ;                                       8136
15735 101742 006200                             ASR     R0
15736 101744 006200                             ASR     R0
15737 101746 006200                             ASR     R0
15738 101750 006200                             ASR     R0
15739 101752 006200                             ASR     R0
15740 101754 006200                             ASR     R0
15741 101756 042700 177700                      BIC     #177700,R0
15742 101762 010066 000020                      MOV     R0,20(SP)               ; *,OLDCHN
15743 101766 012746 000001                      MOV     #1,-(SP)                ;                                       8144
15744 101772 016646 000032                      MOV     32(SP),-(SP)            ; COMMAND,*
15745 101776 010246                             MOV     R2,-(SP)                ; LUN,*
15746 102000 012746 000400                      MOV     #400,-(SP)
15747 102004 016646 000036                      MOV     36(SP),-(SP)            ; PTR,*
15748 102010 016646 000036                      MOV     36(SP),-(SP)            ; OLDSEC,*
15749 102014 004767 144204                      JSR     PC,RETRY
15750 102020 062706 000014                      ADD     #14,SP
15751 102024 020027 000005                      CMP     R0,#5
15752 102030 001051                             BNE     35$
15753 102032 027766 132354 000024               CMP     @ML.REG+44,24(SP)       ; *,OLDSEC                              8153
15754 102040 001034                             BNE     34$
15755 102042 016646 000020                      MOV     20(SP),-(SP)            ; OLDCHN,*
15756 102046 017700 132336                      MOV     @ML.REG+42,R0
15757 102052 006200                             ASR     R0
15758 102054 006200                             ASR     R0
15759 102056 006200                             ASR     R0
15760 102060 006200                             ASR     R0
15761 102062 006200                             ASR     R0
15762 102064 006200                             ASR     R0
15763 102066 042700 177700                      BIC     #177700,R0
15764 102072 020026                             CMP     R0,(SP)+
15765 102074 001016                             BNE     34$
15766 102076 032767 000001 100154               BIT     #1,ERROUT               ;                                       8157
15767 102104 001404                             BEQ     33$
15768 102106 104456                             TRAP    56                      ;                                       8162
15769 102110 012444                             .WORD   12444
15770 102112 011202                             .WORD   MSG4
15771 102114 000000                             .WORD   0
15772 102116 010246                     33$:    MOV     R2,-(SP)                ; LUN,*                                 8165
15773 102120 016746 132216                      MOV     BOARD,-(SP)
15774 102124 004767 135272                      JSR     PC,UP.HARD.COUNT
15775 102130 000426                             BR      37$
15776 102132 032767 000001 100120  34$:         BIT     #1,ERROUT               ;                                       8153
                                                                                ;                                       8170
```

```
15778                                          ;MLX4
15779                                          :                    TESTING RANDOM UNITS, SECTORS, DATA, WORD COUNT 27-Mar-1982 19:24:42   TOPS
15780                                          :                                                                    27-Mar-1982 19:23:44   PA:<
15781 102140 001415                                    BEQ     36$
15782 102142 104457                                    TRAP    57
15783 102144 012445                                    .WORD   12445                ;                                                       8175
15784 102146 011166                                    .WORD   MSG3
15785 102150 000000                                    .WORD   0
15786 102152 000410                                    BR      36$
15787 102154 032767 000001 100076  35$:        BIT     #1,ERROUT                    ;                                                       8178
15788 102162 001404                                    BEQ     36$                  ;                                                       8184
15789 102164 104457                                    TRAP    57
15790 102166 012446                                    .WORD   12446                ;                                                       8189
15791 102170 011166                                    .WORD   MSG3
15792 102172 000000                                    .WORD   0
15793 102174 010246                            36$:    MOV     R2,-(SP)
15794 102176 016746 132140                             MOV     BOARD,-(SP)           ; LUN,*                                           8192
15795 102202 004767 135676                             JSR     PC,UP.SOFT.COUNT
15796 102206 022626                            37$:    CMP     (SP)+,(SP)+
15797 102210 005205                            38$:    INC     R5                    ;                                                 C123
15798 102212 020566 000010                     39$:    CMP     R5,10(SP)             ; SECTOR                                          8119
15799 102216 101002                                    BHI     40$                   ; SECTOR,*
15800 102220 000167 177370                             JMP     31$
15801 102224 005266 000014                     40$:    INC     14(SP)                ; KOUNT
15802 102230 026666 000014 000016                      CMP     14(SP),16(SP)         ; KOUNT,*                                         7920
15803 102236 003002                                    BGT     41$
15804 102240 000167 175632                             JMP     2$
15805 102244 005266 000032                     41$:    INC     32(SP)                ; COUNT
15806 102250 026666 000032 000052                      CMP     32(SP),52(SP)         ; COUNT,REPEAT                                    7916
15807 102256 003002                                    BGT     42$
15808 102260 000167 175562                             JMP     1$
15809 102264 062706 000034                     42$:    ADD     #34,SP                ;
15810 102270 000207                                    RTS     PC                                                                      7852
15811
15812
15813                                          ; Routine Size:  602 words
15818                                          ; Maximum stack depth per invocation:  37 words
15819
```

```
15821 ;MLX4
15822 ;                 DEFINITION OF OPTION 5
15823 ;
15824 ;          8206  %sbttl 'DEFINITION OF OPTION 5'
15825 ;          8207  routine OPT5 : novalue =
15826 ;          8208      begin                                          !* 1 *
15827 ;          8209
15828 ;          8210  !++
15829 ;          8211  !  ROUTINE:     OPT5
15830 ;          8212  !
15831 ;          8213  !  PURPOSE:     TO EXERCISE THE ML11 SYSTEMS UNDER TEST IN A RANDOM
15832 ;          8214  !               MANNER, SO AS TO SIMULATE THE FLEXIBILITY OF TESTING
15833 ;          8215  !               THAT WOULD BE DONE BY AN OPERATING SYSTEM.
15834 ;          8216  !
15835 ;          8217  !               THERE ARE 4 RANDOM TESTS WHICH ARE CALLED BY 'OPT5'
15836 ;          8218  !               TO ACCOMPLISH ALL TESTING.  IT IS THE RESPONSIBILITY
15837 ;          8219  !               OF THIS ROUTINE TO DECIDE HOW MANY TIMES THOSE 4
15838 ;          8220  !               RANDOM TESTS WILL BE EXECUTED.  REFER TO 'RAND1' TO
15839 ;          8221  !               'RAND4' ABOVE FOR MORE INFORMATION.
15840 ;          8222  !--
15841 ;          8223
15842 ;          8224      local
15843 ;          8225          REPEAT;
15844 ;          8226
15845 ;          8227      PRINTB (SAY2, WRD34, RTN5);
15846 ;          8228      !'RUNNING OPT5'
15847 ;          8229      REPEAT = 9 - .NUM_DRIVES;                      !THE MORE UNITS THERE ARE ON THE SYSTEM, THE
15848 ;          8230      !FEWER TIMES EACH RANDOM TEST WILL BE RUN.
15849 ;          8231
15850 ;          8232      incr LUN from 0 to (.LSUNIT - 1) do            !SEE IF THERE ARE ANY UNITS
15851 ;          8233          begin                                      !* 2 *                    !WHICH HAVE 64K CHIPS (ML11-B)
15852 ;          8234
15853 ;          8235          if .DRIVE_STATUS [.LUN] eql ACTIVE
15854 ;          8236          then
15855 ;          8237              begin                                  !* 3 *
15856 ;          8238              UNIT = .DRIVE;
15857 ;          8239
15858 ;          8240              if .MLDT
15859 ;          8241              then
15860 ;          8242                  begin                              !* 4 *
15861 ;          8243                  REPEAT = 1;                        !THERE IS AT LEAST 1 UNIT WHICH IS AN ML11-B, SO
15862 ;          8244                  !LOWER THE REPEAT TIME FOR ALL TESTABLE UNITS.
15863 ;          8245                  exitloop;
15864 ;          8246                  end;                               !* 4 *
15865 ;          8247
15866 ;          8248              end;                                   !* 3 *
15867 ;          8249
15868 ;          8250          end;                                       !* 2 *
15869 ;          8251
15870 ;          8252  !+
15871 ;          8253  ! AT THIS POINT, 'REPEAT' IS BASED ON THE SYSTEM CONFIGURATION.
15872 ;          8254  ! NOW WEIGHT THE LOOP COUNTS OF THE 4 RANDOM TESTS SO THAT RAND1
15873 ;          8255  ! AND RAND2 ARE MUCH QUICKER THAN RAND3 AND RAND4.  RAND3 IS LONGEST.
15874 ;          8256  !-
15875 ;          8257      RAND1 (.REPEAT);                               !TEST USING RANDOM DATA
```

```
15877 ;MLX4
15878 ;                 DEFINITION OF OPTION 5                    27-Mar-1982 19:24:42    TOPS-20 Bliss-16 V2(212)
15879                                                             27-Mar-1982 19:23:44    PA:<NEALE>MLX4.BLI.5 (48)
15880 ;          8258       RAND2 (.REPEAT);                      !TEST USING RANDOM DATA, WORD COUNTS
15881 ;          8259       RAND3 (.REPEAT);                      !TEST USING RANDOM SECTORS, DATA, WORD COUNTS
15882 ;          8260       RAND4 ((.REPEAT*16));                 !TEST USING RANDOM UNITS, SECTORS, DATA, WORD COUNTS
15883 ;          8261       return;
15884 ;          8262       end;                                  !* 1 *
15888
15889                                          .SBTTL  OPT5 DEFINITION OF OPTION 5
15893 102272  004167  103016          OPT5:   JSR     R1,$SAVE4
15894 102276  012746  007656                  MOV     #RTN5,-(SP)             ;                                       8207
15895 102302  012746  007344                  MOV     #WRD34,-(SP)            ;                                       8227
15896 102306  012746  007100                  MOV     #SAY2,-(SP)
15897 102312  012746  000003                  MOV     #3,-(SP)
15898 102316  010600                          MOV     SP,R0
15899 102320  104414                          TRAP    14                      ; SP,*
15900 102322  012703  000011                  MOV     #11,R3
15901 102326  166703  132124                  SUB     NUM.DRIVES,R3           ; *,REPEAT                              8229
15902 102332  016704  077454                  MOV     L$UNIT,R4               ;                                       8232
15903 102336  005001                          CLR     R1                      ; LUN
15904 102340  000450                          BR      3$
15905 102342  010102          1$:             MOV     R1,R2                   ; LUN,*                                 8235
15906 102344  006202                          ASR     R2
15907 102346  006202                          ASR     R2
15908 102350  006202                          ASR     R2
15909 102352  062702  034442                  ADD     #DRIVE.STATUS,R2
15910 102356  010246                          MOV     R2,-(SP)
15911 102360  010146                          MOV     R1,-(SP)                ; LUN,*
15912 102362  042716  177770                  BIC     #177770,(SP)
15913 102366  012746  000001                  MOV     #1,-(SP)
15914 102372  005046                          CLR     -(SP)
15915 102374  004767  101756                  JSR     PC,BL$GT2
15916 102400  062706  000010                  ADD     #10,SP
15917 102404  005300                          DEC     R0
15918 102406  001374                          BNE     2$
15919 102410  010102                          MOV     R1,R2
15920 102412  006302                          ASL     R2                      ; LUN,*                                 8238
15921 102414  016202  034422                  MOV     PTABLE.ADDR(R2),R2
15922 102420  016200  000006                  MOV     6(R2),R0
15923 102424  042700  177770                  BIC     #177770,R0
15924 102430  142777  000007  131720          BICB    #7,@ML.REG+10
15925 102436  150077  131714                  BISB    R0,@ML.REG+10
15926 102442  032777  000001  131724          BIT     #1,@ML.REG+26           ;                                       8240
15927 102450  001403                          BEQ     2$
15928 102452  012703  000001                  MOV     #1,R3                   ; *,REPEAT                              8243
15929 102456  000403                          BR      4$                      ;                                       8245
15930 102460  005201          2$:             INC     R1                      ; LUN                                   8232
15931 102462  020104          3$:             CMP     R1,R4                   ; LUN,*
```

SEQ 0342

```
15933                                   ;MLX4
15934                                   ;                       DEFINITION OF OPTION 5                                  27-Mar-1982 19:24:42    TOPS
15935                                                                                                                   27-Mar-1982 19:23:44    PA:<
15936 102464  002726                            BLT     1$
15937 102466  010316                    4$:     MOV     R3,(SP)
15938 102470  004767  166044                    JSR     PC,RAND1                ; REPEAT,*                              8257
15939 102474  010316                            MOV     R3,(SP)
15940 102476  004767  170422                    JSR     PC,RAND2                ; REPEAT,*                              8258
15941 102502  010316                            MOV     R3,(SP)
15942 102504  004767  172734                    JSR     PC,RAND3                ; REPEAT,*                              8259
15943 102510  010300                            MOV     R3,R0
15944 102512  006300                            ASL     R0                      ; REPEAT,*                              8260
15945 102514  006300                            ASL     R0
15946 102516  006300                            ASL     R0
15947 102520  006300                            ASL     R0
15948 102522  010016                            MOV     R0,(SP)
15949 102524  004767  175256                    JSR     PC,RAND4
15950 102530  062706  000010                    ADD     #10,SP
15951 102534  000207                            RTS     PC                      ;                                       8207
15952
15953                                   ; Routine Size:  82 words
15954                                   ; Maximum stack depth per invocation:  13 words
15959
15960
```

27-Mar-1982 19:24:42 TOPS-20 Bliss-16 V2(212)
27-Mar-1982 19:23:44 PA:<NEALE>MLX4.BLI.5 (49)

```
15962 ;MLX4
15963 ;                 THE OPTION SCHEDULER
15964 ;
15965 ;          8263  %sbttl 'THE OPTION SCHEDULER'
15966 ;          8264  BGNTST;
15967 ;          8265
15968 ;          8266  !++
15969 ;          8267  !  ROUTINE:     T1
15970 ;          8268  !
15971 ;          8269  !  PURPOSE:     THIS IS THE ONE MAIN TEST OF THE EXERCISER.  IT LOOKS
15972 ;          8270  !               AT THE AVAILABILITY OF TEST OPTIONS AND MAKES CALLS TO
15973 ;          8271  !               THE OPTIONS WHEN APPROPRIATE.
15974 ;          8272  !
15975 ;          8273  !               THE QUICK VERIFY PASS INCLUDES THE FOLLOWING ROUTINES:
15976 ;          8274  !
15977 ;          8275  !               (1)  INTEGRITY
15978 ;          8276  !               (2)  OPT1
15979 ;          8277  !               (3)  OPT2
15980 ;          8278  !               (4)  OPT3
15981 ;          8279  !
15982 ;          8280  !               SUBSEQUENT PASSES INCLUDE:
15983 ;          8281  !
15984 ;          8282  !               (1)  OPT1
15985 ;          8283  !               (2)  OPT2
15986 ;          8284  !               (3)  OPT3
15987 ;          8285  !               (4)  OPT4
15988 ;          8286  !               (5)  OPT5
15989 ;          8287  !--
15990 ;          8288
15991 ;          8289  if .QUICK neq 0
15992 ;          8290  then
15993 ;          8291      begin                                   !THIS IS THE QUICK VERIFY PASS
15994 ;          8292      PRINTB (CRLF);
15995 ;          8293      PRINTB (SAY3, WRD2, PHR2, WRD4);
15996 ;          8294      !'BEGIN QUICK VERIFY PASS'
15997 ;          8295      INTEGRITY ();
15998 ;          8296      end;
15999 ;          8297
16000 ;          8298  if .DROP1 eql 0
16001 ;          8299  then
16002 ;          8300      OPT1 ();                                !OPTION 1 IS AVAILABLE
16003 ;          8301
16004 ;          8302  if .DROP2 eql 0
16005 ;          8303  then
16006 ;          8304      OPT2 ();                                !OPTION 2 IS AVAILABLE
16007 ;          8305
16008 ;          8306  if .DROP3 eql 0
16009 ;          8307  then
16010 ;          8308      OPT3 ();                                !OPTION 3 IS AVAILABLE
16011 ;          8309
16012 ;          8310  if .QUICK eql 0
16013 ;          8311  then
16014 ;          8312      begin
16015 ;          8313
16016 ;          8314      if .DROP4 eql 0
```

SEQ 0344

```
16018 ;MLX4
16019 ;                 THE OPTION SCHEDULER                    27-Mar-1982 19:24:42    TOPS-20 Bliss-16 V2(212)
16020                                                           27-Mar-1982 19:23:44    PA:<NEALE>MLX4.BLI.5 (49)
16021 ;          8315       then
16022 ;          8316           OPT4 ();                        !OPTION 4 IS AVAILABLE
16023 ;          8317
16024 ;          8318       if .DROP5 eql 0
16025 ;          8319       then
16026 ;          8320           OPT5 ();                        !OPTION 5 IS AVAILABLE
16027 ;          8321
16028 ;          8322       end;
16029 ;          8323
16030 ;          8324  ENDTST;
16034
16035
16039 102536  005767  130132          $T1:    .SBTTL  $T1 THE OPTION SCHEDULER
16040 102542  001426                          TST     QUICK                   ;                       8289
16041 102544  012746  007066                  BEQ     1$
16042 102550  012746  000001                  MOV     #CRLF,-(SP)             ;                       8292
16043 102554  010600                          MOV     #1,-(SP)
16044 102556  104414                          MOV     SP,R0                   ; SP,*
16045 102560  012716  007212                  TRAP    14
16046 102564  012746  007730                  MOV     #WRD4,(SP)              ;                       8293
16047 102570  012746  007200                  MOV     #PHR2,-(SP)
16048 102574  012746  007112                  MOV     #WRD2,-(SP)
16049 102600  012746  000004                  MOV     #SAY3,-(SP)
16050 102604  010600                          MOV     #4,-(SP)
16051 102606  104414                          MOV     SP,R0                   ; SP,*
16052 102610  004767  144416                  TRAP    14
16053 102614  062706  000014                  JSR     PC,INTEGRITY            ;                       8295
16054 102620  005767  077412          1$:     ADD     #14,SP                  ;                       8291
16055 102624  001002                          TST     DROP1                   ;                       8298
16056 102626  004767  146626                  BNE     2$
16057 102632  005767  077402          2$:     JSR     PC,OPT1                 ;                       8300
16058 102636  001002                          TST     DROP2                   ;                       8302
16059 102640  004767  151210                  BNE     3$
16060 102644  005767  077372          3$:     JSR     PC,OPT2                 ;                       8304
16061 102650  001002                          TST     DROP3                   ;                       8306
16062 102652  004767  155174                  BNE     4$
16063 102656  005767  130012          4$:     JSR     PC,OPT3                 ;                       8308
16064 102662  001012                          TST     QUICK                   ;                       8310
16065 102664  005767  077354                  BNE     6$
16066 102670  001002                          TST     DROP4                   ;                       8314
16067 102672  004767  157546                  BNE     5$
16068 102676  005767  077346          5$:     JSR     PC,OPT4                 ;                       8316
16069 102702  001002                          TST     DROP5                   ;                       8318
16070 102704  004767  177362                  BNE     6$
16071 102710  000207                  6$:     JSR     PC,OPT5
16072                                         RTS     PC                      ;                       8320
                                                                              ;                       8262
```

```
                                ;MLX4
                                ;                       THE OPTION SCHEDULER
                                                                                    27-Mar-1982 19:24:42    TOPS
                                                                                    27-Mar-1982 19:23:44    PA:<

                                ; Routine Size:  54 words
                                ; Maximum stack depth per invocation:  6 words


                                        .SBTTL  T1 THE OPTION SCHEDULER
                                T1::
102712                          1$:     JSR     PC,$T1                          ;                       8322
004767  177620
104466                                  TRAP    66
006000                                  ROR     R0
103773                                  BLO     1$
000207                                  RTS     PC

                                ; Routine Size:  6 words
                                ; Maximum stack depth per invocation:  0 words
```

SEQ 0346

