

ML11

ML11 LOGIC TEST
CZMLA0

AH-S390A-MC
FICHE 1 OF 2

FEB 1981
COPYRIGHT © 1981
MADE IN USA



ML11

ML11 LOGIC TEST
CZMLAAO

AH-S390A-MC
FICHE 2 OF 2

FEB 1981
COPYRIGHT © 1981
MADE IN USA



1
2

.TITLE CZMLAAO ML-11 LOGIC TEST
.SBTTL USER DOCUMENTATION
.REM 8

IDENTIFICATION

PRODUCT CODE: AC-S388A-MC
PRODUCT NAME: CZMLAAO ML11 LOGIC TEST
PRODUCT DATE: 2-FEB-81
MAINTAINER: TOM LANWSBY
AUTHOR: D.W.NEALE

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

NO RESPONSIBILITY IS ASSUMED FOR THE USE OR RELIABILITY OF SOFTWARE ON EQUIPMENT THAT IS NOT SUPPLIED BY DIGITAL OR ITS AFFILIATED COMPANIES.

COPYRIGHT (C) 1981 BY DIGITAL EQUIPMENT CORPORATION

THE FOLLOWING ARE TRADEMARKS OF DIGITAL EQUIPMENT CORPORATION:

DIGITAL PDP UNIBUS MASSBUS
DEC DECUS DECTAPE

TABLE OF CONTENTS

2-	2	USER DOCUMENTATION
7-	1	PROGRAM HEADER AND TABLES
46-	1	MISCELLANEOUS CODING SECTION
47-	1	HARDWARD TEST SECTION

TABLE OF CONTENTS

1.0	GENERAL INFORMATION
1.1	PROGRAM ABSTRACT
1.2	SYSTEM REQUIREMENTS
1.3	RELATED DOCUMENTS AND STANDARDS
1.4	DIAGNOSTIC HIERARCHY PREREQUISITES
2.0	OPERATING INSTRUCTIONS
2.1	COMMANDS
2.2	SWITCHES
2.3	FLAGS
2.4	HARDWARE QUESTIONS
2.5	SOFTWARE QUESTIONS
2.6	EXTENDED P-TABLE DIALOGUE
2.7	QUICK STARTUP PROCEDURE
3.0	ERROR INFORMATION
4.0	PERFORMANCE AND PROGRESS REPORTS
5.0	DEVICE INFORMATION TABLES
6.0	TEST SUMMARIES

1.0 GENERAL INFORMATION

1.1 PROGRAM ABSTRACT

DIAGNOSTIC ENGINEERING WAS CONTRACTED BY MEMORY ENGINEERING TO MAKE THIS DIAGNOSTIC IN ORDER TO AID MEMORY ENGINEERING TO DESIGN AND DEBUG THE ML-11, AID FIELD SERVICE FOR FIELD REPAIRS AND INSTALLATIONS OF THE ML-11.

THIS DIAGNOSTIC PRODUCT WILL BE DESIGNED TO TEST FROM ONE TO EIGHT ML-11 UNITS OFF A SINGLE RH11 OR RH70 CONTROLLER.

THE FUNCTIONAL LEVEL (FRU) OF THIS DIAGNOSTIC PRODUCT WILL BE TO THE LOGIC FUNCTION LEVEL (I.E. DRIVE SELECTION). UPON DETECTION OF AN ERROR BY THE DIAGNOSTIC, THE LOGIC FUNCTION AND RESPECTIVE MODULE WHICH IT IS LOCATED ON WILL BE PRINTED TO THE OPERATOR.