```
16109 ;MLX4
16110 ;                 END OF PASS SUMMARY                                  27-Mar-1982 19:24:42    TOPS-20 Bliss-16 V2(212)
16111 ;                                                                      27-Mar-1982 19:23:44    PA:<NEALE>MLX4.BLI.5 (50)
16112 ;          8325  %sbttl 'END OF PASS SUMMARY'
16113 ;          8326  routine EOP : novalue =
16114 ;          8327      begin
16115 ;          8328                                                         !* 1 *
16116 ;          8329  !++
16117 ;          8330  !   ROUTINE:      EOP
16118 ;          8331  !
16119 ;          8332  !   PURPOSE:      TO PRINT A STATUS REPORT FOR EACH DRIVE.  IF AN
16120 ;          8333  !                 ARRAY HAS ANY HARD OR SOFT ERRORS, THEN ITS ERROR
16121 ;          8334  !                 COUNT WILL APPEAR, AND IF THE COUNT IS TOO HIGH,
16122 ;          8335  !                 THEN A DIAGNOSIS TO RUN THE ML11 PROM MAINTENANCE
16123 ;          8336  !                 PROGRAM WILL ALSO APPEAR.
16124 ;          8337  !
16125 ;          8338  !                 THE FOLLOWING IS A SAMPLE REPORT FOR 2 DRIVES:
16126 ;          8339  !
16127 ;          8340  !
16128 ;          8341  !   PERFORMANCE SUMMARY
16129 ;          8342  !
16130 ;          8343  !   NUMBER OF MBYTES TRANSFERRED:
16131 ;          8344  !    1028 MBYTES WRITTEN
16132 ;          8345  !     250 MBYTES READ
16133 ;          8346  !    1145 MBYTES WRITE CHECKED
16134 ;          8347  !
16135 ;          8348  !   LOGICAL UNIT: 0    DRIVE: 1    SERIAL #: 1234
16136 ;          8349  !     SOFT ERROR COUNT:     9
16137 ;          8350  !       ARRAY  3:       9
16138 ;          8351  !     HARD ERROR COUNT:    11
16139 ;          8352  !       ARRAY  0:       6
16140 ;          8353  !       ARRAY 10:       2
16141 ;          8354  !       ARRAY 15:       3
16142 ;          8355  !     TRANSFER RETRIES:     0
16143 ;          8356  !
16144 ;          8357  !   LOGICAL UNIT: 1    DRIVE: 1    SERIAL #: 9876
16145 ;          8358  !   DRIVE DROPPED   (CONTROLLER FATAL ERROR)
16146 ;          8359  !     SOFT ERROR COUNT:   100
16147 ;          8360  !       ARRAY  1:       9
16148 ;          8361  !       ARRAY 13:      10  --> RUN ML11 PROM MAINTENANCE PROGRAM
16149 ;          8362  !       ARRAY 14:       1
16150 ;          8363  !       ARRAY 15:      80  --> RUN ML11 PROM MAINTENANCE PROGRAM
16151 ;          8364  !     HARD ERROR COUNT:     2
16152 ;          8365  !       ARRAY 14:       1
16153 ;          8366  !       ARRAY 15:       1
16154 ;          8367  !     TRANSFER RETRIES:     0
16155 ;          8368  !--
16156 ;          8369
16157 ;          8370      local
16158 ;          8371          SFT_TOT,
16159 ;          8372          HRD_TOT,
16160 ;          8373          TRY_TOT;
16161 ;          8374
16162 ;          8375      if .EOPSUM
16163 ;          8376      then                                               !THE OPERATOR HAS ALLOWED THE SUMMARY TO PRINT
```

```
16165 ;MLX4
16166 ;                 END OF PASS SUMMARY                                  27-Mar-1982 19:24:42    TOPS-20 Bliss-16 V2(217)
16167 ;                                                                      27-Mar-1982 19:23:44    PA:<NEALE>MLX4.BLI.5 (50)
16168 ;          8377           begin                                        !* 2 *
16169 ;          8378           PRINTB (CRLF);
16170 ;          8379           PRINTB (SAY1, PHR17);
16171 ;          8380           !'PERFORMANCE SUMMARY'
16172 ;          8381           PRINTB (CRLF);
16173 ;          8382           PRINTB (SAY1, PHR12);
16174 ;          8383           !NUMBER OF MBYTES TRANSFERED:
16175 ;          8384           PRINTB (FMT14, .WR_MILLIONS, PHR19);
16176 ;          8385           !'XXXXX MBYTES WRITTEN'
16177 ;          8386           PRINTB (FMT14, .RD_MILLIONS, PHR20);
16178 ;          8387           !'XXXXX MBYTES READ'
16179 ;          8388           PRINTB (FMT14, .WC_MILLIONS, PHR13);
16180 ;          8389           !'XXXXX MBYTES WRITE CHECKED'
16181 ;          8390
16182 ;          8391           incr LUN from 0 to (.L$UNIT - 1) do
16183 ;          8392               begin                                    !* 3 *
16184 ;          8393               UNIT = .DRIVE;
16185 ;          8394
16186 ;          8395               if .DPR eql 0
16187 ;          8396               then
16188 ;          8397                   PRINTB (FMT4A, PHR7, .LUN, WRD11, .DRIVE)
16189 ;          8398                   !'LOGICAL UNIT: X    DRIVE: Y'
16190 ;          8399               else
16191 ;          8400                   SAYWHO (.LUN);
16192 ;          8401
16193 ;          8402               !'LOGICAL UNIT: X    DRIVE: Y    SERIAL #: ZZZZ'
16194 ;          8403
16195 ;          8404               if .DROPT_DRIVES [.LUN] eql ACTIVE
16196 ;          8405               then
16197 ;          8406                   begin                                !* 4 *
16198 ;          8407                   PRINTB (SAY2, WRD11, WRD21);
16199 ;          8408                   !'DRIVE DROPPED'
16200 ;          8409
16201 ;          8410                   selectone .WHY_DROPT [.LUN] of
16202 ;          8411                       set
16203 ;          8412
16204 ;          8413                       [CODE_1] :
16205 ;          8414                           PRINTB (FMT2, CAUSE1);
16206 ;          8415                           !'   (NOT POWERED UP)'
16207 ;          8416
16208 ;          8417                       [CODE_2] :
16209 ;          8418                           PRINTB (FMT2, CAUSE2);
16210 ;          8419                           !'   (NOT AN ML11 UNIT)'
16211 ;          8420
16212 ;          8421                       [CODE_3] :
16213 ;          8422                           PRINTB (FMT2, CAUSE3);
16214 ;          8423                           !'   (OPERATOR SELECTED TEST LIMITS INCORRECTLY)'
16215 ;          8424
16216 ;          8425                       [CODE_4] :
16217 ;          8426                           PRINTB (FMT2, CAUSE4);
16218 ;          8427                           !'   (ALL RETRIES FAILED FOR A NON-FATAL ERROR)'
16219 ;          8428
```

SEQ 0348


```
16221 ;MLX4
16222 ;
16223 ;
16224 ;          8429                        [CODE 5] :
16225 ;          8430                            PRINTB (FMT2, CAUSE5);
16226 ;          8431                        !'   (CONTROLLER FATAL ERROR)'
16227 ;          8432
16228 ;          8433                        [CODE 6] :
16229 ;          8434                            PRINTB (FMT2, CAUSE6);
16230 ;          8435                        !'   (DRIVE FATAL ERROR)'
16231 ;          8436
16232 ;          8437                        [CODE 7] :
16233 ;          8438                            PRINTB (FMT2, CAUSE7);
16234 ;          8439                        !'   (ECC HARD ERROR)'
16235 ;          8440
16236 ;          8441                        [CODE 8] :
16237 ;          8442                            PRINTB (FMT2, CAUSE8);
16238 ;          8443                        !'   (ECC LOGIC FAILURE)'
16239 ;          8444                        tes;
16240 ;          8445
16241 ;          8446                    end;                                !* 4 *
16242 ;          8447
16243 ;          8448                SFT_TOT = 0;
16244 ;          8449                HRD_TOT = 0;
16245 ;          8450                TRY_TOT = 0;
16246 ;          8451
16247 ;          8452                incr ARRAY from 0 to 15 do
16248 ;          8453                    begin
16249 ;          8454                    SFT_TOT = .SFT_TOT + .SOFTS [.LUN, .ARRAY, 0, 16, 0];
16250 ;          8455                    HRD_TOT = .HRD_TOT + .HARDS [.LUN, .ARRAY, 0, 16, 0];
16251 ;          8456                    TRY_TOT = .TRY_TOT + .TRIES [.LUN, .ARRAY, 0, 16, 0];
16252 ;          8457                    end;
16253 ;          8458
16254 ;          8459                PRINTB (FMT7, PHR5, .SFT_TOT);
16255 ;          8460                !'  SOFT ERROR COUNT: DDDDD'
16256 ;          8461
16257 ;          8462                if .SFT_TOT neq 0
16258 ;          8463                then
16259 ;          8464
16260 ;          8465                    incr ARRAY from 0 to 15 do
16261 ;          8466
16262 ;          8467                        if .SOFTS [.LUN, .ARRAY, 0, 16, 0] neq 0
16263 ;          8468                        then
16264 ;          8469                            begin
16265 ;          8470                            PRINTB (FMT9, .ARRAY, .SOFTS [.LUN, .ARRAY, 0, 16, 0]);
16266 ;          8471                            !'     ARRAY XX: YYYYY'
16267 ;          8472  !
16268 ;          8473  ! VER CZMLBB THE FOLLOWING CODE HAS BEEN COMMENTED OUT BECAUSE SOFT ERRORS
16269 ;          8474  ! SHOULD NOT BE COUNTED TOWARD THE CALLING OUT OF THE PROM MAINENANCE PROGRAM
16270 ;          8475  !                           !+
16271 ;          8476  !                           ! NOW SEE IF THRESHOLDS FOR SOFT ERRORS
16272 ;          8477  !                           ! HAVE BEEN REACHED FOR THIS ARRAY BOARD:
16273 ;          8478  !                           !-
16274 ;          8479  !
16275 ;          8480  !                           if ((.ARR_TYP eql 0) and (.SOFTS [.LUN, .ARRAY, 0, 16, 0] geq S16K_LIMIT))
```

```
16277 ;MLX4
16278 ;
16279
16280 ;          8481  !                        then
16281 ;          8482  !                            PRINTB (FMT2, MSG2);
16282 ;          8483  !
16283 ;          8484  !                        !'  --> RUN ML11 PROM MAINTENANCE PROGRAM'
16284 ;          8485  !
16285 ;          8486  !                        if ((.ARR_TYP eql 1) and (.SOFTS [.LUN, .ARRAY, 0, 16, 0] geq S64K_LIMIT))
16286 ;          8487  !                        then
16287 ;          8488  !                            PRINTB (FMT2, MSG2);
16288 ;          8489  !
16289 ;          8490  !                        !'  --> RUN ML11 PROM MAINTENANCE PROGRAM'
16290 ;          8491                           end;
16291 ;          8492
16292 ;          8493                   PRINTB (FMT7, PHR18, .HRD_TOT);
16293 ;          8494                   !'  HARD ERROR COUNT: DDDDD'
16294 ;          8495
16295 ;          8496                   if .HRD_TOT neq 0
16296 ;          8497                   then
16297 ;          8498
16298 ;          8499                       incr ARRAY from 0 to 15 do
16299 ;          8500
16300 ;          8501                           if .HARDS [.LUN, .ARRAY, 0, 16, 0] neq 0
16301 ;          8502                           then
16302 ;          8503                               begin
16303 ;          8504                               PRINTB (FMT9, .ARRAY, .HARDS [.LUN, .ARRAY, 0, 16, 0]);
16304 ;          8505                               !'    ARRAY XX: YYYYY'
16305 ;          8506
16306 ;          8507                               !+
16307 ;          8508                               ! NOW SEE IF THRESHOLDS FOR HARD ERRORS
16308 ;          8509                               ! HAVE BEEN REACHED FOR THIS ARRAY BOARD:
16309 ;          8510                               !-
16310 ;          8511
16311 ;          8512  !
16312 ;          8513  ! VER CZMLBB CHANGED TESTING HARDS [TABLE] TO TESTING PM_SBE_CNT [TABLE]
16313 ;          8514  !
16314 ;          8515
16315 ;          8516                               if ((.ARR_TYP eql 0) and (.PM_SBE_CNT [.LUN, .ARRAY, PM_SBE$_SUM] geq H16K_LIMIT))
16316 ;          8517                               then
16317 ;          8518                                   PRINTB (FMT2, MSG2);
16318 ;          8519
16319 ;          8520                               !'  --> RUN ML11 PROM MAINTENANCE PROGRAM'
16320 ;          8521  !
16321 ;          8522  ! VER CZMLBB CHANGED TESTING HARDS [TABLE] TO TESTING PM_SBE_CNT [TABLE]
16322 ;          8523  !
16323 ;          8524
16324 ;          8525                               if ((.ARR_TYP eql 1) and (.PM_SBE_CNT [.LUN, .ARRAY, PM_SBE$_SUM] geq H64K_LIMIT))
16325 ;          8526                               then
16326 ;          8527                                   PRINTB (FMT2, MSG2);
16327 ;          8528
16328 ;          8529                               !'  --> RUN ML11 PROM MAINTENANCE PROGRAM'
16329 ;          8530                               end;
16330 ;          8531
16331 ;          8532                       PRINTB (FMT7, PHR6, .TRY_TOT);
```

```
16333 ;MLX4
16334 ;                 END OF PASS SUMMARY
16335 ;
16336 ;          8533                   !'  TRANSFER RETRIES: DDDDD'
16337 ;          8534
16338 ;          8535                   if .TRY_TOT neq 0
16339 ;          8536                   then
16340 ;          8537
16341 ;          8538                       incr ARRAY from 0 to 15 do
16342 ;          8539
16343 ;          8540                           if .TRIES [.LUN, .ARRAY, 0, 16, 0] neq 0
16344 ;          8541                           then
16345 ;        P 8542                               PRINTB (FMT9, .ARRAY,
16346 ;        P 8543                                   .TRIES [.LUN,
16347 ;          8544                                       .ARRAY, 0, 16, 0]);
16348 ;          8545
16349 ;          8546                       !'    ARRAY XX: YYYYY'
16350 ;          8547                       end;                                    !* 3 *
16351 ;          8548
16352 ;          8549                   !+
16353 ;          8550                   ! Examine the number of single
16354 ;          8551                   ! bit errors detected during
16355 ;          8552                   ! execution of the exerciser
16356 ;          8553                   ! thus far.  If any were detected
16357 ;          8554                   ! then print them out to the
16358 ;          8555                   ! operator.
16359 ;          8556                   !-
16360 ;          8557
16361 ;          8558                   if .SBE$_COUNT gtr ZERO
16362 ;          8559                   then
16363 ;          8560                       begin
16364 ;          8561                       PRINTB (CRLF);
16365 ;          8562                       PRINTB (SAY1, PHR21);               !Print single bit log summary report
16366 ;          8563                       PRINTB (CRLF);
16367 ;          8564                       PRINTB (SAY1, SBE$HEADER);          !Print the column headings
16368 ;          8565
16369 ;          8566                       !+
16370 ;          8567                       ! Print to the console terminal for the
16371 ;          8568                       ! operator review all the detected single
16372 ;          8569                       ! bit errors during the execution of this
16373 ;          8570                       ! exerciser thus far.
16374 ;          8571                       !-
16375 ;          8572
16376 ;          8573                       incr index from 0 to .SBE$_COUNT - 1 do     !Print all errors logged in this table
16377 ;          8574                           begin
16378 ;        P 8575                           PRINTB (FMT16,                      !This is the printing format
16379 ;        P 8576                               .SBE_LOG [.index, UNIT$_SBE],       !Print the failing unit of this sbe
16380 ;        P 8577                               .SBE_LOG [.index, BRD$_SBE],        !Print the failing board number
16381 ;        P 8578                               .SBE_LOG [.index, BNK$_SBE],        !Print the failing bank number
16382 ;        P 8579                               .SBE_LOG [.index, BIT$_SBE],        !Print the failing bit number
16383 ;          8580                               .SBE_LOG [.index, SUM$_SBE]);       !Print the total sbe's found
16384 ;          8581                           end;
16385 ;          8582
16386 ;          8583                       end;
16387 ;          8584
```

SEQ 0351

```
16389 ;MLX4
16390 ;                 END OF PASS SUMMARY                    27-Mar-1982 19:24:42    TOPS-20 Bliss-16 V2(212)
16391                                                          27-Mar-1982 19:23:44    PA:<NEALE>MLX4.BLI.5 (50)
16392 ;          8585           PRINTB (CRLF);
16393 ;          8586           end;                                        !* 2 *
16394 ;          8587
16395 ;          8588       return;
16396 ;          8589       end;                                            !* 1 *
16400
16401                                                  .SBTTL  EOP END OF PASS SUMMARY
16405 102726  004167  102402          EOP:    JSR     P1,$SAVE5
16406 102732  162706  000010                  SUB     #10,SP                  ;                       8326
16407 102736  032767  000001  077312        BIT     #1,EOPSUM               ;
16408 102744  001002                        BNE     1$                      ;                       8375
16409 102746  000167  001744                JMP     32$
16410 102752  012746  007066          1$:     MOV     #CRLF,-(SP)
16411 102756  012746  000001                  MOV     #1,-(SP)                ;                       8378
16412 102762  010600                        MOV     SP,R0
16413 102764  104414                        TRAP    14                      ; SP,*
16414 102766  012716  010342                MOV     #PHR17,(SP)
16415 102772  012746  007072                MOV     #SAY1,-(SP)             ;                       8379
16416 102776  012746  000002                MOV     #2,-(SP)
16417 103002  010600                        MOV     SP,R0
16418 103004  104414                        TRAP    14                      ; SP,*
16419 103006  012716  007066                MOV     #CRLF,(SP)
16420 103012  012746  000001                MOV     #1,-(SP)                ;                       8381
16421 103016  010600                        MOV     SP,R0
16422 103020  104414                        TRAP    14                      ; SP,*
16423 103022  012716  010202                MOV     #PHR12,(SP)
16424 103026  012746  007072                MOV     #SAY1,-(SP)             ;                       8382
16425 103032  012746  000002                MOV     #2,-(SP)
16426 103036  010600                        MOV     SP,R0
16427 103040  104414                        TRAP    14                      ; SP,*
16428 103042  012716  010410                MOV     #PHR19,(SP)
16429 103046  016746  131246                MOV     WR.MILLIONS,-(SP)       ;                       8384
16430 103052  012746  007046                MOV     #FMT14,-(SP)
16431 103056  012746  000003                MOV     #3,-(SP)
16432 103062  010600                        MOV     SP,R0
16433 103064  104414                        TRAP    14                      ; SP,*
16434 103066  012716  010430                MOV     #PHR20,(SP)
16435 103072  016746  131230                MOV     RD.MILLIONS,-(SP)       ;                       8386
16436 103076  012746  007046                MOV     #FMT14,-(SP)
16437 103102  012746  000003                MOV     #3,-(SP)
16438 103106  010600                        MOV     SP,R0
16439 103110  104414                        TRAP    14                      ; SP,*
16440 103112  012716  010240                MOV     #PHR13,(SP)
16441 103116  016746  131212                MOV     WC.MILLIONS,-(SP)       ;                       8388
16442 103122  012746  007046                MOV     #FMT14,-(SP)
16443 103126  012746  000003                MOV     #3,-(SP)
```

SEQ 0352