THIS DIAGNOSTIC HAS BEEN WRITTEN FOR USE WITH THE DIAGNOSTIC RUNTIME SERVICES SOFTWARE (SUPERVISOR). THESE SERVICES PROVIDE THE INTERFACE TO THE OPERATOR AND TO 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

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

PDP-11 WITH MINIMUM OF 28K WORDS OF MEMORY

CONSOLE TERMINAL

RH11 OR RH70

1 TO 8 ML-11 DRIVES ON INTERMIXED BUS

XXDP+ LOAD MEDIA

1.3 RELATED DOCUMENTS AND STANDARDS

1. SUPPRGC.DOC
2. SUPINT.MEN
3. SUPFUN.C
4. XXDPPLUS.DOC
5. BLISS LANGUAGE GUIDE
6. BLISS-16 USER'S GUIDE

1.4 DIAGNOSTIC HIERARCY PREREQUISITES

IT WILL BE ASSUMED THAT PRIOR TO THE RUNNING OF THIS DIAGNOSTIC THAT ALL APPROPRIATE CPU, MAIN MEMORY AND RH CONTROLLER DIAGNOSTICS HAVE BEEN SUCCESSFULLY RUN.

THIS DIAGNOSTIC WILL HOWEVER PERFORM MINIMAL RH TESTS TO ENSURE ITS EXISTANCE AND BASIC FUNCTIONALITY BEFORE LOGIC TESTS ARE ALLOWED TO RUN.

2.0 OPERATING INSTRUCTIONS.

THIS SECTION CONTAINS A BRIEF DESCRIPTION OF THE RUNTIME SERVICES. FOR DETAILED INFORMATION, REFER TO THE XXDP+ USER'S MANUAL (CHQUS).

2.1 COMMANDS

THERE ARE ELEVEN LEGAL COMMANDS FOR THE DIAGNOSTIC RUNTIME SERVICES (SUPERVISOR). THIS SECTION LISTS THE COMMANDS AND GIVES A VERY BRIEF DESCRIPTION OF THEM. THE XXDP+ USER'S MANUAL HAS MORE DETAILS.

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 XXDP+ MONITOR (XXDP+ OPERATION ONLY!)
ADD	ACTIVATE A UNIT FOR TESTING (ALL UNITS ARE CONSIDERED TO BE ACTIVE AT START TIME)
DROP	DEACTIVATE A UNIT
PRINT	PRINT STATISTICAL INFORMATION (IF IMPLEMENTED BY THE DIAGNOSTIC - SECTION 4.0)
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)

A COMMAND CAN BE RECOGNIZED BY THE FIRST THREE CHARACTERS. SO YOU MAY, FOR EXAMPLE, TYPE "STA" INSTEAD OF "START".

2.2 SWITCHES

THERE ARE SEVERAL SWITCHES WHICH ARE USED TO MODIFY SUPERVISOR OPERATION. THESE SWITCHES ARE APPENDED TO THE LEGAL COMMANDS. ALL OF THE LEGAL SWITCHES ARE TABULATED BELOW WITH A BRIEF DESCRIPTION OF EACH. IN THE DESCRIPTIONS BELOW, A DECIMAL NUMBER IS DESIGNATED BY "DDDD".

SWITCH	EFFECT
/TESTS:LIST	EXECUTE ONLY THOSE TESTS SPECIFIED IN THE LIST. LIST IS A STRING OF TEST NUMBERS, FOR EXAMPLE - /TESTS:1:5:7-10. THIS LIST WILL CAUSE TESTS 1,5,7,8,9,10 TO BE RUN. ALL OTHER TESTS WILL NOT BE RUN.
/PASS:DDDD	EXECUTE DDDDD PASSES (DDDD = 1 TO 64000)
/FLAGS:FLGS	SET SPECIFIED FLAGS. FLAGS ARE DESCRIBED IN 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:0:5:10-12
 USE UNITS 0,5,10,11,12 (UNIT NUMBERS = 0-63)

EXAMPLE OF SWITCH USAGE:

START/TESTS:1-5/PASS:1000/EOP:100

THE EFFECT OF THIS COMMAND WILL BE: 1) TESTS 1 THROUGH 5 WILL BE EXECUTED, 2) ALL UNITS WILL TESTED 1000 TIMES AND 3) THE END OF PASS MESSAGES WILL BE PRINTED AFTER EACH 100 PASSES ONLY. A SWITCH CAN BE RECOGNIZED BY THE FIRST THREE CHARACTERS. YOU MAY, FOR EXAMPLE, TYPE "/TES:1-5" INSTEAD OF "/TESTS:1-5".