```
16445                                          ;MLX4
16446                                          ;                    END OF PASS SUMMARY                                    27-Mar-1982 19:24:42    TOPS
16447                                                                                                                      27-Mar-1982 19:23:44    PA:<
16448 103132 010600                                    MOV     SP,R0                          ; SP,*
16449 103134 104414                                    TRAP    14
16450 103136 016766 076650 000046                      MOV     L$UNIT,46(SP)                  ;                                                  8391
16451 103144 005005                                    CLR     R5                             ; LUN
16452 103146 000167 001246                             JMP     27$
16453 103152 010500                            2$:     MOV     R5,R0                          ; LUN,*                                            8393
16454 103154 006300                                    ASL     R0
16455 103156 016003 034422                             MOV     PTABLE.ADDR(R0),R3
16456 103162 016304 000006                             MOV     6(R3),R4
16457 103166 042704 177770                             BIC     #177770,R4
16458 103172 142777 000007 131156                      BICB    #7,@ML.REG+10
16459 103200 150477 131152                             BISB    R4,@ML.REG+10
16460 103204 032777 000400 131146                      BIT     #400,@ML.REG+12                ;                                                  8395
16461 103212 001022                                    BNE     3$
16462 103214 016003 034422                             MOV     PTABLE.ADDR(R0),R3             ;                                                  8397
16463 103220 016346 000006                             MOV     6(R3),-(SP)
16464 103224 012746 007240                             MOV     #WRD11,-(SP)
16465 103230 010546                                    MOV     R5,-(SP)                       ; LUN,*
16466 103232 012746 010050                             MOV     #PHR7,-(SP)
16467 103236 012746 006216                             MOV     #FMT4A,-(SP)
16468 103242 012746 000005                             MOV     #5,-(SP)
16469 103246 010600                                    MOV     SP,R0                          ; SP,*
16470 103250 104414                                    TRAP    14
16471 103252 062706 000012                             ADD     #12,SP
16472 103256 000403                                    BR      4$
16473 103260 010546                            3$:     MOV     R5,-(SP)                       ;                                                  8395
16474 103262 004767 132236                             JSR     PC,SAYWHO                      ; LUN,*                                            8400
16475 103266 010500                            4$:     MOV     R5,R0                          ;                                                  8404
16476 103270 006200                                    ASR     R0                             ; LUN,*
16477 103272 006200                                    ASR     R0
16478 103274 006200                                    ASR     R0
16479 103276 062700 034444                             ADD     #DROPT.DRIVES,R0
16480 103302 010046                                    MOV     R0,-(SP)
16481 103304 010546                                    MOV     R5,-(SP)
16482 103306 042716 177770                             BIC     #177770,(SP)                   ; LUN,*
16483 103312 012746 000001                             MOV     #1,-(SP)
16484 103316 005046                                    CLR     -(SP)
16485 103320 004767 101032                             JSR     PC,BL$GT2
16486 103324 062706 000010                             ADD     #10,SP
16487 103330 005300                                    DEC     R0
16488 103332 001160                                    BNE     14$
16489 103334 012746 007322                             MOV     #WRD21,-(SP)                   ;                                                  8407
16490 103340 012746 007240                             MOV     #WRD11,-(SP)
16491 103344 012746 007100                             MOV     #SAY2,-(SP)
16492 103350 012746 000003                             MOV     #3,-(SP)
16493 103354 010600                                    MOV     SP,R0                          ; SP,*
16494 103356 104414                                    TRAP    14
16495 103360 005003                                    CLR     R3                             ;
16496 103362 156503 034446                             BISB    WHY.DROPT(R5),R3               ; *(LUN),*                                         8410
16497 103366 020327 000001                             CMP     R3,#1
16498 103372 001011                                    BNE     5$
16499 103374 012746 010460                             MOV     #CAUSE1,-(SP)                  ;                                                  8414
```

```
16501                                   ;MLX4
16502                                   .                       END OF PASS SUMMARY                                     27-Mar-1982 19:24:42    TOPS
16503                                                                                                                   27-Mar-1982 19:23:44    PA:<
16504 103400  012746  006160                       MOV     #FMT2,-(SP)
16505 103404  012746  000002                       MOV     #2,-(SP)
16506 103410  010600                               MOV     SP,R0
16507 103412  104414                               TRAP    14                      ; SP,*
16508 103414  000523                               BR      12$
16509 103416  020327  000002            5$:        CMP     R3,#2
16510 103422  001011                               BNE     6$                      ;                                       8410
16511 103424  012746  010502                       MOV     #CAUSE2,-(SP)
16512 103430  012746  006160                       MOV     #FMT2,-(SP)             ;                                       8418
16513 103434  012746  000002                       MOV     #2,-(SP)
16514 103440  010600                               MOV     SP,R0
16515 103442  104414                               TRAP    14                      ; SP,*
16516 103444  000507                               BR      12$
16517 103446  020327  000003            6$:        CMP     R3,#3
16518 103452  001011                               BNE     7$                      ;                                       8410
16519 103454  012746  010526                       MOV     #CAUSE3,-(SP)
16520 103460  012746  006160                       MOV     #FMT2,-(SP)             ;                                       8422
16521 103464  012746  000002                       MOV     #2,-(SP)
16522 103470  010600                               MOV     SP,R0
16523 103472  104414                               TRAP    14                      ; SP,*
16524 103474  000473                               BR      12$
16525 103476  020327  000004            7$:        CMP     R3,#4
16526 103502  001011                               BNE     8$                      ;                                       8410
16527 103504  012746  010602                       MOV     #CAUSE4,-(SP)
16528 103510  012746  006160                       MOV     #FMT2,-(SP)             ;                                       8426
16529 103514  012746  000002                       MOV     #2,-(SP)
16530 103520  010600                               MOV     SP,R0
16531 103522  104414                               TRAP    14                      ; SP,*
16532 103524  000457                               BR      12$
16533 103526  020327  000005            8$:        CMP     R3,#5
16534 103532  001011                               BNE     9$                      ;                                       8410
16535 103534  012746  010656                       MOV     #CAUSE5,-(SP)
16536 103540  012746  006160                       MOV     #FMT2,-(SP)             ;                                       8430
16537 103544  012746  000002                       MOV     #2,-(SP)
16538 103550  010600                               MOV     SP,R0
16539 103552  104414                               TRAP    14                      ; SP,*
16540 103554  000443                               BR      12$
16541 103556  020327  000006            9$:        CMP     R3,#6
16542 103562  001011                               BNE     10$                     ;                                       8410
16543 103564  012746  010710                       MOV     #CAUSE6,-(SP)
16544 103570  012746  006160                       MOV     #FMT2,-(SP)             ;                                       8434
16545 103574  012746  000002                       MOV     #2,-(SP)
16546 103600  010600                               MOV     SP,R0
16547 103602  104414                               TRAP    14                      ; SP,*
16548 103604  000427                               BR      12$
16549 103606  020327  000007            10$:       CMP     R3,#7
16550 103612  001011                               BNE     11$                     ;                                       8410
16551 103614  012746  010734                       MOV     #CAUSE7,-(SP)
16552 103620  012746  006160                       MOV     #FMT2,-(SP)             ;                                       8438
16553 103624  012746  000002                       MOV     #2,-(SP)
16554 103630  010600                               MOV     SP,R0
16555 103632  104414                               TRAP    14                      ; SP,*
```

```
                                    ;MLX4
                                    ;               END OF PASS SUMMARY

103634  000413                              BR      12$
103636  020327  000010              11$:    CMP     R3,#10                  ;                       8410
103642  00´012                              BNE     13$
103644  012746  010756                      MOV     #CAUSE8,-(SP)           ;                       8442
103650  012746  006160                      MOV     #FMT2,-(SP)
103654  012746  000002                      MOV     #2,-(SP)
103660  010600                              MOV     SP,R0                   ; SP,*
103662  104414                              TRAP    14
103664  062706  000006              12$:    ADD     #6,SP
103670  062706  000010              13$:    ADD     #10,SP                  ;                       8406
103674  005066  000046              14$:    CLR     46(SP)                  ; SFT.TOT               8448
103700  005066  000044                      CLR     44(SP)                  ; HRD.TOT               8449
103704  005066  000042                      CLR     42(SP)                  ; TRY.TOT               8450
103710  010500                              MOV     R5,R0                   ; LUN,*                 8454
103712  006300                              ASL     R0
103714  006300                              ASL     R0
103716  006300                              ASL     R0
103720  006300                              ASL     R0
103722  010004                              MOV     R0,R4
103724  005002                              CLR     R2                      ; ARRAY                 8452
103726  010403                      15$:    MOV     R4,R3                   ;                       8454
103730  060203                              ADD     R2,R3                   ; ARRAY,*
103732  006303                              ASL     R3
103734  066366  032714  000046              ADD     SOFTS(R3),46(SP)        ; *,SFT.TOT
103742  066366  033314  000044              ADD     HARDS(R3),44(SP)        ; *,HRD.TOT             8455
103750  066366  033714  000042              ADD     TRIES(R3),42(SP)        ; *,TRY.TOT             8455
103756  005202                              INC     R2                      ; ARRAY                 8452
103760  020227  000017                      CMP     R2,#17                  ; ARRAY,*
103764  003760                              BLE     15$
103766  016616  000046                      MOV     46(SP),(SP)             ; SFT.TOT,*             8459
103772  012746  010004                      MOV     #PHR5,-(SP)
103776  012746  006446                      MOV     #FMT7,-(SP)
104002  012746  000003                      MOV     #3,-(SP)
104006  010600                              MOV     SP,R0                   ; SP,*
104010  104414                              TRAP    14
104012  00576C  000054                      TST     54(SP)                  ; SFT.TOT               8462
104016  001425                              BEQ     18$
104020  005002                              CLR     R2                      ; ARRAY                 8465
104022  010403                      1C$.    MOV     R4,R3                   ;                       8467
104024  060203                              ADD     R2,R3                   ; ARRAY,*
104026  006303                              ASL     R3
104030  016303  032714                      MOV     SOFTS(R3),R3
104034  001412                              BEQ     17$
104036  010346                              MOV     R3,-(SP)                ;                       8470
104040  010246                              MOV     R2,-(SP)                ; ARRAY,*
104042  012746  006534                      MOV     #FMT9,-(SP)
104046  012746  000003                      MOV     #3,-(SP)
104052  010600                              MOV     SP,R0                   ; SP,*
104054  104414                              TRAP    14
104056  062706  000010                      ADD     #10,SP                  ;                       8469
104062  005202                      17$:    INC     R2                      ; ARRAY                 8465
104064  020227  000017                      CMP     R2,#17                  ; ARRAY,*
```

SEQ 0355

```
16613                                   ;MLX4
16614                                   ;
16615                                   ;       END OF PASS SUMMARY                                     27-Mar-1982 19:24:42    TOPS
16616 104070  003754                                                                                    27-Mar-1982 19:23:44    PA:<
16617 104072  016616  000052            18$:    BLE     16$
16618 104076  012746  010366                    MOV     52(SP),(SP)             ; HRD.TOT,*                                     8493
16619 104102  012746  006446                    MOV     #PHR18,-(SP)
16620 104106  012746  000003                    MOV     #FMT7,-(SP)
16621 104112  010600                            MOV     #3,-(SP)
16622 104114  104414                            MOV     SP,R0                   ; SP,*
16623 104116  005766  000060                    TRAP    14
16624 104122  001471                            TST     60(SP)                  ; HRD.TOT                                       8496
16625 104124  005001                            BEQ     23$
16626 104126  010403                            CLR     R1                      ; ARRAY                                         8499
16627 104130  060103                    19$:    MOV     R4,R3                   ;
16628 104132  006303                            ADD     R1,R3                   ; ARRAY,*                                       8501
16629 104134  016302  033314                    ASL     R3
16630 104140  001456                            MOV     HARDS(R3),R2
16631 104142  010246                            BEQ     22$
16632 104144  010146                            MOV     R2,-(SP)                ;
16633 104146  012746  006534                    MOV     R1,-(SP)                ; ARRAY,*                                       8504
16634 104152  012746  000003                    MOV     #FMT9,-(SP)
16635 104156  010600                            MOV     #3,-(SP)
16636 104160  104414                            MOV     SP,R0                   ; SP,*
16637 104162  032777  002000  130202            TRAP    14
16638 104170  001016                            BIT     #2000,@ML.REG+24        ;                                               8516
16639 104172  026327  012264  000012            BNE     20$
16640 104200  002412                            CMP     PM.SBE.CNT(R3),#12
16641 104202  012746  011120                    BLT     20$
16642 104206  012746  006160                    MOV     #MSG2,-(SP)             ;                                               8518
16643 104212  012746  000002                    MOV     #FMT2,-(SP)
16644 104216  010600                            MOV     #2,-(SP)
16645 104220  104414                            MOV     SP,R0                   ; SP,*
16646 104222  062706  000006                    TRAP    14
16647 104226  032777  002000  130136  20$:     ADD     #6,SP
16648 104234  001416                            BIT     #2000,@ML.REG+24        ;                                               8525
16649 104236  026327  012264  000012            BEQ     21$
16650 104244  002412                            CMP     PM.SBE.CNT(R3),#12
16651 104246  012746  011120                    BLT     21$
16652 104252  012746  006160                    MOV     #MSG2,-(SP)             ;                                               8527
16653 104256  012746  000002                    MOV     #FMT2,-(SP)
16654 104262  010600                            MOV     #2,-(SP)
16655 104264  104414                            MOV     SP,R0                   ; SP,*
16656 104266  062706  000006                    TRAP    14
16657 104272  062706  000010            21$:    ADD     #6,SP
16658 104276  005201                    22$:    ADD     #10,SP                  ;                                               8503
16659 104300  020127  000017                    INC     R1                      ; ARRAY                                         8499
16660 104304  003710                            CMP     R1,#17                  ; ARRAY,*
16661 104306  016616  000056                    BLE     19$
16662 104312  012746  010026            23$:    MOV     56(SP),(SP)             ; TRY.TOT,*                                     8532
16663 104316  012746  006446                    MOV     #PHR6,-(SP)
16664 104322  012746  000003                    MOV     #FMT7,-(SP)
16665 104326  010600                            MOV     #3,-(SP)
16666 104330  104414                            MOV     SP,R0                   ; SP,*
16667 104332  005766  000064                    TRAP    14
                                                TST     64(SP)                  ; TRY.TOT                                       8535
```

SEQ 0356

```
16669
16670                                   ;MLX4
16671                                   ;       END OF PASS SUMMARY                                     27-Mar-1982 19:24:42    TOPS
16672 104336  001425                                                                                    27-Mar-1982 19:23:44    PA:<
16673 104340  005002                            BEQ     26$
16674 104342  010403                            CLR     R2
16675 104344  060203                    24$:    MOV     R4,R3                   ; ARRAY                                         8538
16676 104346  006303                            ADD     R2,R3                   ;                                               8540
16677 104350  016303  033714                    ASL     R3                      ; ARRAY,*
16678 104354  001412                            MOV     TRIES(R3),R3
16679 104356  010346                            BEQ     25$
16680 104360  010246                            MOV     R3,-(SP)                ;                                               8544
16681 104362  012746  006534                    MOV     R2,-(SP)                ; ARRAY,*
16682 104366  012746  000003                    MOV     #FMT9,-(SP)
16683 104372  010600                            MOV     #3,-(SP)
16684 104374  104414                            MOV     SP,R0                   ; SP,*
16685 104376  062706  000010                    TRAP    14
16686 104402  005202                            ADD     #10,SP
16687 104404  020227  000017            25$:    INC     R2                      ; ARRAY                                         8538
16688 104410  003754                            CMP     R2,#17                  ; ARRAY,*
16689 104412  062706  000024                    BLE     24$
16690 104416  005205                    26$:    ADD     #24,SP                  ;                                               8392
16691 104420  020566  000046                    INC     R5                      ; LUN                                           8391
16692 104424  002002            27$:    CMP     R5,46(SP)               ; LUN,*
16693 104426  000167  176520                    BGE     28$
16694 104432  005767  106226                    JMP     2$
16695 104436  003517            28$:    TST     SBE$.COUNT              ;                                               8558
16696 104440  012746  007066                    BLE     31$
16697 104444  012746  000001                    MOV     #CRLF,-(SP)             ;                                               8561
16698 104450  010600                            MOV     #1,-(SP)
16699 104452  104414                            MOV     SP,R0                   ; SP,*
16700 104454  012716  005662                    TRAP    14
16701 104460  012746  007072                    MOV     #PHR21,(SP)             ;                                               8562
16702 104464  012746  000002                    MOV     #SAY1,-(SP)
16703 104470  010600                            MOV     #2,-(SP)
16704 104472  104414                            MOV     SP,R0                   ; SP,*
16705 104474  012716  007065                    TRAP    14
16706 104500  012746  000001                    MOV     #CRLF,(SP)              ;                                               8563
16707 104504  010600                            MOV     #1,-(SP)
16708 104506  104414                            MOV     SP,R0                   ; SP,*
16709 104510  012716  006042                    TRAP    14
16710 104514  012746  007072                    MOV     #SBE$HEADER,(SP)        ;                                               8564
16711 104520  012746  000002                    MOV     #SAY1,-(SP)
16712 104524  010600                            MOV     #2,-(SP)
16713 104526  104414                            MOV     SP,R0                   ; SP,*
16714 104530  016705  106130                    TRAP    14
16715 104534  005002                            MOV     SBE$.COUNT,R5           ;                                               8573
16716 104536  000453                            CLR     R2                      ; INDEX
16717 104540  010203                            BR      30$
16718 104542  006303            29$:    MOV     R2,R3                   ; INDEX,*                                       8580
16719 104544  006303                            ASL     R3
16720 104546  016346  011266                    ASL     R3
16721 104552  062703  011264                    MOV     SBE.LOG+2(R3),-(SP)
16722 104556  011304                            ADD     #SBE.LOG,R3
16723 104560  006204                            MOV     (R3),R4
                                                ASR     R4
```

SEQ 0357

```
16725                                   ;MLX4
16726                                   ;                       END OF PASS SUMMARY                     27-Mar-1982 19:24:42    TOPS
16727                                                                                                   27-Mar-1982 19:23:44    PA:<
16728 104562  006204                            ASR     R4
16729 104564  006204                            ASR     R4
16730 104566  006204                            ASR     R4
16731 104570  006204                            ASR     R4
16732 104572  006204                            ASR     R4
16733 104574  042704  177600                    BIC     #177600,R4
16734 104600  010446                            MOV     R4,-(SP)
16735 104602  111346                            MOVB    (R3),-(SP)
16736 104604  042716  177774                    BIC     #177774,(SP)
16737 104610  111304                            MOVB    (R3),R4
16738 104612  006204                            ASR     R4
16739 104614  006204                            ASR     R4
16740 104616  042704  177760                    BIC     #177760,R4
16741 104622  010446                            MOV     R4,-(SP)
16742 104624  011304                            MOV     (R3),R4
16743 104626  006104                            ROL     R4
16744 104630  006104                            ROL     R4
16745 104632  006104                            ROL     R4
16746 104634  006104                            ROL     R4
16747 104636  042704  177770                    BIC     #177770,R4
16748 104642  010446                            MOV     R4,-(SP)
16749 104644  012746  005736                    MOV     #FMT16,-(SP)
16750 104650  012746  000006                    MOV     #6,-(SP)
16751 104654  010600                            MOV     SP,R0                   ; SP,*
16752 104656  104414                            TRAP    14                      ;
16753 104660  062706  000016                    ADD     #16,SP                  ;                                       8574
16754 104664  005202                            INC     R2                      ; INDEX                                 8573
16755 104666  020205                    30$:    CMP     R2,R5                   ; INDEX,*
16756 104670  002723                            BLT     29$                     ;
16757 104672  062706  000016                    ADD     #16,SP                  ;                                       8560
16758 104676  012716  007066            31$:    MOV     #CRLF,(SP)              ;                                       8585
16759 104702  012746  000001                    MOV     #1,-(SP)
16760 104706  010600                            MOV     SP,R0                   ; SP,*
16761 104710  104414                            TRAP    14                      ;
16762 104712  062706  000042                    ADD     #42,SP                  ;
16763 104716  062706  000010            32$:    ADD     #10,SP                  ;                                       8377
16764 104722  000207                            RTS     PC                      ;                                       8326
16765
16766                                   ; Routine Size:  511 words
16767                                   ; Maximum stack depth per invocation:  40 words
16772
16773
```