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

	TESTS	PASS	FLAGS	EOP	UNITS
START	X	X	X	X	X
RESTART	X	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 OPERATIONAL 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 OR CLEARED AS SPECIFIED BY THE LAST FLAG SWITCH.

FLAG	EFFECT
HOE	HALT ON ERROR - CONTROL IS RETURNED TO RUNTIME SERVICES COMMAND MODE
LOE	LOOP ON ERROR
IER*	INHIBIT ALL ERROR REPORTS
IBR*	INHIBIT ALL ERROR REPORTS EXCEPT FIRST LEVEL (FIRST LEVEL CONTAINS ERROR TYPE, NUMBER, PC, TEST AND UNIT)
IXR*	INHIBIT EXTENDED ERROR REPORTS (THOSE CALLED BY PRINTX MACRO'S)
PRI	DIRECT MESSAGES TO LINE PRINTER

PNT	PRINT TEST NUMBER AS TEST EXECUTES
BOE	"BELL" ON ERROR
UAM	UNATTENDED MODE (NO MANUAL INTERVENTION)
ISR	INHIBIT STATISTICAL REPORTS (DOES NOT APPLY TO DIAGNOSTICS WHICH DO NOT SUPPORT STATISTICAL REPORTING)
IDR	INHIBIT PROGRAM DROPPING OF UNITS
ADR	EXECUTE AUTODROP CODE
LOT	LOOP ON TEST
EVL	EXECUTE EVALUATION (ON DIAGNOSTICS WHICH HAVE EVALUATION SUPPORT)

*ERROR MESSAGES ARE DESCRIBED IN SECTION 3.1

SEE THE XXDP+ USER'S MANUAL FOR MORE DETAILS ON FLAGS. YOU MAY SPECIFY MORE THAN ONE FLAG WITH THE FLAG 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 RUNTIME SERVICES WILL PROMPT 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 CHAPTER 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.

PARAMETER CODING CALLS

GPRMA	MSGH1,0,0,0,177777,YES	:RH ADDRESS
GPRMD	MSGH2,2,0,77,11,70,YES	:RH TYPE
GPRMD	MSGH3,4,0,777,0,777,YES	:RH VECTOR ADDRESS
GPRMD	MSGH4,6,D,77,1,16,,YES	:NUMBER FO ARRAYS
GPRML	MSGH5,10,1,YES	:DRIVE OPTIONS
GPRMD	MSGH6,12,0,7,0,7,YES	:DRIVE NUMBER
GPRML	MSGH7,14,1,YES	:PARITY DISABLED

PARAMETER CODING MESSAGES

MSGH1:	.ASCIZ	/RH ADDRESS?/
MSGH2:	.ASCIZ	/IS RH AN '70' OR '11?/
MSGH3:	.ASCIZ	/RH VECTOR ADDRESS?/
MSGH4:	.ASCIZ	/NUMBER OF ARRAY MODULES?/
MSGH5:	.ASCIZ	/IS DRIVE OPTION AN ML11A?/
MSGH6:	.ASCIZ	/ML-11 DRIVE NUMBER?/
MSGH7:	.ASCIZ	/IS PARITY DISABLED?/

SAMPLE DIALOGUE

```
DR> STA <CR>
CHANGE HW <L> ? Y <CR>
# UNITS <D> ? 1 <CR>
UNIT 0
RH ADDRESS <O> 176400 ? <CR>
IS RH AN '70' OR '11' <O> ? <CR>
RH VECTOR ADDRESS <O> 204 ? <CR>
NUMBER OF ARRAYS MODULES ? <D> 16 ? 14 <CR>
IS DRIVE OPTION AN ML11A ? <L> Y ? <CR>
ML-11 DRIVE NUMBER ? <O> 0 ? <CR>
IS PARITY DISABLED ? <L> N ? <CR>
```

2.5 SOFTWARE QUESTIONS

AFTER YOU HAVE ANSWERED THE HARDWARE QUESTIONS OR AFTER A RESTART 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).

PARAMETER CODING CALLS

```
GPRML MSGS1,0,1,NO ;PRINT THE DRIVE SERIAL NUMBER
```

PARAMETER CODING MESSAGES

```
MSG1: .ASCIZ /PRINT SERIAL NO.??/
```

SAMPLE DIALOGUE

```
PRINT SERIAL NO. ? <L> N ? Y <CR>
```

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, THIS BECOMES TEDIOUS SINCE MOST OF THE ANSWERS ARE REPETITIOUS.

TO ILLUSTRATE A MORE EFFICIENT METHOD, SUPPOSE YOU ARE TESTING A FICTIONAL DEVICE, THE XY11. SUPPOSE THIS DEVICE CONSISTS OF A CONTROL MODULE WITH EIGHT UNITS (SUB-DEVICES) ATTACHED TO IT. THESE UNITS ARE DESCRIBED BY THE OCTAL NUMBERS 0 THROUGH 7. THERE

IS ONE HARDWARE PARAMETER THAT CAN VARY AMONG UNITS CALLED THE Q-FACTOR. THIS Q-FACTOR MAY BE 0 OR 1. BELOW IS A SIMPLE WAY TO BUILD A TABLE FOR ONE XY11 WITH EIGHT UNITS.

UNITS (D) ? 8<CR>

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

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

UNIT 3
CSR ADDRESS (O) ? 160000<CR>
SUB-DEVICE # (O) ? 2<CR>
Q-FACTOR (O) 0 ? <CR>

UNIT 4
CSR ADDRESS (O) ? 160000<CR>
SUB-DEVICE # (O) ? 3<CR>
Q-FACTOR (O) 0 ? <CR>

UNIT 5
CSR ADDRESS (O) ? 160000<CR>
SUB-DEVICE # (O) ? 4<CR>
Q-FACTOR (O) 0 ? <CR>

UNIT 6
CSR ADDRESS (O) ? 160000<CR>
SUB-DEVICE # (O) ? 5<CR>
Q-FACTOR (O) 0 ? <CR>

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

UNIT 8
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 FROM THE ABOVE EXAMPLE, THE HARDWARE PARAMETERS DO NOT VARY SIGNIFICANTLY FROM UNIT TO UNIT. THE PROCEDURE SHOWN IS NOT VERY EFFICIENT.

THE RUNTIME SERVICES CAN TAKE MULTIPLE UNIT SPECIFICATIONS HOWEVER. LET'S BUILD THE SAME TABLE USING THE MULTIPLE SPECIFICATION FEATURE.

UNITS (D) ? 8<CR>

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

UNIT 3
CSR ADDRESS (O) ? 160000<CR>
SUB-DEVICE # (O) ? 2-5<CR>
Q-FACTOR (O) 0 ? 0<CR>

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

AS YOU CAN SEE IN THE ABOVE DIALOGUE, THE RUNTIME SERVICES WILL BUILD AS MANY ENTRIES AS IT CAN WITH THE INFORMATION GIVEN IN ANY ONE PASS THROUGH THE QUESTIONS. IN THE FIRST PASS, TWO ENTRIES ARE BUILT SINCE TWO SUB-DEVICES AND Q-FACTORS WERE SPECIFIED. THE SERVICES ASSUME THAT THE CSR ADDRESS IS 160000 FOR BOTH SINCE IT WAS SPECIFIED ONLY ONCE. IN THE SECOND PASS, FOUR ENTRIES WERE BUILT. THIS IS BECAUSE FOUR SUB-DEVICES WERE SPECIFIED. THE "-" CONSTRUCT TELLS THE RUNTIME SERVICES TO INCREMENT THE DATA FROM THE FIRST NUMBER TO THE SECOND. IN THIS CASE, SUB-DEVICES 2, 3, 4 AND 5 WERE SPECIFIED. (IF THE SUB-DEVICE WERE SPECIFIED BY ADDRESSES, THE INCREMENT WOULD BE BY 2 SINCE ADDRESSES MUST BE ON AN EVEN BOUNDARY.) THE CSR ADDRESSES AND Q-FACTORS FOR THE FOUR ENTRIES ARE ASSUMED TO BE 160000 AND 0 RESPECTIVELY SINCE THEY WERE ONLY SPECIFIED ONCE. THE LAST TWO UNITS ARE SPECIFIED IN THE THIRD PASS.