```
16775 ;MLX4
16776 ;
16777 ;                 CLEANUP CODING SECTION
16778 ;          8590  %sbttl 'CLEANUP CODING SECTION'
16779 ;          8591  BGNCLN;
16780 ;          8592  !+
16781 ;          8593  ! THE CLEANUP CODING SECTION IS EXECUTED AFTER
16782 ;          8594  ! EACH PASS THROUGH THE EXERCISER.
16783 ;          8595  !-
16784 ;          8596  EOP_COUNT = .EOP_COUNT + 1;
16785 ;          8597
16786 ;          8598  if .EOP_COUNT eql 1
16787 ;          8599  then
16788 ;          8600      PRINTB (SAY3, WRD3, PHR2, WRD4)
16789 ;          8601      !'END QUICK VERIFY PASS'
16790 ;          8602  else
16791 ;          8603      PRINTB (FMT3, WRD3, WRD4, .EOP_COUNT);
16792 ;          8604
16793 ;          8605  !'** END PASS XX **'
16794 ;          8606  EOP ();
16795 ;          8607  return;
16796 ;          8608  ENDCLN;
16800
16801                                          .SBTTL  LCLEAN CLEANUP CODING SECTION
16805 104724 005267 125754           LCLEAN: INC     EOP.COUNT                       ;                    8596
16806 104730 026727 125750 000001            CMP     EOP.COUNT,#1                    ;                    8598
16807 104736 001015                          BNE     1$
16808 104740 012746 007212                   MOV     #WRD4,-(SP)                     ;                    8600
16809 104744 012746 007730                   MOV     #PHR2,-(SP)
16810 104750 012746 007206                   MOV     #WRD3,-(SP)
16811 104754 012746 007112                   MOV     #SAY3,-(SP)
16812 104760 012746 000004                   MOV     #4,-(SP)
16813 104764 010600                          MOV     SP,R0                           ; SP,*
16814 104766 104414                          TRAP    14
16815 104770 000414                          BR      2$
16816 104772 016746 125706           1$:     MOV     EOP.COUNT,-(SP)                 ;                    8598
16817 104776 012746 007212                   MOV     #WRD4,-(SP)                     ;                    8603
16818 105002 012746 007206                   MOV     #WRD3,-(SP)
16819 105006 012746 006166                   MOV     #FMT3,-(SP)
16820 105012 012746 000004                   MOV     #4,-(SP)
16821 105016 010600                          MOV     SP,R0                           ; SP,*
16822 105020 104414                          TRAP    14
16823 105022 004767 175700           2$:     JSR     PC,EOP
16824 105026 062706 000012                   ADD     #12,SP                          ;                    8606
16825 105032 000207                          RTS     PC                              ;                    8589
16826
16827
16828                                    ; Routine Size:  36 words
                                        ; Maximum stack depth per invocation:  5 words
```

SEQ 0359

```
16837
16838
16842
16843
16847 105034                            .SBTTL  L$CLEAN CLEANUP CODING SECTION
16848 105034  004767  177664     L$CLEAN::
16849 105040  104412                    JSR     PC,LCLEAN                       ;  8607
16850 105042  000207                    TRAP    12
16851                                   RTS     PC
16852
16853                           ; Routine Size:  4 words
16858                           ; Maximum stack depth per invocation:  0 words
16859
16860 ;          8609  LASTAD;
16861 ;          8610  BGNSETUP (0);
16862 ;          8611  ENDSETUP;
16866
16867
16868 105044  105050             BL$LAS::.WORD   T$FREE
16869 105046  000000                     .WORD   <<T$FREE-<BL$LAS+4>>/2>
16870 105050  000000             T$FREE::.WORD   0
16871
16872
16873          105050             L$LAST==            BL$LAS+4
16874          000000             T$PTHV==            0
16875
16876
16877                                   .SBTTL  $END.LINK CLEANUP CODING SECTION
16881 105052                     $END.LINK::
16882 105052  000207                    RTS     PC                              ;  8608
16883
16884                           ; Routine Size:  1 word
```

SEQ 0360


```
16886                                   ;MLX4
16887                                   ;               CLEANUP CODING SECTION
16888
16889                                   ; Maximum stack depth per invocation:  0 words
16894
16895
16896 ;         8612  end
16897 ;         8613
16898 ;         8614  eludom
16902
16903                                   ; OTS external references
16904                                           .GLOBL  BL$GT2, $SAVE5, $SAVE4, $SAVE3
16905                                           .GLOBL  $SAVE2, BL$PU2, BL$SHF, BL$DIV
16906                                           .GLOBL  BL$MOD, BL$MUL
16907
16908
16909
16910
16911
16912
16913                                   ; Size:         10337 code + 5852 data words
16914                                   ; Run Time:     01:43.8
16915                                   ; Elapsed Time: 02:08.4
16916                                   ; Memory Used:  161 pages
16917                                   ; Compilation Complete
16918
16919         000001                            .END
```

SYMBOL TABLE

```
ADR    = 000020 G
ASSEMB= 000010
BANK     034344
BASE.A   032706
BIT.NU   012666
BIT0   = 000001 G
BIT00  = 000001 G
BIT01  = 000002 G
BIT02  = 000004 G
BIT03  = 000010 G
BIT04  = 000020 G
BIT05  = 000040 G
BIT06  = 000100 G
BIT07  = 000200 G
BIT08  = 000400 G
BIT09  = 001000 G
BIT1   = 000002 G
BIT10  = 002000 G
BIT11  = 004000 G
BIT12  = 010000 G
BIT13  = 020000 G
BIT14  = 040000 G
BIT15  = 100000 G
BIT2   = 000004 G
BIT3   = 000010 G
BIT4   = 000020 G
BIT5   = 000040 G
BIT6   = 000100 G
BIT7   = 000200 G
BIT8   = 000400 G
BIT9   = 001000 G
BL$DIV   005170 G
BL$GT1   004234 G
BL$GT2   004356 G
BL$LAS   105044 G
BL$MOD   005202 G
BL$MUL   004744 G
BL$PU1   004520 G
BL$PU2   004614 G
BL$SHF   005214 G
BOARD    034342
BOE    = 000400 G
BR.LEV   032712
CAUSE1= 010460
CAUSE2= 010502
CAUSE3= 010526
CAUSE4= 010602
CAUSE5= 010656
CAUSE6= 010710
CAUSE7= 010734
CAUSE8= 010756
CHECK    046000
CHOOSE   046166
CLRTBL   034554
COMP.C   032702
CONFIG   036002
CRLF   = 007066
C$AU   = 000052
C$AUTO= 000061
C$BRK  = 000022
C$BSEG= 000004
C$BSUB= 000002
C$CEFG= 000045
C$CLCK= 000062
C$CLEA= 000012
C$CLOS= 000035
C$CLP1= 000006
C$CVEC= 000036
C$DCLN= 000044
C$DODU= 000051
C$DRPT= 000024
C$DU   = 000053
C$EDIT= 000003
C$ERDF= 000055
C$ERHR= 000056
C$ERRO= 000060
C$ERSF= 000054
C$ERSO= 000057
C$ESCA= 000010
C$ESEG= 000005
C$ESUB= 000003
C$ETST= 000001
C$EXIT= 000032
C$GETB= 000026
C$GETW= 000027
C$GMAN= 000043
C$GPHR= 000042
C$GPLO= 000030
C$GPRI= 000040
C$INIT= 000011
C$INLP= 000020
C$MANI= 000050
C$MEM  = 000031
C$MSG  = 000023
C$OPEN= 000034
C$PNTB= 000014
C$PNTF= 000017
C$PNTS= 000016
C$PNTX= 000015
C$QIO  = 000377
C$RDBU= 000007
C$REFG= 000047
C$RESE= 000033
C$REVI= 000003
C$RFLA= 000021
C$RPT  = 000025
C$SEFG= 000046
C$SPRI= 000041
C$SVEC= 000037
C$TPRI= 000013
DATA.C   032700
DECODE   037176
DFPTBL   002210 G
DIAGMC= 000000
DIVMOD   005006
DOUBLE   044326
DRIVE.   034442 G
DROPNE   002234 G
DROPT.   034444 G
DROP1    002236 G
DROP2    002240 G
DROP3    002242 G
DROP4    002244 G
DROP5    002250 G
ECCDIS   002254 G
EFN$21   002262 G
EF.CON= 000036 G
EF.NEW= 000035 G
EF.PWR= 000034 G
EF.RES= 000037 G
EF.STA= 000040 G
END.RB= 032670
END.WB= 022670
EOP      102726 G
EOPSUM   002256 G
EOP.CO   032704
ERRBLK   002200 G
ERRMSG   002176 G
ERRNBR   002174 G
ERROUT   002260 G
ERRTYP   002172 G
EVL    = 000004 G
ESEND  = 002100
ESLOAD= 000035
FILLER   041266
FMT1A  = 006126
FMT1B  = 006142
FMT10A= 006564
FMT10B= 006640
FMT11  = 006654
FMT12A= 006710
FMT12B= 006756
FMT13  = 007024
FMT14  = 007046
FMT15  = 007060
FMT16  = 005736
FMT2   = 006160
FMT3   = 006166
FMT4A  = 006216
FMT4B  = 006252
FMT4C  = 006270
FMT5   = 006320
FMT6   = 006376
FMT7   = 006446
FMT8   = 006466
FMT9   = 006534
F$AU   = 000015
F$AUTO= 000020
F$BGN  = 000040
F$CLEA= 000007
F$DU   = 000016
F$END  = 000041
F$HARD= 000004
F$HW   = 000013
F$INIT= 000006
F$JMP  = 000050
F$MOD  = 000000
F$MSG  = 000011
F$PROT= 000021
F$PWR  = 000017
F$RPT  = 000012
F$SEG  = 000003
F$SOFT= 000005
F$SRV  = 000010
F$SUB  = 000002
F$SW   = 000014
F$TEST= 000001
GEN1     040732
GEN2     041432
GEN3     041466
GEN4     041564
GEN5     041620
GETCNT   041106
GET.WR   047176
G$CNTO= 000200
G$DELM= 000372
G$DISP= 000003
G$EXCP= 000400
G$HILI= 000002
G$LOLI= 000001
G$NO   = 000000
G$OFFS= 000400
G$OFSI= 000376
G$PRMA= 000001
G$PRMD= 000002
G$PRML= 000000
G$RADA= 000140
G$RADB= 000100
G$RADD= 000040
G$RADL= 000120
G$RADO= 000020
G$XFER= 000004
G$YES  = 000010
HARDS    033314
HELP   = 000000
HOE    = 100000 G
IBE    = 010000 G
IDU    = 000040 G
IER    = 020000 G
INIT.A   035206
INTEGR   047232
ISOLAT   037332
ISR    = 000100 G
IXE    = 004000 G
I$AU   = 000041
I$AUTO= 000041
I$CLN  = 000041
I$DU   = 000041
I$HRD  = 000041
I$INIT= 000041
I$MOD  = 000041
I$MSG  = 000041
I$PROT= 000040
I$PTAB= 000041
I$PWR  = 000041
I$RPT  = 000041
I$SEG  = 000041
I$SETU= 000041
I$SFT  = 000041
I$SRV  = 000041
I$SUB  = 000041
I$TST  = 000041
I.AM.D   034336
J$JMP  = 000167
LAU      005650
LAUTO    005504
LCLEAN   104724
LDU      005516
LIMIT    002222 G
LINIT    040474
LOE    = 040000 G
LOT    = 000010 G
LOW.SE   034460 G
LRPT     005466
LSECT    002226 G
L$ACP    002110 G
L$APT    002036 G
L$AU     005652 G
L$AUT    002070 G
L$AUTO   005506 G
L$CCP    002106 G
L$CLEA   105034 G
L$CO     002032 G
L$DEPO   002011 G
L$DESC   002130 G
L$DESP   002076 G
L$DEVP   002060 G
L$DISP   002204 G
L$DLY    002116 G
L$DTP    002040 G
L$DTYP   002034 G
L$DU     005640 G
L$DUT    002072 G
L$DVTY   002122 G
L$EF     002052 G
L$ENVI   002044 G
L$ERRT   002172 G
L$ETP    002102 G
L$EXP1   002046 G
L$EXP4   002064 G
L$EXP5   002066 G
L$HARD   002266 G
L$HIME   002120 G
L$HPCP   002016 G
L$HPTP   002022 G
```

N 12

SEQ 0362

```
L$HW    002210 G        MSG4   = 011202        P.AAD   006126        P.ACI   007500        P.AEN   011070
L$ICP   002104 G        MSG5   = 011216        P.AAE   006142        P.ACJ   007504        P.AEO   011120
L$INIT  040722 G        NUM.DR  034456 G       P.AAF   006160        P.ACK   007510        P.AEP   011166
L$LADP  002026 G        ONEFIL= 000001          P.AAG   006166        P.ACL   007514        P.AEQ   011202
L$LAST= 105050 G        ONLY    002232 G        P.AAH   006216        P.ACM   007520        P.AER   011216
L$LOAD  002100 G        OPT1    051460          P.AAI   006252        P.ACN   007524        QH1     002332
L$LUN   002074 G        OPT2    054054          P.AAJ   006270        P.ACO   007530        QH2     002351
L$MREV  002050 G        OPT3    060052          P.AAK   006320        P.ACP   007534        QH3     002374
L$NAME  002000 G        OPT4    062444          P.AAL   006376        P.ACQ   007542        QH4     002425
L$PRIO  002042 G        OPT5    102272          P.AAM   006446        P.ACR   007550        QS1     002664
L$PROT  004126 G        O$APTS= 000001          P.AAN   006466        P.ACS   007554        QS10    003534
L$PRT   002112 G        O$AU   = 000001         P.AAO   006534        P.ACT   007560        QS11    003554
L$REPP  002062 G        O$BGNR= 000001          P.AAP   006564        P.ACU   007564        QS12    003637
L$REV   002010 G        O$BGNS= 000001          P.AAQ   006640        P.ACV   007570        QS13    003657
L$RPT   005474 G        O$DU   = 000001         P.AAR   006654        P.ACW   007574        QS14    003724
L$SOFT  002450 G        O$ERRT= 000001          P.AAS   006710        P.ACX   007626        QS15    003763
L$SPC   002056 G        O$GNSW= 000001          P.AAT   006756        P.ACY   007634        QS16    004030
L$SPCP  002020 G        O$POIN= 000001          P.AAU   007024        P.ACZ   007642        QS17    004066
L$SPTP  002024 G        O$SETU= 000001          P.AAV   007046        P.ADA   007650        QS2     002732
L$STA   002030 G        PATTBL  034520 G        P.AAW   007060        P.ADB   007656        QS3     003007
L$SW    002222 G        PATTER  032676          P.AAX   007066        P.ADC   007664        QS4     003047
L$TEST  002114 G        PHR1   = 007714         P.AAY   007072        P.ADD   007672        QS5     003107
L$TIML  002014 G        PHR10  = 010120         P.AAZ   007100        P.ADE   007700        QS6     003147
L$UNIT  002012 G        PHR11  = 010140         P.ABA   007112        P.ADF   007706        QS7     003454
L10000  002220          PHR12  = 010202         P.ABB   007130        P.ADG   007714        QS8     003474
L10001  002264          PHR13  = 010240         P.ABC   007152        P.ADH   007730        QS9     003514
L10002  002332          PHR14  = 010266         P.ABD   007200        P.ADI   007746        QUICK   032674
L10003  002664          PHR15  = 010304         P.ABE   007206        P.ADJ   007770        RANDOM  005464 G
MARPAT  002246 G        PHR16  = 010322         P.ABF   007212        P.ADK   010004        RAND1   070540
MLB10  = 007472         PHR17  = 010342         P.ABG   007220        P.ADL   010026        RAND2   073124
MLB11  = 007500         PHR18  = 010366         P.ABH   007230        P.ADM   010050        RAND3   075444
MLB12  = 007504         PHR19  = 010410         P.ABI   007240        P.ADN   010066        RAND4   100006
MLB13  = 007510         PHR2   = 007730         P.ABJ   007246        P.ADO   010100        RANGE   002224 G
MLB14  = 007514         PHR20  = 010430         P.ABK   007256        P.ADP   010120        RBUFF   022670
MLB15  = 007520         PHR21  = 005662         P.ABL   007264        P.ADQ   010140        RCBUFF= 023670
MLB16  = 007524         PHR3   = 007746         P.ABM   007272        P.ADR   010202        RDBUFF= 022670
MLB17  = 007530         PHR4   = 007770         P.ABN   007300        P.ADS   010240        RD.COU  034322
MLB18  = 007534         PHR5   = 010004         P.ABO   007312        P.ADT   010266        RD.MIL  034326
MLB19  = 007542         PHR6   = 010026         P.ABP   007322        P.ADU   010304        RD.THO  034324
MLB2   = 007430         PHR7   = 010050         P.ABQ   007332        P.ADV   010322        READ    045612
MLB20  = 007550         PHR8   = 010066         P.ABR   007336        P.ADW   010342        REFRES  002252 G
MLB21  = 007554         PHR9   = 010100         P.ABS   007344        P.ADX   010366        RETRY   046224
MLB22  = 007560         PM.SBE  012264          P.ABT   007354        P.ADY   010410        RETRYI  034340
MLB23  = 007564         PNT    = 001000 G       P.ABU   007370        P.ADZ   010430        RE2     005362
MLB24  = 007570         PRI    = 002000 G       P.ABV   007400        P.AEA   010444        RE3     005360
MLB3   = 007434         PRI00  = 000000 G       P.ABW   007412        P.AEB   010446        RE4     005356
MLB4   = 007440         PRI01  = 000040 G       P.ABX   007416        P.AEC   010450        RN      005370 G
MLB5   = 007444         PRI02  = 000100 G       P.ABY   007424        P.AED   010454        RNDSEC  070240
MLB6   = 007450         PRI03  = 000140 G       P.ABZ   007430        P.AEE   010460        RNDU    070336
MLB7   = 007454         PRI04  = 000200 G       P.ACA   007434        P.AEF   010502        RNDWC   070204
MLB8   = 007460         PRI05  = 000240 G       P.ACB   007440        P.AEG   010526        RPTR    032672
MLB9   = 007464         PRI06  = 000300 G       P.ACC   007444        P.AEH   010602        RTN0   = 007574
ML.REG  034346 G        PRI07  = 000340 G       P.ACD   007450        P.AEI   010656        RTN1   = 007626
MSG0   = 011002         PTABLE  034422 G        P.ACE   007454        P.AEJ   010710        RTN2   = 007634
MSG1   = 011070         P.AAA   005662          P.ACF   007460        P.AEK   010734        RTN3   = 007642
MSG2   = 011120         P.AAB   005736          P.ACG   007464        P.AEL   010756        RTN4   = 007650
MSG3   = 011166         P.AAC   006042          P.ACH   007472        P.AEM   011002        RTN5   = 007656
```

SEQ 0363

```
RTN5A = 007664          SVCSUB= 000001          T$NS1 = 000021          VEC     032710          WRD36 = 007370
RTN5B = 007672          SVCTAG= 000001          T$PTHV= 000000 G        WAITER  044420          WRD37 = 007400
RTN5C = 007700          SVCTST= 000001          T$PTNU= 000000          WBUFF   012670          WRD38 = 007412
RTN5D = 007706          SYSERR  041710          T$SAVL= 177777          WCBUFF= 013670          WRD4  = 007212
SAYWHO  035524          S$LSYM= 010000          T$SEGL= 177777          WC.COU  034330          WRD40 = 007416
SAY1  = 007072          TOP.SE  034500 G        T$SUBN= 000000          WC.MIL  034334          WRD41 = 007424
SAY2  = 007100          TRIES   033714          T$TAGL= 177777          WC.THO  034332          WRD6  = 007220
SAY3  = 007112          TRT00 = 010444          T$TAGN= 010005          WDBUFF= 012670          WRD7  = 007230
SAY4  = 007130          TRT01 = 010446          T$TEMP= 000000          WHY.DR  034446 G        WRITE   045424
SAY5  = 007152          TRT10 = 010450          T$TEST= 000000          WPTR    032670          WR.COU  034314
SBE$HE= 006042          TRT11 = 010454          T$TSTM= 177777          WRD11 = 007240          WR.MIL  034320
SBE$.C  012664          TSECT   002230 G        T$TSTS= 000000          WRD15 = 007246          WR.THO  034316
SBE.LO  011264          T$ARGC= 000002          T$$HAR= 010002          WRD16 = 007256          X$ALWA= 000000
SEED1   005456 G        T$CODE= 020130          T$$HW = 010000          WRD17 = 007264          X$FALS= 000040
SEED2   005460 G        T$ERRN= 000000          T$$PRO= 010004          WRD18 = 007272          X$OFFS= 000400
SEED3   005462 G        T$EXCP= 000000          T$$SOF= 010003          WRD19 = 007300          X$TRUE= 000020
SELPAT  041066          T$FREE  105050 G        T$$SW = 010001          WRD2  = 007200          $END.L  105052 G
SERVIC  034544 G        T$GMAN= 000000          T1      102712 G        WRD20 = 007312          $PATCH  004134 G
SET.PT  047132          T$HILI= 000012          UAM   = 000200 G        WRD21 = 007322          $SAVE2  005262 G
SFPTBL  002222 G        T$LAST= 000000          UP.HAR  037422          WRD24 = 007332          $SAVE3  005276 G
SOFTS   032714          T$LOLI= 000001          UP.RD.  045240          WRD25 = 007336          $SAVE4  005314 G
START.  044424          T$LSYM= 010000          UP.SOF  040104          WRD3  = 007206          $SAVE5  005334 G
SVCGBL= 000001          T$NEST= 177777          UP.WC.  045332          WRD34 = 007344          $T1     102536
SVCINS= 000001          T$NS0 = 000000          UP.WR.  045146          WRD35 = 007354