THE WHOLE PROCESS COULD HAVE BEEN ACCOMPLISHED IN ONE PASS AS SHOWN BELOW.

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 NULL FIELD) TELL THE RUNTIME SERVICES TO REPEAT THE LAST REPLY.

2.7 QUICK START-UP PROCEDURE (XXDP+)

TO START-UP THIS PROGRAM:

1. BOOT XXDP+
2. GIVE THE DATE AND ANSWER THE LSI AND 50HZ (IF THERE IS A CLOCK) QUESTIONS
3. TYPE 'R NAME', WHERE NAME IS THE NAME 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. THESE DEFAULTS ARE DESCRIBED IN SECTIONS 2.3 AND 2.5.

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

ERROR MESSAGES ARE HANDLED VIA A DICTIONARY STRUCTURE.

WORDS AND PHRASES ARE MULTIPLY REFERENCED USING ONLY ONE COPY OF THE WORD OR PHRASE IN CORE.

THIS PERMITS EXTENSIVE ERROR MESSAGE PRINTING AT MINIMAL STORAGE REQUIREMENTS.

THE FOLLOWING DEMONSTRATES TYPICAL ERROR MESSAGES:

ML11 DVC FTL ERR 00077 ON UNIT 07 TST 027 SUB 002 PC: 050432

ASYNCHRONOUS MODULE FAILURE
EXCESSIVE DATA ERRORS DURING INITIAL ARR RD_WRT

ML11 DVC FTL ERR 00112 ON UNIT 03 TST 037 SUB 000 PC: 056466
ASYNCHRONOUS MODULE FAILURE
ARRAY ADRS MULTIPLEXER FAILURE
FAILED AT DSA: 000000

3.2.1 ERROR NUMBER DEFINITION

ERROR NO.	FAILING LOGIC
-----	-----
1	DRIVE DID NOT RESPOND WITHIN 1.5 US
2	DSA REG READ/WRITE ERROR DURING DRIVE SEL TEST
3	UNIQUE DRIVE SELECTION ERROR
4	ML REGISTER READ WRITE ONES/ZEROES ERROR
5	ML REGISTER READ WRITE ONES/ZEROES ERROR
6	ML REGISTER !INITIALIZATION ERROR
7	CONTROL BUS BAD PARITY NOT DETECTED
8	CONTROL BUS GOOD PARITY NOT DETECTED.
9	CONTROL BUS BAD PARITY GENERATED.
10	ARRAY SIZING LOGIC ERROR
11	GO BIT NOT CLR AFTER NOOP FUNCTION
12	ILF BIT SET DURING NOOP FUNCTION
13	OPI BIT SET DURING NOOP FUNCTION
14	GO BIT NOT SET DURING WRITE CHECK FUNCTION
15	DRY BIT NOT CLEAR DURING WRITE CHECK FUNCTION