. ABS.  105054      000
        000000      001
ERRORS DETECTED:  0

VIRTUAL MEMORY USED:  31919 WORDS  ( 125 PAGES)
DYNAMIC MEMORY:  21558 WORDS  ( 82 PAGES)
ELAPSED TIME:  00:09:52
CZMLBB.BIN,CZMLBB/CR/-SP=SVC/ML,CZMLBB.DOC,MLX2,OTS,RANDOM,MLX3,MLX4
```

SYMBOL	VALUE		REFERENCES								
ADR	= 000020	G	#83-1773								
ASSEMB	= 000010		6-13	6-13							
BANK	034344		#82-1685	*98-2978	*99-2991	100-3052	103-3237	105-3302	108-3485	109-3530	155-6371
			*155-6383								
BASE.A	032706		#81-1650	*88-2289	89-2338	97-2909					
BIT.NU	012666		#81-1637	*99-3005	103-3226	104-3290	108-3474	109-3522			
BIT0	= 000001	G	#83-1754								
BIT00	= 000001	G	#83-1744								
BIT01	= 000002	G	#83-1743								
BIT02	= 000004	G	#83-1742								
BIT03	= 000010	G	#83-1741								
BIT04	= 000020	G	#83-1740								
BIT05	= 000040	G	#83-1739								
BIT06	= 000100	G	#83-1738								
BIT07	= 000200	G	#83-1737								
BIT08	= 000400	G	#83-1736								
BIT09	= 001000	G	#83-1735								
BIT1	= 000002	G	#83-1753								
BIT10	= 002000	G	#83-1734								
BIT11	= 004000	G	#83-1733								
BIT12	= 010000	G	#83-1732								
BIT13	= 020000	G	#83-1731								
BIT14	= 040000	G	#83-1730								
BIT15	= 100000	G	#83-1729								
BIT2	= 000004	G	#83-1752								
BIT3	= 000010	G	#83-1751								
BIT4	= 000020	G	#83-1750								
BIT5	= 000040	G	#83-1749								
BIT6	= 000100	G	#83-1748								
BIT7	= 000200	G	#83-1747								
BIT8	= 000400	G	#83-1746								
BIT9	= 001000	G	#83-1745								
BL$DIV	005170	G	#37-1170	185-8022	205-9118	209-9381	225-10222	273-13132	282-13730	297-14575	298-14593
			313-15422	341-16905							
BL$GT1	004234	G	#13-169								
BL$GT2	004356	G	#18-384	164-6887	180-7728	200-8836	205-9143	220-9931	242-11178	265-12395	273-12846
			281-13705	297-14560	322-15915	333-16485	341-16904				
BL$LAS	105044	G	#340-16868	340-16869	340-16873						
BL$MOD	005202	G	#38-1201	118-4116	261-12206	263-12297	265-12380	265-12404	341-16906		
BL$MUL	004744	G	#34-1022	95-2787	341-16906						
BL$PU1	004520	G	#23-592								
BL$PU2	004614	G	#29-829	47-220	47-231	50-339	85-2133	85-2140	111-3672	112-3685	118-4112
			341-16905								
BL$SHF	005214	G	13-181	13-184	13-188	13-192	14-217	19-406	19-409	19-413	19-417
			19-438	23-595	23-599	23-605	24-632	29-838	29-842	29-848	30-880
			#39-1236	88-2294	341-16905						
BOARD	034342		#82-1684	*99-2996	100-3053	104-3246	104-3271	105-3306	108-3490	109-3534	155-6370
			155-6379	*155-6382	168-7119	169-7140	171-7267	171-7284	185-7996	185-8013	187-8108
			187-8125	204-9088	205-9109	209-9350	209-9367	211-9467	211-9484	225-10196	225-10213
			227-10308	227-10325	247-11460	248-11481	252-11739	253-11760	257-12018	258-12039	259-12125
			259-12142	273-13106	273-13123	273-13216	273-13233	286-13980	287-14001	288-14085	289-14106
			302-14838	302-14855	304-14946	304-14963	318-15671	318-15688	319-15773	320-15794	

CZMLBB CREATED BY MACRO ON 29-MAR-82 AT 13:44 PAGE 2
SYMBOL CROSS REFERENCE CREF SEQ 0365

SYMBOL	VALUE		REFERENCES								
BOE	= 000400	G	#83-1777								
BR.LEV	032712		#81-1653	*88-2291	88-2292						
CAUSE1	= 010460		#83-1905	94-2733	333-16499						
CAUSE2	= 010502		#83-1906	94-2756	334-16511						
CAUSE3	= 010526		#83-1907	96-2831	96-2858	334-16519					
CAUSE4	= 010602		#83-1908	334-16527							
CAUSE5	= 010656		#83-1909	334-16535							
CAUSE6	= 010710		#83-1910	334-16543							
CAUSE7	= 010734		#83-1911	334-16551							
CAUSE8	= 010756		#83-1912	335-16563							
CHECK	046000		#147-5961	149-6059	153-6265	154-6301	154-6312	169-7147	169-7152	171-7238	182-7826
			201-8923	206-9185	222-10026	244-11290	249-11569	254-11848	273-12939	283-13815	299-14673
			302-14809	315-15506							
CHOOSE	046166		#149-6052	181-7804	201-8905	206-9166	221-10004	244-11269	249-11548	254-11827	273-12918
			283-13797	299-14655	314-15484						
CLRTBL	034554		#84-2078	111-3644							
COMP.C	032702		#81-1646	*116-3994	118-4102						
CONFIG	036002		#93-2703	112-3678							
CRLF	= 007066		#83-1814	88-2301	89-2330	154-6342	325-16041	332-16410	332-16419	337-16696	337-16705
			338-16758								
C$AU	= 000052		#6-13								
C$AUTO	= 000061		#6-13								
C$BRK	= 000022		#6-13								
C$BSEG	= 000004		#6-13								
C$BSUB	= 000002		#6-13								
C$CEFG	= 000045		#6-13								
C$CLCK	= 000062		#6-13								
C$CLEA	= 000012		#6-13								
C$CLOS	= 000035		#6-13								
C$CLP1	= 000006		#6-13								
C$CVEC	= 000036		#6-13								
C$DCLN	= 000044		#6-13								
C$DODU	= 000051		#6-13								
C$DRPT	= 000024		#6-13								
C$DU	= 000053		#6-13								
C$EDIT	= 000003		#6-13								
C$ERDF	= 000055		#6-13								
C$ERHR	= 000056		#6-13								
C$ERRO	= 000060		#6-13								
C$ERSF	= 000054		#6-13								
C$ERSO	= 000057		#6-13								
C$ESCA	= 000010		#6-13								
C$ESEG	= 000005		#6-13								
C$ESUB	= 000003		#6-13								
C$ETST	= 000001		#6-13								
C$EXIT	= 000032		#6-13								
C$GETB	= 000026		#6-13								
C$GETW	= 000027		#6-13								
C$GMAN	= 000043		#6-13								
C$GPHR	= 000042		#6-13								
C$GPLO	= 000030		#6-13								
C$GPRI	= 000040		#6-13								

CZMLBB CREATED BY MACRO ON 29-MAR-82 AT 13:44 PAGE 3
SYMBOL CROSS REFERENCE CREF SEQ 0366

SYMBOL	VALUE		REFERENCES								
C$INIT	= 000011		#6-13								
C$INLP	= 000020		#6-13								
C$MANI	= 000050		#6-13								
C$MEM	= 000031		#6-13								
C$MSG	= 000023		#6-13								
C$OPEN	= 000034		#6-13								
C$PNTB	= 000014		#6-13								
C$PNTF	= 000017		#6-13								
C$PNTS	= 000016		#6-13								
C$PNTX	= 000015		#6-13								
C$QIO	= 000377		#6-13								
C$RDBU	= 000007		#6-13								
C$REFG	= 000047		#6-13								
C$RESE	= 000033		#6-13	#6-13							
C$REVI	= 000003		#6-13	6-65							
C$RFLA	= 000021		#6-13								
C$RPT	= 000025		#6-13								
C$SFFG	= 000046		#6-13								
C$SPRI	= 000041		#6-13								
C$SVEC	= 000037		#6-13								
C$TPRI	= 000013		#6-13								
DATA.C	032700		#81-1644	*116-3993	118-4104						
DECODE	037176		#98-2972	100-3049	155-6373						
DFPTBL	002210	G	#7-117								
DIAGMC	= 000000		6-13	6-13							
DIVMOD	005006		#35-1091	37-1171	38-1202						
DOUBLE	044326		#133-5198	166-6966	182-7835	201-8931	206-9193	222-10035	244-11299	249-11578	254-11857
			273-12948	284-13828	299-14682	315-15515					
DRIVE.	034442	G	46-201	47-214	#82-1692	85-2128	111-3653	164-6881	180-7722	200-8830	205-9137
			220-9925	242-11172	265-12389	273-12840	281-13699	297-14554	322-15909		
DROPNE	002234	G	#8-160	83-1723	85-2146						
DROPT.	034444	G	46-201	47-225	#82-1695	85-2135	333-16479				
DROP1	002236	G	#8-161	83-1723	*85-2148	325-16054					
DROP2	002240	G	#8-162	83-1724	*85-2149	325-16057					
DROP3	002242	G	#8-163	83-1724	*85-2150	325-16060					
DROP4	002244	G	#8-164	83-1724	*85-2151	325-16065					
DROP5	002250	G	#8-166	83-1724	*85-2152	325-16068					
ECCDIS	002254	G	#8-169	83-1725	88-2296	131-4772	131-4807	131-4858	131-4956	138-5458	
EFN$21	002262	G	#8-192	83-1722	185-8027	210-9390	225-10227	258-12048	273-13137	287-14008	303-14869
			318-15697								
EF.CON	= 000036	G	#83-1757								
EF.NEW	= 000035	G	#83-1758								
EF.PWR	= 000034	G	#83-1759								
EF.RES	= 000037	G	#83-1756								
EF.STA	= 000040	G	#83-1755								
END.RB	= 032670		#83-1790								
END.WB	= 022670		#83-1789	156-6439							
EOP	102726	G	44-31	44-39	#332-16405	339-16823					
EOPSUM	002256	G	#8-188	83-1725	332-16407						
EOP.CO	032704		#81-1648	*84-2101	*339-16805	339-16806	339-16816				
ERRBLK	002200	G	#7-94								
ERRMSG	002176	G	#7-94								

CZMLBB CREATED BY MACRO ON 29-MAR-82 AT 13:44 PAGE 4

SYMBOL CROSS REFERENCE CREF SEQ 0367

SYMBOL	VALUE		REFERENCES
ERRNBR	002174	G	#7-94
ERROUT	002260	G	#8-190 83-1725 100-3050 131-4818 131-5100 144-5788 146-5886 148-5984 153-6252 154-6334 155-6358 167-7059 168-7112 169-7126 169-7133 170-7207 171-7260 171-7270 171-7277 184-7936 185-7989 185-7999 185-8006 186-8048 187-8101 187-8111 187-8118 203-9028 204-9081 204-9091 204-9098 208-9290 209-9343 209-9353 209-9360 210-9407 211-9460 211-9470 211-9477 224-10136 224-10185 225-10199 225-10206 226-10248 226-10297 227-10311 227-10318 246-11400 247-11453 247-11463 247-11470 251-11679 252-11732 252-11742 252-11749 256-11958 257-12011 257-12021 257-12028 258-12065 259-12118 259-12128 259-12135 273-13047 273-13099 273-13109 273-13116 273-13157 273-13209 273-13219 273-13226 285-13920 286-13973 286-13983 287-13994 287-14025 288-14078 288-14088 288-14095 301-14778 302-14831 302-14841 302-14848 303-14886 304-14939 304-14949 304-14956 316-15607 317-15660 318-15674 318-15681 318-15713 319-15766 319-15776 320-15787
ERRTYP	002172	G	#7-94
EVL	= 000004	G	#83-1771
ESEND	= 002100		#6-13
ESLOAD	= 000035		#6-13 6-65
FILLER	041266		#118-4094 120-4178 123-4315
FMT1A	= 006126		#83-1794 199-8804
FMT1B	= 006142		#83-1795 199-8813
FMT10A	= 006564		#83-1806 100-3057
FMT10B	= 006640		#83-1807 168-7074 170-7218 184-7947 186-8059 203-9039 208-9301 210-9418 224-10147 226-10259 246-11411 251-11690 256-11969 258-12076 273-13058 273-13168 285-13931 287-14036 301-14789 303-14897 317-15622 319-15728
FMT11	= 006654		#83-1808 96-2840 96-2867
FMT12A	= 006710		#83-1809 166-6979 182-7848 202-8948 206-9206 222-10048 245-11316 250-11595 255-11874 273-12961 284-13841 300-14699 315-15528
FMT12B	= 006756		#83-1810 166-6986 182-7855 202-8955 206-9213 222-10055 245-11323 250-11602 255-11881 273-12968 284-13848 300-14706 315-15535
FMT13	= 007024		#83-1811 104-3280
FMT14	= 007046		#83-1812 332-16430 332-16436 332-16442
FMT15	= 007060		#83-1813 131-4832 131-4840 131-4851 131-4861 131-4871 131-4879 131-4920 131-4928 131-4936 131-4946 131-4959 131-4967 131-4975 131-4985 131-4993 131-5003 131-5014 131-5022 131-5030 131-5038 131-5046 131-5056 131-5067 131-5075
FMT16	= 005736		#83-1792 338-16749
FMT2	= 006160		#83-1796 94-2734 94-2757 96-2832 96-2859 242-11190 244-11265 249-11544 254-11823 334-16504 334-16512 334-16520 334-16528 334-16536 334-16544 334-16552 335-16564 336-16642 336-16652
FMT3	= 006166		#83-1797 339-16819
FMT4A	= 006216		#83-1798 90-2439 94-2723 333-16467
FMT4B	= 006252		#83-1799 90-2453 97-2911
FMT4C	= 006270		#83-1800 90-2463
FMT5	= 006320		#83-1801 97-2883
FMT6	= 006376		#83-1802 139-5500 144-5796 146-5894 148-5992
FMT7	= 006446		#83-1803 335-16591 336-16619 336-16663
FMT8	= 006466		#83-1804 97-2905
FMT9	= 006534		#83-1805 335-16605 336-16633 337-16681
FSAU	= 000015		#6-13
FSAUTO	= 000020		#6-13
FSBGN	= 000040		#6-13 6-39 8-259 8-302 9-379 9-411
FSCLEA	= 000007		#6-13
FSDU	= 000016		#6-13
FSEND	= 000041		#6-13 6-13 6-13 6-13 6-13 6-13 6-13 6-13 6-13

```
SYMBOL  VALUE    REFERENCES
                  6-13      6-13      6-13      6-13      6-13      6-13      6-13      6-13      6-39
                  8-277     8-339     9-411
F$HARD  = 000004  #6-13     8-259     8-277     8-314     8-316     8-319     8-322     8-324     8-329
F$HW    = 000013  #6-13     7-117     7-134
F$INIT  = 000006  #6-13
F$JMP   = 000050  #6-13
F$MOD   = 000000  #6-13     6-39      9-411
F$MSG   = 000011  #6-13
F$PROT  = 000021  #6-13     9-379     9-385
F$PWR   = 000017  #6-13
F$RPT   = 000012  #6-13
F$SEG   = 000003  #6-13
F$SOFT  = 000005  #6-13     8-302     8-314     8-316     8-319     8-322     8-324     8-329     8-339
F$SRV   = 000010  #6-13
F$SUB   = 000002  #6-13
F$SW    = 000014  #6-13     8-143     8-196
F$TEST  = 000001  #6-13
GEN1      040732  #114-3793 180-7743
GEN2      041432  #120-4173 199-8818
GEN3      041466  #121-4241 164-6873  220-9917
GEN4      041564  #123-4310 242-11156 242-11160
GEN5      041620  #125-4371 273-12832 281-13694 297-14549 313-15397
GETCNT    041106  #116-3959 120-4174  123-4311
GET.WR    047176  #157-6486 165-6903  180-7746  200-8850  205-9158  220-9946  273-12861
G$CNTO  = 000200  #6-13
G$DELM  = 000372  #6-13
G$DISP  = 000003  #6-13
G$EXCP  = 000400  #6-13
G$HIL!  = 000002  #6-13
G$LOLI  = 000001  #6-13
G$NO    = 000000  #6-13
G$OFFS  = 000400  #6-13     8-272     8-273     8-274     8-275     8-313     8-315     8-317     8-318
                    8-320   8-321     8-323     8-325     8-326     8-327     8-328     8-330     8-331
                    8-332   8-333     8-334     8-335     8-336
G$OFSI  = 000376  #6-13     8-272     8-273     8-274     8-275     8-313     8-315     8-317     8-318
                    8-320   8-321     8-323     8-325     8-326     8-327     8-328     8-330     8-331
                    8-332   8-333     8-334     8-335     8-336
G$PRMA  = 000001  #6-13     8-272     8-273
G$PRMD  = 000002  #6-13     8-274     8-275     8-317     8-318     8-320     8-323     8-330
G$PRML  = 000000  #6-13     8-313     8-315     8-321     8-325     8-326     8-327     8-328     8-331
                    8-332   8-333     8-334     8-335     8-336
G$RADA  = 000140  #6-13
G$RADB  = 000000  #6-13
G$RADD  = 000040  #6-13     8-320     8-323     8-330
G$RADL  = 000120  #6-13     8-313     8-315     8-321     8-325     8-326     8-327     8-328     8-331
                    8-332   8-333     8-334     8-335     8-336
G$RADO  = 000020  #6-13     8-272     8-273     8-274     8-275     8-317     8-318
G$XFER  = 000004  #6-13     8-314     8-316     8-319     8-322     8-324     8-329
G$YES   = 000010  #6-13     8-272     8-273     8-274     8-275     8-313     8-315     8-317     8-318
                    8-320   8-321     8-323     8-325     8-326     8-327     8-328     8-330     8-331
                    8-332   8-333     8-334     8-335     8-336
HARDS     033314  #82-1660  *84-2092  *103-3214 335-16584 336-16629
```

```
SYMBOL  VALUE          REFERENCES
HELP    = 000000       #6-4     6-8      6-30     6-48     6-67     7-103    7-119    8-145    #8-200
                       8-261    8-279    8-304    9-341    9-387    9-404    9-416
HOE     = 100000 G     #83-1784
IBE     = 010000 G     #83-1781
IDU     = 000040 G     #83-1774
IER     = 020000 G     #83-1782
INIT.A    035206       #88-2286 111-3664
INTEGR    047232       #164-6862 325-16052
ISOLAT    037332       #100-3048 167-7046 167-7058 170-7195 170-7206 183-7918 184-7935 186-8047 203-9015
                       203-9027 208-9277 208-9289 210-9406 223-10118 224-10135 226-10247 246-11386 246-11399
                       251-11665 251-11678 256-11944 256-11957 258-12064 273-13030 273-13046 273-13156 285-13907
                       285-13919 287-14024 301-14765 301-14777 303-14885 316-15594 316-15606 318-15712
ISR     = 000100 G     #83-1775
IXE     = 004000 G     #83-1780
I$AU    = 000041       #6-13
I$AUTO  = 000041       #6-13
I$CLN   = 000041       #6-13
I$DU    = 000041       #6-13
I$HRD   = 000041       #8-259   #8-277
I$INIT  = 000041       #6-13
I$MOD   = 000041       #6-13    6-39     #6-39    9-411    #9-411
I$MSG   = 000041       #6-13
I$PROT  = 000040       #6-13    #9-379
I$PTAB  = 000041       #6-13
I$PWR   = 000041       #6-13
I$RPT   = 000041       #6-13
I$SEG   = 000041       #6-13
I$SETU  = 000041       #6-13
I$SFT   = 000041       #8-302   #8-339
I$SRV   = 000041       #6-13
I$SUB   = 000041       #6-13
I$TST   = 000041       #6-13
I.AM.D    034336       #82-1680 *83-1927 *138-5465 138-5471 138-5477
J$JMP   = 000167       #6-13
LAU       005650       #48-296  49-315
LAUTO     005504       #45-101  45-119
LCLEAN    104724       #339-16805 340-16848
LDU       005516       #47-210  47-257
LIMIT     002222 G     #8-153   83-1722  95-2794
LINIT     040474       #111-3636 112-3710
LOE     = 040000 G     #83-1783
LOT     = 000010 G     #83-1772
LOW.SE    034460 G     #82-1702 *85-2122 *96-2850 97-2881 164-6894 180-7738 185-8033 200-8845 205-9153
                       210-9396 220-9941 225-10233 243-11200 258-12054 263-12285 273-12856 273-13143 282-13717
                       287-14014 297-14573 303-14875 318-15702
LRPT      005466       #44-39   44-58
LSECT     002226 G     #8-157   83-1723  *95-2806 *95-2814 96-2848 96-2850 96-2865
LSACP     002110 G     #6-65
LSAPT     002036 G     #6-65
LSAU      005652 G     6-65     #49-315
LSAUT     002070 G     #6-65
LSAUTO    005506 G     6-65     #45-119
```

SYMBOL	VALUE		REFERENCES
L$CCP	002106	G	#6-65
L$CLEA	105034	G	6-65 #340-16847
L$CO	002032	G	#6-65
L$DEPO	002011	G	#6-65
L$DESC	002130	G	6-65 #7-85
L$DESP	002076	G	#6-65
L$DEVP	002060	G	#6-65
L$DISP	002204	G	6-65 #7-101
L$DLY	002116	G	#6-65 137-5400 137-5428 138-5442
L$DTP	002040	G	#6-65
L$DTYP	002034	G	#6-65
L$DU	005640	G	6-65 #47-257
L$DUT	002072	G	#6-65
L$DVTY	002122	G	6-65 #6-80
L$EF	002052	G	#6-65
L$ENVI	002044	G	#6-65
L$ERRT	002172	G	6-65 #7-94
L$ETP	002102	G	#6-65
L$EXP1	002046	G	#6-65
L$EXP4	002064	G	#6-65
L$EXP5	002066	G	#6-65
L$HARD	002266	G	6-65 8-259 #8-259
L$HIME	002120	G	#6-65
L$HPCP	002016	G	#6-65
L$HPTP	002022	G	#6-65
L$HW	002210	G	6-65 7-117 #7-117
L$ICP	002104	G	#6-65
L$INIT	040722	G	6-65 #112-3710
L$LADP	002026	G	#6-65
L$LAST	= 105050	G	6-65 #340-16873
L$LOAD	002100	G	#6-65
L$LUN	002074	G	#6-65 46-201 47-210 47-216 47-221 47-227 83-1722 *85-2119 *111-3648 *164-6891 *180-7733 *200-8840 *205-9148 *220-9936 *242-11183 *273-12851 *282-13714 *297-14565 *313-15405
L$MREV	002050	G	#6-65
L$NAME	002000	G	#6-65
L$PRIO	002042	G	#6-65
L$PROT	004126	G	6-65 #9-379
L$PRT	002112	G	#6-65
L$REPP	002062	G	#6-65
L$REV	002010	G	#6-65
L$RPT	005474	G	6-65 #44-58
L$SOFT	002450	G	6-65 8-302 #8-302
L$SPC	002056	G	#6-65
L$SPCP	002020	G	#6-65
L$SPTP	002024	G	#6-65
L$STA	002030	G	#6-65
L$SW	002222	G	6-65 8-143 #8-143
L$TEST	002114	G	#6-65
L$TIML	002014	G	#6-65
L$UNIT	002012	G	#6-65 83-1722 84-2079 111-3645 164-6874 180-7715 199-8819 205-9130 220-9918 242-11165 265-12372 265-12379 265-12382 265-12403 273-12833 281-13691 297-14546 322-15902

SYMBOL	VALUE		REFERENCES								
			333-16450								
L10000	002220		7-117	#7-134							
L10001	002264		8-143	#8-196							
L10002	002332		8-259	#8-277							
L10003	002664		8-302	#8-339							
MARPAT	002246	G	#8-165	83-1724	242-11154	242-11157					
MLB10	= 007472		#83-1853	131-4992							
MLB11	= 007500		#83-1854	131-5002							
MLB12	= 007504		#83-1855	131-5013							
MLB13	= 007510		#83-1856	131-5021							
MLB14	= 007514		#83-1857	131-5029							
MLB15	= 007520		#83-1858	131-5037							
MLB16	= 007524		#83-1859	131-5045							
MLB17	= 007530		#83-1860	131-5055							
MLB18	= 007534		#83-1861	131-5066							
MLB19	= 007542		#83-1862	131-5074							
MLB2	= 007430		#83-1845	131-4919							
MLB20	= 007550		#83-1863	131-4831							
MLB21	= 007554		#83-1864	131-4870							
MLB22	= 007560		#83-1865	131-4839							
MLB23	= 007564		#83-1866	131-4850							
MLB24	= 007570		#83-1867	131-4878							
MLB3	= 007434		#83-1846	131-4927							
MLB4	= 007440		#83-1847	131-4935							
MLB5	= 007444		#83-1848	131-4945							
MLB6	= 007450		#83-1849	131-4860	131-4958						
MLB7	= 007454		#83-1850	131-4966							
MLB8	= 007460		#83-1851	131-4974							
MLB9	- 007464		#83-1852	131-4984							
ML.REG	034346	G	#82-1689	*89-2340	90-2411	90-2421	90-2424	90-2430	90-2451	94-2714	94-2715
			94-2716	94-2745	95-2770	95-2780	95-2798	97-2887	98-2972	99-2997	100-3048
			100-3054	131-4732	131-4739	131-4740	131-4742	131-4745	131-4747	131-4749	131-4751
			131-4753	131-4756	131-4758	131-4761	131-4763	131-4765	131-4767	131-4770	131-4775
			131-4777	131-4779	131-4781	131-4783	131-4785	131-4791	131-4793	131-4795	131-4798
			131-4800	131-4802	131-4805	131-4810	131-4812	131-4829	131-4837	131-4848	131-4856
			131-4868	131-4876	131-4917	131-4925	131-4933	131-4943	131-4951	131-4964	131-4972
			131-4980	131-4982	131-4990	131-5000	131-5011	131-5019	131-5027	131-5035	131-5043
			131-5053	131-5061	131-5072	137-5416	137-5417	137-5422	137-5425	138-5439	138-5452
			138-5453	138-5454	138-5457	138-5460	138-5461	138-5462	138-5463	138-5464	138-5469
			138-5474	167-7061	168-7079	168-7080	168-7099	168-7102	170-7209	170-7223	170-7224
			171-7247	171-7250	184-7938	184-7952	184-7953	184-7972	184-7975	186-8050	186-8064
			186-8065	186-8084	186-8087	203-9030	203-9044	203-9045	204-9068	204-9071	208-9292
			208-9306	208-9307	208-9326	209-9333	210-9409	210-9423	210-9424	211-9447	211-9450
			224-10138	224-10152	224-10153	224-10172	224-10175	226-10250	226-10264	226-10265	226-10284
			226-10287	246-11402	247-11420	247-11421	247-11440	247-11443	251-11681	251-11695	252-11700
			252-11719	252-11722	256-11960	256-11974	256-11975	257-11998	257-12001	258-12067	258-12081
			258-12082	259-12105	259-12108	273-13049	273-13063	273-13064	273-13083	273-13086	273-13159
			273-13173	273-13174	273-13193	273-13196	285-13922	286-13940	286-13941	286-13960	286-13963
			287-14027	287-14041	287-14042	288-14065	288-14068	301-14780	301-14794	301-14795	302-14818
			302-14821	303-14888	303-14902	303-14903	304-14926	304-14929	317-15613	317-15627	317-15628
			317-15647	317-15650	318-15715	319-15733	319-15734	319-15753	319-15756	322-15924	322-15925
			322-15926	333-16458	333-16459	333-16460	336-16637	336-16647			

```
SYMBOL  VALUE          REFERENCES
MSG0    = 011002       #83-1913   139-5504
MSG1    = 011070       #83-1914   165-6926   165-6936   165-6945   166-6993   167-7019   167-7029   167-7038   169-7164
                        169-7174   170-7187   181-7776   181-7787   181-7796   182-7863   183-7889   183-7900   183-7911
                        200-8875   201-8889   201-8898   202-8962   202-8984   203-8998   203-9007   207-9224   207-9246
                        207-9256   207-9265   221-9976   221-9987   221-9996   222-10063  223-10089  223-10100  223-10109
                        243-11229  243-11240  244-11253  245-11331  245-11353  246-11368  246-11377  248-11506  248-11517
                        248-11526  250-11610  250-11632  251-11647  251-11656  253-11785  253-11[illegible]  253-11805  255-11889
                        255-11911  256-11926  256-11935  273-12890  273-12901  273-12910  273-129[illegible]  273-13001  273-13012
                        273-13021  283-13771  283-13781  283-13790  284-13855  284-13876  285-13890  285-13899  298-14625
                        298-14635  299-14648  300-14713  300-14734  300-14744  301-14757  314-15458  314-15468  314-15477
                        315-15542  316-15567  316-15577  316-15586
MSG2    = 011120       #83-1915   104-3278   167-7049   170-7198   183-7921   203-9018   208-9280   223-10121  246-11389
                        251-11668  256-11947  273-13033  285-13910  301-14768  316-15597  336-16641  336-16651
MSG3    = 011166       #83-1916   169-7130   169-7137   171-7274   171-7281   185-8003   185-8010   187-8115   187-8122
                        204-9095   204-9102   209-9357   209-9364   211-9474   211-9481   225-10203  225-10210  227-10315
                        227-10322  247-11467  248-11478  252-11746  253-11757  257-12025  258-12036  259-12132  259-12139
                        273-13113  273-13120  273-13223  273-13230  286-13987  287-13998  288-14092  288-14099  302-14845
                        302-14852  304-14953  304-14960  318-15678  318-15685  320-15784  320-15791
MSG4    = 011202       #83-1917   168-7116   171-7264   185-7993   187-8105   204-9085   209-9347   211-9464   225-10193
                        227-10305  247-11457  252-11736  257-12015  259-12122  273-13103  273-13213  286-13977  288-14082
                        302-14835  304-14943  317-15664  319-15770
MSG5    = 011216       #83-1918   166-6972   182-7841   202-8941   206-9199   222-10041  245-11309  250-11588  254-11863
                        273-12954  284-13834  299-14688  315-15521
NUM.DR    034456  G    46-202     *47-232    #82-1700   *84-2102   111-3661   *111-3666  313-15398  322-15901
ONEFIL  = 000001       #2-5       4-1665     5-1666     6-34       8-223      9-429
ONLY      002232  G    #8-159     83-1723    95-2800    95-2810
OPT1      051460       #179-7702  325-16056
OPT2      054054       #199-8781  325-16059
OPT3      060052       #219-9903  325-16062
OPT4      062444       #242-11146 325-16067
OPT5      [illegible]  #322-15893 325-16070
O$APTS  = 000001       #6-13      #6-46      6-65
O$AU    = 000001       #6-13      #6-46      6-65
O$BGNR  = 000001       #6-13      #6-46      6-65
O$BGNS  = 000001       #6-13      #6-46      6-65
O$DU    = 000001       #6-13      #6-46      6-65
O$ERRT  = 000001       #6-13      #6-46      6-65
O$GNSW  = 000001       #6-13      #6-46      6-65
O$POIN  = 000001       #6-13      #6-46      6-46       6-65
O$SETU  = 000001       #6-13      #6-46      6-65
PATTBL    034520  G    #82-1706   116-3993   116-3994
PATTER    032676       #81-1643   *115-3870  115-3871   *115-3873  120-4173   *199-8789  199-8800   199-8802   199-8809
                        199-8817
PHR1    = 007714       #83-1881   139-5495   148-5990   153-6268
PHR10   = 010120       #83-1890   131-4822   131-4895   131-4909
PHR11   = 010140       #83-1891   131-5104
PHR12   = 010202       #83-1892   332-16423
PHR13   = 010240       #83-1893   332-16440
PHR14   = 010266       #83-1894   96-2839    96-2864
PHR15   = 010304       #83-1895   96-2866
PHR16   = 010322       #83-1896   96-2837
PHR17   = 010342       #83-1897   332-16414
```

SYMBOL	VALUE		REFERENCES							
PHR18	= 010366		#83-1898	336-16618						
PHR19	= 010410		#83-1899	332-16428						
PHR2	= 007730		#83-1882	325-16046	339-16809					
PHR20	= 010430		#83-1900	332-16434						
PHR21	= 005662		#83-1791	337-16700						
PHR3	= 007746		#83-1883	89-2324						
PHR4	= 007770		#83-1884	97-2910						
PHR5	= 010004		#83-1885	335-16590						
PHR6	= 010026		#83-1886	336-16662						
PHR7	= 010050		#83-1887	90-2438	94-2722	333-16466				
PHR8	= 010066		#83-1888	90-2452	90-2462					
PHR9	= 010100		#83-1889	199-8803	199-8812					
PM.SBE	012264		#81-1633	*86-2176	*104-3273	104-3274	336-16639	336-16649		
PNT	= 001000	G	#83-1778							
PRI	= 002000	G	#83-1779							
PRI00	= 000000	G	#83-1770							
PRI01	= 000040	G	#83-1769							
PRI02	= 000100	G	#83-1768							
PRI03	= 000140	G	#83-1764							
PRI04	= 000200	G	#83-1763							
PRI05	= 000240	G	#83-1762							
PRI06	= 000300	G	#83-1761							
PRI07	= 000340	G	#83-1760							
PTABLE	034422	G	#82-1690	90-2434	94-2711	94-2718	*111-3660	137-5413	138-5449	322-15921
			333-16462							333-16455
P.AAA	005662		#70-944	83-1791						
P.AAB	005736		#70-959	83-1792						
P.AAC	006042		#70-982	83-1793						
P.AAD	006126		#70-1003	83-1794						
P.AAE	006142		#70-1007	83-1795						
P.AAF	006160		#70-1012	83-1796						
P.AAG	006166		#70-1014	83-1797						
P.AAH	006216		#70-1022	83-1798						
P.AAI	006252		#70-1032	83-1799						
P.AAJ	006270		#70-1037	83-1800						
P.AAK	006320		#71-1049	83-1801						
P.AAL	006376		#71-1065	83-1802						
P.AAM	006446		#71-1079	83-1803						
P.AAN	006466		#71-1085	83-1804						
P.AAO	006534		#72-1102	83-1805						
P.AAP	006564		#72-1110	83-1806						
P.AAQ	006640		#72-1125	83-1807						
P.AAR	006654		#72-1129	83-1808						
P.AAS	006710		#72-1139	83-1809						
P.AAT	006756		#73-1156	83-1810						
P.AAU	007024		#73-1169	83-1811						
P.AAV	007046		#73-1175	83-1812						
P.AAW	007060		#73-1179	83-1813						
P.AAX	007066		#73-1181	83-1814						
P.AAY	007072		#73-1183	83-1815						
P.AAZ	007100		#73-1185	83-1816						
P.ABA	007112		#73-1189	83-1817						

SYMBOL	VALUE	REFERENCES	
P.ABB	007130	#73-1194	83-1818
P.ABC	007152	#73-1200	83-1819
P.ABD	007200	#74-1212	83-1823
P.ABE	007206	#74-1214	83-1824
P.ABF	007212	#74-1216	83-1825
P.ABG	007220	#74-1218	83-1826
P.ABH	007230	#74-1221	83-1827
P.ABI	007240	#74-1224	83-1828
P.ABJ	007246	#74-1226	83-1829
P.ABK	007256	#74-1229	83-1830
P.ABL	007264	#74-1231	83-1831
P.ABM	007272	#74-1233	83-1832
P.ABN	007300	#74-1235	83-1833
P.ABO	007312	#74-1239	83-1834
P.ABP	007322	#74-1242	83-1835
P.ABQ	007332	#74-1245	83-1836
P.ABR	007336	#74-1247	83-1837
P.ABS	007344	#74-1249	83-1838
P.ABT	007354	#74-1252	83-1839
P.ABU	007370	#74-1256	83-1840
P.ABV	007400	#74-1259	83-1841
P.ABW	007412	#75-1267	83-1842
P.ABX	007416	#75-1269	83-1843
P.ABY	007424	#75-1271	83-1844
P.ABZ	007430	#75-1273	83-1845
P.ACA	007434	#75-1275	83-1846
P.ACB	007440	#75-1277	83-1847
P.ACC	007444	#75-1279	83-1848
P.ACD	007450	#75-1281	83-1849
P.ACE	007454	#75-1283	83-1850
P.ACF	007460	#75-1285	83-1851
P.ACG	007464	#75-1287	83-1852
P.ACH	007472	#75-1289	83-1853
P.ACI	007500	#75-1291	83-1854
P.ACJ	007504	#75-1293	83-1855
P.ACK	007510	#75-1295	83-1856
P.ACL	007514	#75-1297	83-1857
P.ACM	007520	#75-1299	83-1858
P.ACN	007524	#75-1301	83-1859
P.ACO	007530	#75-1303	83-1860
P.ACP	007534	#75-1305	83-1861
P.ACQ	007542	#75-1307	83-1862
P.ACR	007550	#75-1309	83-1863
P.ACS	007554	#75-1311	83-1864
P.ACT	007560	#75-1313	83-1865
P.ACU	007564	#75-1315	83-1866
P.ACV	007570	#75-1317	83-1867
P.ACW	007574	#76-1323	83-1868
P.ACX	007626	#76-1332	83-1869
P.ACY	007634	#76-1334	83-1870
P.ACZ	007642	#76-1336	83-1871
P.ADA	007650	#76-1338	83-1872

SYMBOL	VALUE	REFERENCES		
P.ADB	007656	#76-1340	83-1873	
P.ADC	007664	#76-1342	83-1874	
P.ADD	007672	#76-1344	83-1878	
P.ADE	007700	#76-1346	83-1879	
P.ADF	007706	#76-1348	83-1880	
P.ADG	007714	#76-1350	83-1881	
P.ADH	007730	#76-1354	83-1882	
P.ADI	007746	#76-1359	83-1883	
P.ADJ	007770	#76-1365	83-1884	
P.ADK	010004	#76-1369	83-1885	
P.ADL	010026	#77-1379	83-1886	
P.ADM	010050	#77-1385	83-1887	
P.ADN	010066	#77-1390	83-1888	
P.ADO	010100	#77-1394	83-1889	
P.ADP	010120	#77-1400	83-1890	
P.ADQ	010140	#77-1406	83-1891	
P.ADR	010202	#77-1418	83-1892	
P.ADS	010240	#77-1428	83-1893	
P.ADT	010266	#78-1440	83-1894	
P.ADU	010304	#78-1445	83-1895	
P.ADV	010322	#78-1450	83-1896	
P.ADW	010342	#78-1456	83-1897	
P.ADX	010366	#78-1463	83-1898	
P.ADY	010410	#78-1469	83-1899	
P.ADZ	010430	#78-1475	83-1900	
P.AEA	010444	#78-1479	83-1901	
P.AEB	010446	#78-1480	83-1902	
P.AEC	010450	#78-1481	83-1903	
P.AED	010454	#78-1483	83-1904	
P.AEE	010460	#78-1485	83-1905	
P.AEF	010502	#79-1495	83-1906	
P.AEG	010526	#79-1502	83-1907	
P.AEH	010602	#79-1517	83-1908	
P.AEI	010656	#79-1532	83-1909	
P.AEJ	010710	#79-1541	83-1910	
P.AEK	010734	#80-1552	83-1911	
P.AEL	010756	#80-1558	83-1912	
P.AEM	011002	#80-1565	83-1913	
P.AEN	011070	#80-1583	83-1914	
P.AEO	011120	#80-1591	83-1915	
P.AEP	011166	#81-1608	83-1916	
P.AEQ	011202	#81-1612	83-1917	
P.AER	011216	#81-1616	83-1918	
QH1	002332	8-272	#8-286	
QH2	002351	8-273	#8-287	
QH3	002374	8-274	#8-288	
QH4	002425	8-275	#8-289	
QS1	002664	8-313	#9-347	
QS10	003534	8-328	#9-362	
QS11	003554	8-323	8-330	#9-363
QS12	003637	8-331	#9-364	
QS13	003657	8-332	#9-365	

SYMBOL	VALUE		REFERENCES								
QS14	003724		8-333	#9-366							
QS15	003763		8-334	#9-367							
QS16	004030		8-335	#9-368							
QS17	004066		8-336	#9-369							
QS2	002732		8-315	#9-348							
QS3	003007		8-317	#9-349							
QS4	003047		8-318	#9-350							
QS5	003107		8-320	#9-351							
QS6	003147		8-321	#9-352							
QS7	003454		8-325	#9-359							
QS8	003474		8-326	#9-360							
QS9	003514		8-327	#9-361							
QUICK	032674		#81-1642	*111-3643	*112-3691	199-8790	205-9127	325-16039	325-16063		
RANDOM	005464	G	*42-27	#42-37	83-1726	125-4379	149-6053	149-6055	*261-12203	261-12204	*263-12292
			263-12295	*265-12377	265-12378	*265-12381	265-12385	265-12391	265-12399	265-12401	*265-12405
RAND1	070540		#273-12823	323-15938							
RAND2	073124		#281-13682	323-15940							
RAND3	075444		#297-14537	323-15942							
RAND4	100006		#312-15384	323-15949							
RANGE	002224	G	#8-155	83-1722	95-2796						
RBUFF	022670		#81-1639	83-1787	83-1788	83-1790	166-6958	166-6964	166-7005	168-7093	180-7735
			185-8037	200-8842	205-9150	210-9400	220-9938	225-10237	258-12058	273-12853	273-13150
			282-13719	287-14018	297-14567	303-14879	313-15407	318-15706			
RCBUFF	= 023470		#83-1788	242-11164							
RDBUFF	= 022670		#83-1787	242-11163							
RD.COU	034322		#82-1668	*85-2111	141-5627	*141-5628	*141-5632				
RD.MIL	034326		#82-1672	*85-2113	*141-5636	*141-5638	332-16435				
RD.THO	034324		#82-1670	*85-2112	*141-5631	141-5633	*141-5635				
READ	045612		#145-5863	149-6057	153-6275	154-6315	154-6327	166-6960	166-7002	168-7090	181-7807
			181-7815	185-8039	201-8908	201-8916	206-9170	206-9178	210-9402	221-10007	221-10015
			225-10239	244-11275	244-11283	249-11554	249-11562	254-11833	254-11841	258-12060	273-12921
			273-12932	273-13152	283-13800	283-13808	287-14020	299-14658	299-14666	303-14881	314-15487
			314-15495	318-15708							
REFRES	002252	G	#8-167	83-1725	89-2318	138-5455					
RETRY	046224		#153-6245	165-6919	166-7007	168-7095	169-7157	171-7243	181-7768	183-7881	184-7968
			186-8080	200-8868	202-8976	204-9064	207-9238	208-9322	211-9443	221-9968	223-10081
			224-10168	226-10280	243-11221	245-11345	247-11436	248-11498	250-11624	252-11715	253-11777
			255-11903	257-11994	259-12101	273-12882	273-12993	273-13079	273-13189	282-13760	284-13869
			286-13956	288-14061	298-14618	300-14727	302-14814	304-14922	314-15451	316-15560	317-15643
			319-15749								
RETRY:	034340		#82-1682	*84-2103	131-4815	131-5102	138-5479	144-5779	144-5790	146-5877	146-5888
			148-5975	148-5986	*153-6246	*155-6385					
RE2	005362		41-1349	#41-1379							
RE3	005360		41-1357	#41-1378							
RE4	005356		41-1366	#41-1377							
RN	005370	G	#42-12	83-1726	125-4377	149-6052	261-12202	263-12291	265-12376		
RNDSEC	070240		#263-12282	298-14589	313-15418						
RNDU	070336		#264-12367	313-15403							
RNDWC	070204		#261-12202	282-13724	298-14584	313-15413					
RPTR	032672		#81-1641	*156-6444	*180-7735	181-7810	181-7813	182-7833	*200-8842	201-8911	201-8914
			201-8929	*205-9150	206-9173	206-9176	206-9191	*220-9938	221-10010	221-10013	222-10033
			*273-12853	273-12924	273-12930	273-12946	*282-13719	283-13803	283-13806	284-13826	*297-14567