16 DRY BIT SET WHEN GO SET
DURING WRITE CHECK FUNCTION

17 ILF SET DURING WRITE
CHECK FUNCTION

18 OPI BIT SET DURING WRITE
CHECK FUNCTION

19 GO BIT NOT CLEAR AFTER WRITE
CHECK FUNCTION 'ASYNC FAILURE'

20 GO BIT NOT CLEAR AFTER WRITE
CHECK FUNCTION 'SYNC FUNCTION'

21 DRY BIT NOT SET AFTER WRITE
CHECK FUNCTION

22 GO BIT NOT CLEAR AFTER WRITE
CHECK FUNCTION 'ASYNC FAILURE'

23 GO BIT NOT CLEAR AFTER WRITE
CHECK FUNCTION 'SYNC FAILURE'

24 GO BIT NOT SET DURING WRITE
FUNCTION

25 DRY BIT CLEAR WITH GO CLEAR
DURING WRITE FUNCTION.

26 DRY BIT SET WITH GO BIT
DURING WRITE FUNCTION.

27 ILF BIT SET DURING WRITE
FUNCTION

28 OPI BIT SET DURING WRITE FUNCTION

29 GO BIT NOT CLEAR AFTER WRITE
FUNCTION 'ASYNC FAILURE'

30 GO BIT NOT CLEAR AFTER WRITE
FUNCTION 'SYNC FAILURE'

31 DRY BIT NOT SET AFTER WRITE
FUNCTION.

32 GO BIT NOT CLEAR AFTER WRITE
FUNCTION 'ASYNC FAILURE'

33 GO BIT NOT CLEAR AFTER WRITE
FUNCTION 'ASYNC FAILURE'

34 GO BIT NOT SET DURING READ
FUNCTION.

- 35 DRY BIT SET WHILE GO CLEAR
DURING READ FUNCTION.
- 36 DRY BIT SET WHILE GO SET
DURING READ FUNCTION.
- 37 ILF BIT SET DURING READ
FUNCTION.
- 38 OPI BIT SET DURING READ
FUNCTION
- 39 GO BIT NOT CLEAR AFTER READ
FUNCTION 'ASYNC FAILURE'
- 40 GO BIT NOT CLEAR AFTER READ
FUNCTION 'SYNC FAILURE'
- 41 DRY BIT NOT SET AFTER READ
FUNCTION
- 42 GO BIT NOT CLEAR AFTER READ
FUNCTION 'ASYNC FAILURE'
- 43 GO BIT NOT CLEAR AFTER READ
FUNCTION 'SYNC FAILURE'
- 44 GO BIT NOT CLEAR AFTER
CLEAR FUNCTION
- 45 DRY BIT SET WHILE GO SET
DURING CLEAR FUNCTION
- 46 DRY BIT NOT SET AFTER
CLEAR FUNCTION.
- 47 ILF BIT SET DURING CLEAR
FUNCTION
- 48 OPI BIT SET DURING CLEAR
FUNCTION
- 50 COMPOSITE ERROR BIT NOT
SET AFTER MLER BITS SET
- 51 ATA BIT SETTING ERROR
- 52 ATTN BIT SETTING ERROR
- 53 ATTN BIT NOT CLEARED BY
MLCS1 NOOP FUNCTION
- 54 ATA BIT NOT CLEAR AFTER
WRITING A ONE TO IT
- 55 WRITING A ONE TO OTHER
DRIVES ATA BIT CLEARED