```
                                                                         C 14
CZMLBB     CREATED BY  MACRO  ON 29-MAR-82 AT 13:44        PAGE 14
SYMBOL CROSS REFERENCE                                     CREF
SYMBOL   VALUE               REFERENCES                                                                                         SEQ 0377
                              299-14661   299-14664   299-14680  *313-15407   314-15490   314-15493   315-15513
RTN0     = 007574            #83-1868     164-6865
RTN1     = 007626            #83-1869     180-7708
RTN2     = 007634            #83-1870     199-8783
RTN3     = 007642            #83-1871     219-9905
RTN4     = 007650            #83-1872     242-11148
RTN5     = 007656            #83-1873     322-15894
RTN5A    = 007664            #83-1874     273-12825
RTN5B    = 007672            #83-1878     281-13684
RTN5C    = 007700            #83-1879     297-14539
RTN5D    = 007706            #83-1880     313-15390
SAYWHO     035524            #90-2410     94-2744     104-3277    131-4821    131-4894    131-4908    138-5482    166-6971    182-7840
                              202-8940    206-9198    222-10040   245-11308   249-11583   254-11862   273-12953   284-13833   299-14687
                              315-15520   333-16474
SAY1     = 007072            #83-1815     131-4823    131-4896    131-4910    131-5105    139-5505    166-6973    182-7842    202-8942
                              206-9200    222-10042   242-11185   245-11310   250-11589   255-11868   273-12826   273-12955   281-13685
                              284-13835   297-14540   299-14689   313-15391   315-15522   332-16415   332-16424   337-16701   337-16710
SAY2     = 007100            #83-1816     94-2729     94-2752     96-2827     96-2854     154-6338    155-6362    164-6867    180-7710
                              199-8785    220-9911    242-11150   322-15896   333-16491
SAY3     = 007112            #83-1817     89-2326     153-6260    153-6270    153-6280    325-16048   339-16811
SAY4     = 007130            #83-1818
SAY5     = 007152            #83-1819     89-2314
SBE$HE   = 006042            #83-1793     337-16709
SBE$.C     012664            #81-1635    *84-2100     103-3215    104-3266    104-3285   *105-3320    108-3463   *109-3548    337-16694
                              337-16714
SBE.LO     011264            #81-1632    *85-2157    *85-2158     103-3224   *104-3262    104-3288   *105-3319    108-3472   *109-3510
                              109-3520   *109-3547    337-16720   337-16721
SEED1      005456   G        *42-15       42-18      #42-33       83-1726     125-4372
SEED2      005460   G         42-13      *42-20       42-27      #42-34       83-1726     125-4373
SEED3      005462   G         42- 9       42-23      *42-26      #42-35       83-1726     125-4374
SELPAT     041066            #115-3870    199-8799
SERVIC     034544   G        #83-1926     89-2348
SET.PT     047132            #156-6435    180-7749    200-8853    206-9165    220-9949    273-12864   282-13727   298-14587   313-15416
SFPTBL     002222   G        #8-143
SOFTS      032714            #82-1659    *84-2091    *108-3462    335-16583   335-16601
START.     044424            #137-5407    144-5777    146-5875    148-5973
SVCGBL   = 000001            #6-13       #6-22        6-65        6-65        6-65        6-65        6-65        6-65        6-65
                              6-65        6-65        6-65        6-65        6-65        6-65        6-65        6-65        6-65
                              6-65        6-65        6-65        6-65        6-65        6-65        6-65        6-65        6-65
                              6-65        6-65        6-65        6-65        6-65        6-65        6-65        6-65        6-65
                              6-65        6-65        6-65        6-65        6-80        7-85        7-94        7-101       7-117
                              7-117       8-143       8-143       8-259       8-302       9-379
SVCINS   = 000001            #6-13       #6-19        6-65        6-65        6-65        6-65        6-65        6-65        6-65
                              6-65        6-65        6-65        6-65        6-65        6-65        6-65        6-65        6-65
                              6-65        6-65        6-65        6-65        6-65        6-65        6-65        6-65        6-65
                              6-65        6-65        6-65        6-65        6-65        6-65        6-65        6-65        6-65
                              6-65        6-65        6-65        6-65        6-65        6-65        6-65        6-65        6-65
                              6-65        6-65        6-65        6-65        6-65        6-65        6-65        6-65        6-65
                              6-65        6-65        6-65        6-65        6-65        6-65        6-65        6-65        6-65
                              6-65        6-65        6-65        6-65        6-65        6-65        6-65        6-65        6-65
                              6-65        6-65        6-65        6-65        6-65        6-65        6-65        6-65        6-65
                              6-65        6-65        6-65        6-65        6-65        6-65        6-65        6-65        6-65
```

```
SYMBOL  VALUE                  REFERENCES
                               6-65       6-65       6-65       6-65       6-65       6-65       6-65       6-65       6-65
                               6-65       6-65       6-65       6-65       6-65       6-65       6-65       6-65       6-65
                               6-65       6-65       6-65       6-65       6-65       6-65       6-65       6-65       6-65
                               6-65       6-65       6-65       6-65       6-65       6-65       6-65       6-65       6-65
                               6-65       6-65       6-65       6-65       6-65       6-65       6-65       6-65       6-65
                               6-65       6-65       6-65       6-65       6-65       6-65       6-65       6-65       6-80
                               6-80       6-80       6-80       6-80       6-80       7-85       7-85       7-85       7-85
                               7-85       7-85       7-101      7-101      7-101      7-101      7-101      7-101      7-117
                               7-117      7-117      8-143      8-143      8-143      8-259      8-259      8-259      8-272
                               8-272      8-272      8-272      8-272      8-272      8-272      8-272      8-272      8-272
                               8-272      8-272      8-273      8-273      8-273      8-273      8-273      8-273      8-273
                               8-273      8-273      8-273      8-273      8-273      8-274      8-274      8-274      8-274
                               8-274      8-274      8-274      8-274      8-274      8-274      8-274      8-274      8-274
                               8-274      8-274      8-275      8-275      8-275      8-275      8-275      8-275      8-275
                               8-275      8-275      8-275      8-275      8-275      8-275      8-275      8-275      8-277
                               8-277      8-277      8-302      8-302      8-302      8-313      8-313      8-313      8-313
                               8-313      8-313      8-313      8-313      8-313      8-314      8-314      8-314      8-315
                               8-315      8-315      8-315      8-315      8-315      8-315      8-315      8-315      8-316
                               8-316      8-316      8-317      8-317      8-317      8-317      8-317      8-317      8-317
                               8-317      8-317      8-317      8-317      8-317      8-317      8-317      8-317      8-318
                               8-318      8-318      8-318      8-318      8-318      8-318      8-318      8-318      8-318
                               8-318      8-318      8-318      8-318      8-318      8-319      8-319      8-319      8-320
                               8-320      8-320      8-320      8-320      8-320      8-320      8-320      8-320      8-320
                               8-320      8-320      8-320      8-320      8-320      8-321      8-321      8-321      8-321
                               8-321      8-321      8-321      8-321      8-321      8-322      8-322      8-322      8-323
                               8-323      8-323      8-323      8-323      8-323      8-323      8-323      8-323      8-323
                               8-323      8-323      8-323      8-323      8-323      8-324      8-324      8-324      8-325
                               8-325      8-325      8-325      8-325      8-325      8-325      8-325      8-325      8-326
                               8-326      8-326      8-326      8-326      8-326      8-326      8-326      8-326      8-327
                               8-327      8-327      8-327      8-327      8-327      8-327      8-327      8-327      8-328
                               8-328      8-328      8-328      8-328      8-328      8-328      8-328      8-328      8-329
                               8-329      8-329      8-330      8-330      8-330      8-330      8-330      8-330      8-330
                               8-330      8-330      8-330      8-330      8-330      8-330      8-330      8-330      8-331
                               8-331      8-331      8-331      8-331      8-331      8-331      8-331      8-331      8-332
                               8-332      8-332      8-332      8-332      8-332      8-332      8-332      8-332      8-333
                               8-333      8-333      8-333      8-333      8-333      8-333      8-333      8-333      8-334
                               8-334      8-334      8-334      8-334      8-334      8-334      8-334      8-334      8-335
                               8-335      8-335      8-335      8-335      8-335      8-335      8-335      8-335      8-336
                               8-336      8-336      8-336      8-336      8-336      8-336      8-336      8-336      8-339
                               8-339      8-339
SVCSUB  = 000001               #6-13      #6-21
SVCTAG  = 000001               #6-13      #6-23      7-134      8-196      8-277      8-339
SVCTST  = 000001               #6-13      #6-20
SYSERR    041710               #131-4722   139-5511
S$LSYM  = 010000               #6-13      #7-134     #8-196     #8-277     #8-339
TOP.SE    034500   G           #82-1704    95-2777    165-6902   180-7739   180-7745   185-8032   200-8846   200-8849   205-9154
                                205-9157   210-9395   220-9942   220-9945   225-10232  243-11202  258-12053  263-12287  263-12293
                                273-12857  273-12860  273-13142  282-13720  287-14013  297-14570  303-14874  313-15410
TRIES     033714               #82-1661   *84-2093   *155-6391   335-16585  337-16677
TRT00   = 010444               #83-1901    97-2891
TRT01   = 010446               #83-1902    97-2895
TRT10   = 010450               #83-1903    97-2899
```

CZMLBB CREATED BY MACRO ON 29-MAR-82 AT 13:44 PAGE 16
SYMBOL CROSS REFERENCE CREF SEQ 0379

SYMBOL VALUE REFERENCES

TRT11 = 010454 #83-1904 97-2903
TSECT 002230 G #8-158 83-1723 *95-2807 *95-2808 *95-2815 *95-2816 96-2821 96-2823 96-2838
96-2848 96-2863
TSARGC = 000002 #6-65 6-65 #6-65 6-65 6-65 #6-65 6-65 6-65 #6-65
6-65 6-65 #6-65 6-65 6-65 #6-65 6-65 6-65
TSCODE = 020130 #8-272 8-272 #8-272 8-272 #8-272 8-272 #8-273 8-273 #8-273
8-273 #8-273 8-273 #8-274 #8-274 8-274 #8-274 8-274
#8-275 8-275 #8-275 8-275 #8-275 8-275 #8-313 #8-313 #8-313
8-313 #8-313 8-313 #9-314 8-314 8-314 #8-314 8-314
#8-314 8-314 #8-314 8-314 #8-315 8-315 #8-315 #8-315
8-315 #8-316 8-316 8-316 #8-316 8-316 #8-316 8-316
#8-316 8-316 #8-317 8-317 #8-317 8-317 #8-317 #8-318
8-318 #8-318 8-318 #8-318 8-318 #8-319 8-319 #8-319
8-319 #8-319 8-319 #8-319 8-319 #8-320 8-320 #8-320
8-320 #8-320 8-320 #8-321 8-321 #8-321 8-321 #8-321
#8-322 8-322 8-322 #8-322 8-322 #8-322 8-322 #8-322
8-322 #8-323 8-323 #8-323 8-323 #8-323 8-323 #8-324
8-324 #8-324 8-324 8-324 #8-324 8-324 #8-324 8-324
8-325 #8-325 8-325 #8-325 8-325 #8-326 8-326 #8-326
#8-326 8-326 #8-327 8-327 #8-327 8-327 #8-327 8-327
8-328 #8-328 8-328 #8-328 8-328 #8-329 8-329 #8-329
8-329 8-329 #8-329 8-329 #8-329 8-329 #8-330 8-330
8-330 #8-330 8-330 #8-331 8-331 #8-331 8-331 #8-331
#8-332 8-332 #8-332 8-332 #8-332 8-332 #8-333 8-333
8-333 #8-333 8-333 #8-334 8-334 #8-334 8-334 #8-334
#8-335 8-335 #8-335 8-335 #8-335 8-335 #8-336 8-336
8-336 #8-336 8-336
TSERRN = 000000 #6-13
TSEXCP = 000000 #8-272 8-272 #8-273 8-273 #8-274 8-274 #8-275 8-275 #8-317
8-317 #8-318 8-318 #8-320 8-320 #8-323 8-323 #8-330 8-330
TSFREE 105050 G 340-16868 340-16869 #340-16870
TSGMAN = 000000 #6-13
TSHILI = 000012 #8-272 8-272 #8-273 8-273 #8-274 8-274 #8-275 8-275 #8-317
8-317 #8-318 8-318 #8-320 8-320 #8-323 8-323 #8-330 8-330
TSLAST = 000000 #6-13
TSLOLI = 000001 #8-272 8-272 #8-273 8-273 #8-274 8-274 #8-275 8-275 #8-317
8-317 #8-318 8-318 #8-320 8-320 #8-323 8-323 #8-330 8-330
TSLSYM = 010000 #6-13 6-13 7-134 8-196 8-277 8-339
TSNEST = 177777 #6-13 6-39 #6-39 6-39 7-117 #7-117 7-117 7-134 7-134
7-134 #7-134 8-143 #8-143 8-143 8-196 8-196 8-196 #8-196
8-259 #8-259 8-259 8-277 8-277 8-277 #8-277 8-302 #8-302
8-302 8-314 8-316 8-319 8-322 8-324 8-329 8-339 8-339
8-339 #8-339 9-379 #9-379 9-379 9-385 9-385 9-385 #9-385
9-411 9-411 9-411 #9-411
TSNS0 = 000000 #6-39 9-411
TSNS1 = 000021 #7-117 7-134 #8-143 8-196 #8-259 8-277 #8-302 8-314 8-316
8-319 8-322 8-324 8-329 8-339 #9-379 9-385
TSPTHV = 000000 G 6-65 #340-16874
TSPTNU = 000000 #6-13
TSSAVL = 177777 #6-13
TSSEGL = 177777 #6-13
TSSUBN = 000000 #6-13

SYMBOL VALUE REFERENCES

```
T$TAGL  = 177777   #6-13
T$TAGN  = 010005   #6-13  7-117  7-117  #7-117  8-143  8-143  #8-143  8-259  8-259
                   #8-259  8-302  8-302  #8-302  9-379  9-379  #9-379
T$TEMP  = 000000   #7-101  7-101  7-101  #7-101  #7-134  7-134  #8-196  8-196  #8-272
                    8-272  #8-272  8-272  #8-272  8-272  #8-273  8-273  #8-273  8-273
                   #8-273  8-273  #8-274  8-274  #8-274  8-274  #8-274  8-274  #8-275
                    8-275  #8-275  8-275  #8-275  8-275  #8-277  8-277  #8-313  8-313
                   #8-313  8-313  #8-313  8-313  #8-315  8-315  #8-315  8-315  #8-315
                    8-315  #8-317  8-317  #8-317  8-317  #8-317  8-317  #8-318  8-318
                   #8-318  8-318  #8-318  8-318  #8-320  8-320  #8-320  8-320  #8-320
                    8-320  #8-321  8-321  #8-321  8-321  #8-321  8-321  #8-323  8-323
                   #8-323  8-323  #8-323  8-323  #8-325  8-325  #8-325  8-325  #8-325
                    8-325  #8-326  8-326  #8-326  8-326  #8-326  8-326  #8-327  8-327
                   #8-327  8-327  #8-327  8-327  #8-328  8-328  #8-328  8-328  #8-328
                    8-328  #8-330  8-330  #8-330  8-330  #8-330  8-330  #8-331  8-331
                   #8-331  8-331  #8-331  8-331  #8-332  8-332  #8-332  8-332  #8-332
                    8-332  #8-333  8-333  #8-333  8-333  #8-333  8-333  #8-334  8-334
                   #8-334  8-334  #8-334  8-334  #8-335  8-335  #8-335  8-335  #8-335
                    8-335  #8-336  8-336  #8-336  8-336  #8-336  8-336  #8-339  8-339
                   #9-385  9-385  #9-411  9-411
T$TEST  = 000000   #6-13
T$TSTM  = 177777   #6-13
T$TSTS  = 000000   #6-13
T$$HAR  = 010002   #8-259  8-259  8-277
T$$HW   = 010000   #7-117  7-117  7-134
T$$PRO  = 010004   #9-379
T$$SOF  = 010003   #8-302  8-302  8-339
T$$SW   = 010001   #8-143  8-143  8-196
T1      = 102712 G  7-101  #326-16093
UAM     = 000200 G #83-1776
UP.HAR  = 037422   #103-3203  168-7120  171-7268  185-7997  187-8109  204-9089  209-9351  211-9468  225-10197
                    227-10309  247-11461  252-11740  257-12019  259-12126  273-13107  273-13217  286-13981  288-14086
                    302-14839  304-14947  318-15672  319-15774
UP.RD.  = 045240   #141-5624  146-5880
UP.SOF  = 040104   #108-3452  169-7141  171-7285  185-8014  187-8126  205-9110  209-9368  211-9485  225-10214
                    227-10326  248-11482  253-11761  258-12040  259-12143  273-13124  273-13234  287-14002  289-14107
                    302-14856  304-14964  318-15689  320-15795
UP.WC.  = 045332   #142-5688  148-5978
UP.WR.  = 045146   #140-5560  144-5782
VEC     = 032710   #81-1652  *88-2290  89-2349
WAITER  = 044420   #134-5254  137-5421  138-5473
WBUFF   = 012670   #81-1638  83-1785  83-1786  83-1789  *114-3806  *114-3817  120-4175  *122-4250  *122-4251
                   *122-4258  *125-4372  *125-4373  *125-4374  *125-4379  156-6441  165-6907  165-6917  166-6963
                    169-7145  169-7155  171-7241  180-7734  200-8841  205-9149  220-9937  273-12852  282-13718
                    297-14566  313-15406
WCBUFF  = 013670   #83-1786  242-11159  242-11162
WC.COU  = 034330   #82-1674  *85-2114  142-5691  *142-5692  *142-5696
WC.MIL  = 034334   #82-1678  *85-2116  *142-5700  *142-5702  332-16441
WC.THO  = 034332   #82-1676  *85-2115  *142-5695  142-5697  *142-5699
WDBUFF  = 012670   #83-1785  242-11155  242-11161
WHY.DR  = 034446 G #82-1698  *85-2141  *94-2738  *95-2765  *96-2844  *96-2871  *165-6923  *165-6933  *165-6947
                   *166-6990  *167-7016  *167-7026  *167-7040  *167-7051  *169-7161  *169-7171  *170-7189  *170-7200
```

SYMBOL VALUE REFERENCES

*181-7773 *181-7784 *181-7799 *182-7860 *183-7886 *183-7897 *183-7908 *183-7924 *200-8872
*201-8886 *201-8900 *202-8959 *202-8981 *203-8995 *203-9009 *203-9020 *207-9221 *207-9243
*207-9253 *207-9267 *208-9282 *221-9973 *221-9984 *221-9999 *222-10060 *223-10086 *223-10097
*223-10112 *223-10124 *24?-11226 *243-11237 *244-11256 *245-11328 *245-11350 *246-11365 *246-11380
*246-11392 *248-11503 *248-11514 *249-11533 *250-11607 *250-11629 *251-11644 *251-11659 *251-11671
*253-11782 *253-11793 *254-11812 *255-11886 *255-11908 *255-11919 *256-11938 *256-11950 *273-12887
*273-12898 *273-12913 *273-12973 *273-12998 *273-13009 *273-13024 *273-13039 *282-13764 *283-13778
*283-13792 *284-13852 *284-13873 *285-13887 *285-13901 *285-13912 *298-14622 *298-14632 *299-14650
*300-14710 *300-14731 *300-14741 *301-14759 *301-14770 *314-15455 *314-15465 *314-15479 *315-15539
*316-15564 *316-15574 *316-15588 *316-15599 333-16496

WPTR 032670 #81-1640 156-6438 *156-6441 156-6442 *180-7734 180-7752 181-7766 182-7821 182-7824
182-7832 185-8018 *185-8019 *200-8841 200-8856 200-8866 201-8918 201-8921 201-8928
205-9114 *205-9115 *205-9149 206-9180 206-9183 206-9190 209-9377 *209-9378 *220-9937
220-9952 220-9962 221-10017 222-10024 222-10032 225-10218 *225-10219 *273-12852 273-12867
273-12880 273-12934 273-12937 273-12945 273-13128 *273-13129 *282-13718 282-13748 282-13758
283-13810 283-13813 283-13821 *287-14005 *297-14566 298-14606 298-14616 299-14668 299-14671
299-14679 302-14812 *302-14858 *313-15406 313-15435 314-15449 315-15501 315-15504 315-15512
*318-15691

WRD11 = 007240 #83-1828 90-2436 94-2720 94-2728 94-2751 96-2826 96-2853 333-16464 333-16490
WRD15 = 007246 #83-1829 100-3055 139-5497 144-5793 146-5891 148-5989
WRD16 = 007256 #83-1830 138-5485 144-5794 153-6258
WRD17 = 007264 #83-1831 138-5488 146-5892 153-6278
WRD18 = 007272 #83-1832 153-6257 153-6267 153-6277 154-6337 155-6361
WRD19 = 007300 #83-1833 154-6336
WRD2 = 007200 #83-1823 153-6259 153-6269 153-6279 325-16047
WRD20 = 007312 #83-1834 100-3056 155-6360
WRD21 = 007322 #83-1835 94-2727 94-2750 96-2825 96-2852 333-16489
WRD24 = 007332 #83-1836 242-11189 244-11264 254-11822
WRD25 = 007336 #83-1837 249-11543
WRD3 = 007206 #83-1824 339-16810 339-16818
WRD34 = 007344 #83-1838 89-2313 164-6866 180-7709 199-8784 219-9906 242-11149 322-15895
WRD35 = 007354 #83-1839 242-11184
WRD36 = 007370 #83-1840 88-2300 89-2320
WRD37 = 007400 #83-1841 88-2298 89-2322
WRD38 = 007412 #83-1842 89-2311
WRD4 = 007212 #83-1825 325-16045 339-16808 339-16817
WRD40 = 007416 #83-1843 89-2312 89-2325
WRD41 = 007424 #83-1844 89-2309
WRD6 = 007220 #83-1826 95-2772
WRD7 = 007230 #83-1827 95-2775

WRITE 045424 #143-5765 153-6255 153-6288 153-6294 154-6307 154-6322 165-6909 165-6914 180-7754
180-7759 200-8858 200-8863 220-9954 220-9959 243-11211 243-11216 248-11488 248-11493
253-11767 253-11772 273-12869 273-12877 282-13750 282-13755 298-14608 298-14613 313-15437
314-15446

WR.COU 034314 #82-1662 *84-2104 140-5563 *140-5564 *140-5568
WR.MIL 034320 #82-1666 *84-2106 *140-5572 *140-5574 332-16429
WR.THO 034316 #82-1664 *84-2105 *140-5567 140-5569 *140-5571
XSALWA = 000000 #6-13 8-319 8-324
XSFALS = 000040 #6-13 8-314 8-316
XSOFFS = 000400 #6-13 8-314 8-316 8-319 8-322 8-324 8-329
XSTRUE = 000020 #6-13 8-322 8-329
$END.L 105052 G #340-16881

SYMBOL	VALUE		REFERENCES								
$PATCH	004134	G	#9-402								
$SAVE2	005262	G	34-1022	39-1269	#41-1344	116-3959	125-4371	143-5765	145-5863	147-5961	341-16905
$SAVE3	005276	G	13-169	14-217	18-384	19-438	29-829	30-880	#41-1351	263-12282	341-16904
$SAVE4	005314	G	#41-1359	84-2078	88-2287	111-3636	114-3793	137-5408	264-12367	322-15893	341-16904
$SAVE5	005334	G	35-1091	39-1269	#41-1368	90-2410	93-2703	103-3204	108-3453	118-4094	131-4722
			133-5199	153-6245	164-6863	179-7702	199-8781	219-9903	242-11146	273-12823	281-13682
			297-14537	312-15384	332-16405	341-16904					
$T1	102536		#325-16039	326-16094							

```
                                                                I 14
CZMLBB     CREATED BY  MACRO  ON 29-MAR-82 AT 13:44     PAGE 20
MACRO CROSS REFERENCE                                   CREF
MACRO NAME      REFERENCES                                                                                     SEQ 0383
BGNHRD           8-259
BGNHW           #7-117
BGNMOD           6-39
BGNPRO           9-379
BGNSFT          #8-302
BGNSW           #8-143
DESCRI          #7-85
DEVTYP          #6-80
DISPAT           7-101
ENDHRD          #8-277
ENDHW            7-134
ENDMOD           9-411
ENDPRO          #9-385
ENDSFT           8-339
ENDSW            8-196
ERRTBL           7-94
GPRMA           #8-272  #8-273
GPRMD            8-274   8-275   8-317   8-318   8-320   8-323   8-330
GPRML           #8-313  #8-315  #8-321  #8-325  #8-326  #8-327  #8-328  #8-331  #8-332  #8-333
                #8-334  #8-335  #8-336
HEADER           6-65
M$BYTE          #6-65   #6-65   #6-65   #6-65
M$CNTO          #8-272   8-272  #8-273   8-273  #8-274   8-274  #8-275   8-275  #8-313   8-313
                #8-315   8-315  #8-317   8-317  #8-318   8-318  #8-320   8-320  #8-321   8-321
                #8-323   8-323  #8-325   8-325  #8-326   8-326  #8-327   8-327  #8-328   8-328
                #8-330   8-330  #8-331   8-331  #8-332   8-332  #8-333   8-333  #8-334   8-334
                #8-335   8-335  #8-336   8-336
M$DATA          #6-65    6-65    6-65    6-65    6-65    6-65    6-65    6-65    6-65    6-65
                 6-65    6-65    6-65    6-65    6-65    6-65   #6-65    6-65    6-65    6-65
                 6-65    6-65    6-65    6-65    6-65    6-65    6-65    6-65    6-65    6-65
                 6-65    6-65    6-65    6-65    6-65    6-65    6-65    6-65    6-65   #6-80
                 6-80   #7-85    7-85
M$DECR          #7-134   7-134  #8-196   8-196  #8-277   8-277  #8-339   8-339  #9-385   9-385
                #9-411   9-411
M$DEFA          #8-272  #8-272  #8-273  #8-273  #8-274  #8-274  #8-275  #8-275  #8-313  #8-313
                #8-315  #8-315  #8-317  #8-317  #8-318  #8-318  #8-320  #8-320  #8-321  #8-321
                #8-323  #8-323  #8-325  #8-325  #8-326  #8-326  #8-327  #8-327  #8-328  #8-328
                #8-330  #8-330  #8-331  #8-331  #8-332  #8-332  #8-333  #8-333  #8-334  #8-334
                #8-335  #8-335  #8-336  #8-336
M$ENDE          #7-134  #8-196  #8-277  #8-339  #9-411
M$EXCP          #8-272   8-272   8-272  #8-273   8-273   8-273  #8-274   8-274   8-274  #8-275
                 8-275   8-275  #8-317   8-317   8-317  #8-318   8-318   8-318  #8-320   8-320
                 8-320  #8-323   8-323   8-323  #8-330   8-330   8-330
M$GEN           #6-65   #6-65   #6-65   #6-65   #6-65   #6-65   #6-65   #6-65   #6-65   #6-65
                #6-65   #6-65   #6-65   #6-65   #6-65   #6-65   #6-65   #6-65   #6-65   #6-65
                #6-65   #6-65   #6-65   #6-65   #6-65   #6-65   #6-65   #6-65   #6-65   #6-65
                #6-65   #6-65   #6-65   #6-65   #6-65   #6-65   #6-65   #6-65   #6-65   #6-65
                #6-65   #6-65   #6-65   #6-65   #6-65   #6-65   #6-65   #6-65   #6-65   #6-65
                #6-65   #6-65   #6-65   #6-65   #6-65   #6-65   #6-65   #6-65   #6-65   #6-65
                #6-65   #6-65   #6-65   #6-65   #6-65   #6-65   #6-65   #6-65   #6-65   #6-65
                #6-65   #6-65   #6-65   #6-65   #6-65   #6-65   #6-80   #6-80   #7-85   #7-85
                #7-94   #7-94   #7-101  #7-101  #7-117  #7-117  #7-117  #7-117  #7-134  #7-134
```

```
CZMLBB     CREATED BY  MACRO  ON 29-MAR-82 AT 13:44     PAGE 21        J 14
MACRO CROSS REFERENCE                                   CREF
MACRO NAME          REFERENCES                                                                                                  SEQ 0384
                    #8-143    #8-143    #8-143    #8-143    #8-196    #8-196    #8-259    #8-259    #8-277    #8-277
                    #8-302    #8-302    #8-339    #8-339    #9-379    #9-379
M$GETS              #7-134     7-134    #8-196     8-196    #8-277     8-277    #8-314     8-314    #8-316     8-316
                    #8-319     8-319    #8-322     8-322    #8-324     8-324    #8-329     8-329    #8-339     8-339
                    #9-385     9-385    #9-411     9-411
M$GETT              #8-314     8-314    #8-316     8-316    #8-319     8-319    #8-322     8-322    #8-324     8-324
                    #8-329     8-329
M$GNGB              #6-39     #6-65     #6-65     #6-65     #6-65     #6-65     #6-65     #6-65     #6-65     #6-65
                    #6-65     #6-65     #6-65     #6-65     #6-65     #6-65     #6-65     #6-65     #6-65     #6-65
                    #6-65     #6-65     #6-65     #6-65     #6-65     #6-65     #6-65     #6-65     #6-65     #6-65
                    #6-65     #6-65     #6-65     #6-65     #6-65     #6-65     #6-65     #6-65     #6-65     #6-65
                    #6-65     #6-65     #6-65     #6-65     #6-65     #6-65     #6-65     #6-65     #6-65     #6-65
                    #6-65     #6-65     #6-65     #6-65     #6-65     #6-65     #6-65     #6-65     #6-65     #6-65
                    #6-65     #6-65     #6-65     #6-65     #6-65     #6-65     #6-65     #6-65     #6-65     #6-65
                    #6-65     #6-65     #6-65     #6-65     #6-65     #6-65     #6-65     #6-80     #6-80     #7-85
                    #7-85     #7-94     #7-94     #7-101    #7-101    #7-117    #7-117    #7-117    #8-143    #8-143
                    #8-143    #8-259    #8-259    #8-302    #8-302    #9-379    #9-379
M$GNIN              #6-65      6-65      6-65      6-65      6-65      6-65     #6-65      6-65     #6-65      6-65
                    #6-65      6-65     #6-65      6-65     #6-65      6-65     #6-65      6-65     #6-65      6-65
                    #6-65      6-65     #6-65      6-65     #6-65      6-65     #6-65      6-65     #6-65      6-65
                    #6-65      6-65     #6-65      6-65     #6-65      6-65     #6-65      6-65     #6-65      6-65
                    #6-65      6-65     #6-65     #6-65      6-65     #6-65      6-65     #6-65      6-65      6-65
                    #6-65      6-65     #6-65      6-65     #6-65      6-65     #6-65      6-65     #6-65      6-65
                    #6-65      6-65     #6-65      6-65     #6-65      6-65     #6-65      6-65     #6-65      6-65
                    #6-65      6-65     #6-65      6-65     #6-65      6-65     #6-65      6-65     #6-65      6-65
                    #6-65      6-65     #6-65      6-65     #6-65      6-65     #6-65      6-65     #6-65      6-65
                    #6-80     #6-80      6-80      6-80     #7-85     #7-85      7-85      7-85     #7-101     7-101
                    #7-101     7-101    #7-117     7-117    #8-143     8-143    #8-259     8-259    #8-272     8-272
                     8-272     8-272     8-272    #8-273     8-273     8-273     8-273     8-273    #8-274     8-274
                     8-274     8-274     8-274     8-274    #8-275     8-275     8-275     8-275     8-275     8-275
                    #8-277     8-277    #8-302     8-302    #8-313     8-313     8-313     8-313    #8-314     8-314
                    #8-315     8-315     8-315     8-315    #8-316     8-316    #8-317     8-317     8-317     8-317
                     8-317     8-317    #8-318     8-318     8-318     8-318     8-318     8-318    #8-319     8-319
                    #8-320     8-320     8-320     8-320     8-320     8-320    #8-321     8-321     8-321     8-321
                    #8-322     8-322    #8-323     8-323     8-323     8-323     8-323     8-323    #8-324     8-324
                    #8-325     8-325     8-325     8-325    #8-326     8-326     8-326     8-326    #8-327     8-327
                     8-327     8-327    #8-328     8-328     8-328     8-328    #8-329     8-329    #8-330     8-330
                     8-330     8-330     8-330     8-330    #8-331     8-331     8-331     8-331    #8-332     8-332
                     8-332     8-332    #8-333     8-333     8-333     8-333    #8-334     8-334     8-334     8-334
                    #8-335     8-335     8-335     8-335    #8-336     8-336     8-336     8-336    #8-339     8-339
M$GNTA              #7-134    #7-134    #8-196    #8-196    #8-277    #8-277    #8-339    #8-339
M$HAPT              #6-65     #6-65
M$HNAP              #6-65     #6-65
M$INCR              #6-39     #6-39     #7-117    #7-117    #7-117    #7-117    #8-143    #8-143    #8-143    #8-143
                    #8-259    #8-259    #8-259    #8-259    #8-302    #8-302    #8-302    #8-302    #9-379    #9-379
                    #9-379    #9-379
M$MCHI              #6-13     #6-13
M$MCLO              #6-13     #6-13
M$POP               #7-134     7-134    #8-196     8-196    #8-277     8-277    #8-339     8-339    #9-385     9-385
                    #9-411     9-411
M$PUSH              #6-39     #6-39     #7-117    #7-117    #8-143    #8-143    #8-259    #8-259    #8-302    #8-302
                    #9-379    #9-379
```

MACRO NAME REFERENCES

```
MSRADI    #8-272  #8-272  #8-273  #8-273  #8-274  #8-274  #8-275  #8-275  #8-313  #8-313
          #8-315  #8-315  #8-317  #8-317  #8-318  #8-318  #8-320  #8-320  #8-321  #8-321
          #8-323  #8-323  #8-325  #8-325  #8-326  #8-326  #8-327  #8-327  #8-328  #8-328
          #8-330  #8-330  #8-331  #8-331  #8-332  #8-332  #8-333  #8-333  #8-334  #8-334
          #8-335  #8-335  #8-336  #8-336
MSSETS    #6-39   #6-39   #7-117  #7-117  #8-143  #8-143  #8-259  #8-259  #8-302  #8-302
          #9-379  #9-379
MSWORD    #6-65   #6-65   #7-101  #7-101  #7-101  #8-272  #8-272  #8-273  #8-273  #8-274
          #8-274  #8-275  #8-275  #8-313  #8-313  #8-314  #8-314  #8-315  #8-315  #8-316
          #8-316  #8-317  #8-317  #8-318  #8-318  #8-319  #8-319  #8-320  #8-320  #8-321
          #8-321  #8-322  #8-322  #8-323  #8-323  #8-324  #8-324  #8-325  #8-325  #8-326
          #8-326  #8-327  #8-327  #8-328  #8-328  #8-329  #8-329  #8-330  #8-330  #8-331
          #8-331  #8-332  #8-332  #8-333  #8-333  #8-334  #8-334  #8-335  #8-335  #8-336
          #8-336
MSXFER    #8-314  #8-314  #8-316  #8-316  #8-319  #8-319  #8-322  #8-322  #8-324  #8-324
          #8-329  #8-329
POINTE     6-46
SVC       #6-12    6-13
XFER       8-319    8-324
XFERF      8-314    8-316
XFERT     #8-322  #8-329
```