THIS DRIVES ATA BIT

56 GO BIT NOT CLEARED AFTER
SEARCH FUNCTION

57 ILF BIT SET DURING SEARCH
FUNCTION

58 NO-OP FUNCTION DID NOT
CLEAR THE ATA BIT AFTER BEING
SET

59 ATA BIT NOT SET AFTER
SEARCH FUNCTION

60 OPI BIT SET DURING SEARCH
FUNCTION AT PRESENT ARRAYS

61 ATA BIT SET AFTER SEARCH
AT NOT PRESENT ARRAYS.

62 GO BIT NOT CLEAR AFTER
READ-IN-PRESET FUNCTION

63 ILF BIT SET DURING
READ-IN-SET FUNCTION

64 OPI BIT SET DURING
READ-IN-PRESET FUNCTION

65 UV BIT NOT SET AFTER
READ-IN-PRESET

66 GO BIT NOT CLEARED
AFTER ILLEGAL FUNCTION

67 ILLEGAL FUNCTION NOT
DETECTED

68 OPI BIT SET WITH ILLEGAL
FUNCTION

69 RMR BIT NOT SET AFTER
MODIFYING REG WITH FUNCTION
IN PROGRESS.

70 MEMORY ARRAY PROM
CHECK SUM ERRORS DURING
INITIAL PROM READS

71 NIBBLE OFF SET COUNTS
GREATER THAN 14 DETECTED.

72 UNS BIT SET WITH GOOD
UV DATA

73 UNS BIT SET WITH GOOD

UV DATA

74 UNS BIT NOT SET WITH BAD
UV DATA

75 UNS BIT NOT SET WITH BAD
UV DATA

76 MEMORY ARRAY PROM ROW/
COL DATA ORING ERROR

77 BAD NIBBLE THRESHOLD OF
36 EXCEEDED DURING
INITIAL ARRAY READ/WRITE
TEST

78 UNIQUE PROM SELECTION
ERROR

79 FAILURE TO FIND GOOD ROW
DURING READ WRITE ARRAY
WITH PROM DATA

80 MEMORY ARRAY TIMING AND
CONTROL FAILURE TO REFRESH
MEMORY

81 DATA ERRORS DETECTED AT
LAST BLOCK DURING ADDRESSES
COUNTER TEST. (TEST ABORTED)

82 ADDRESS COUNTER ERROR

83 UNIQUE MEMORY ARRAY
MODULE SELECTION FAILURE

84 ALL BITS IN ALL NIBBLES
TESTED DURING SEQUENCER
EXISTENCE TEST WERE IN
ERROR (FAIL UNIT)

85 INTERMEDIATE FAILURE. SOME
BITS IN NIBBLES TESTED WERE
IN ERROR (CONTINUE TESTING)

86 SYNC BUS DATA BIT
WRITE PATH CONTINUITY
FAILURE

87 SYNC BUS DATA BIT
READ PATH CONTINUITY
FAILURE.

88 RAM BUS ADRS COUNTER
FAILURE TO LOAD/UNLOAD
SKIP RAM DURING WRITE
FUNCTION

- 89 RAM BUS ADRS COUNTER
FAILURE TO LOAD/UNLOAD
SKIP RAM DURING READ
FUNCTION.
- 90 SYNC DATA BUS WRITE PATH
UNIQUE DATA BIT
FAILURE (ALL ONES NIBBLE
PATTERN)
- 91 SYNC DATA BUS WRITE
PATH UNIQUE DATA BIT
FAILURE (SHIFTED BIT
NIBBLE PATTERN)
- 92 SYNC DATA BUS WRITE
PATH UNIQUE DATA BIT
FAILURE (ALL ONES NIBBLE
PATTERN)
- 93 SYNC DATA BUS READ PATH
UNIQUE DATA BIT FAILURE
- 94 NIBBLE OFF SET
COUNTERS FAILURE
- 95 CS1 FUNCTION ABORT
FAILURE DURING CLASS
'A' ERROR.
- 96 CS1 FUNCTION ABORT
FAILURE DURING CLASS
'B' ERROR
- 97 LBT BIT SET BEFORE
A LAST BLOCK TRANSFER
- 98 DSA REGISTER INCREMENT
FAILURE DURING NON LAST
BLOCK TRANSFERS.
- 99 LBT BIT NOT CLEAR
AFTER LOADING DSA REG
- 100 LBT BIT NOT SET
AFTER A LAST BLOCK
TRANSFER
- 101 DSA REGISTER
INCREMENT FAILURE
AFTER A LAST BLOCK
TRANSFER
- 102 IAE BIT NOT SET AT
INVALID SECTOR ADDRESSES

103 AOE BIT NOT SET
AFTER ADDRESS OVERFLOW

104 SC BIT NOT AFTER
CS1 FUNCTION ADORT

105 GOOD DATA BUS
PARITY NOT DETECTED

106 GOOD DATA BUS
PARITY NOT GENERATED

107 UNS BIT SET AFTER
WRITING TO A SECTOR
DURING PROM DATA
TEST

108 UNS BIT NOT SET WITH
BAD UV DATA

109 WCE BIT SET DURING
MBUS WRITE/READ
FUNCTION TROUBLE SHOOTING
LOOP TEST

110 UNIQUE REGISTER
SELECTION TEST FAILURE

111 FAILURE TO FIND GOOD
MOS RAM ROW DURING
ARRAY ADRS MUX TEST
(INTERMEDIATE DIAG MSG)

112 UNIQUE ARRAY MODULE
ROW/COL ADDRESSING
FAILURE

113 DRIVE TYPE REGISTER VALUE
WAS NOT CORRECT

114 TRE BIT SET UNEXPECTEDLY
DURING A WRITE CHECK TRANSFER
(INTERMEDIATE DIAG ERROR)

115 TRE BIT SET UNEXPECTEDLY
DURING A WRITE TRANSFER
(INTERMEDIATE DIAG ERROR)

116 TRE BIT SET UNEXPECTEDLY
DURING A READ TRANSFER
(INTERMEDIATE DIAG ERROR)

117 TRE BIT DID NOT SET AFTER
A REGISTER MODIFICATION ERROR
(EXCEPTION WAS NOT ASSERTED)

4.0 PERFORMANCE AND PROGRESS REPORTS

AT THE END OF EACH PASS, THE PASS COUNT IS GIVEN ALONG WITH THE TOTAL NUMBER OF ERRORS REPORTED SINCE THE DIAGNOSTIC WAS STARTED. THE 'EOP' SWITCH CAN BE USED TO CONTROL HOW OFTEN THE END OF PASS MESSAGE IS PRINTED. SECTION 2.2 DESCRIBES SWITCHES.

5.0 DEVICE INFORMATION TABLES

HARDWARE DEFAULT PTABLE

.WORD	176400	;RH ADDRESS
.WORD	70	;RH TYPE
.WORD	204	;RH VECTOR ADDRESS
.WORD	16.	;NUMBER OF ARRAY MODULES
.WORD	1	;IS DRVE OPTION ML11A, 1=16K, 0=64K
.WORD	0	;ML-11 DRIVE NUMBER
.WORD	0	;IS PARITY DISABLED, 1=YES, 0=NO

SOFTWARE DEFAULT TABLE

PRSN: .WORD 0 ;PRINT SERIAL NUMBER, 1=YES, 0=NO

6.0 TEST SUMMARIES

TST1. DRIVE PRESENT

TEST TO SEE IF THE DRIVE UNDER TEST EXIST.

TST2. DRIVE SELECTION

SEE IF SELECTING OTHER DRIVES ON RH EFFECIS DRIVE UNDER TEST.

TST3. READ WRITE REGISTER ONE'S ZERO'S TEST

TEST REGISTERS READ WRITE CAPABILITY AND UNIQUENESS

TST4. READ WRITE REGISTER SHIFTING ONE'S AND ZEROES

TEST REGISTERS FOR UNIQUE DATA BITS.

TST5. REGISTER INITIALIZATION

TEST REGISTERS FOR CORRECT INIT DATA.

TST6. REGISTER SELECTION TEST

TEST FOR UNIQUE REGISTER SELECTION

TST7. PRINT DRIVE SERIAL NUMBER

PRINT THE CONTENTS OF MLSN IF THE SOFTWARE QUESTION WAS ANSWERED 'YES'.

TST8. C-BUS PARITY

TEST IF DRIVE CAN DETECT BAD PARITY ON C-BUS AND GENERATE GOOD PARITY.

TST9. MEMORY SIZING

SEE IF MEMORY SIZING LOGIC DETECTS AND RECORD CORRECT NUMBER OF ARRAYS PRESENT.

TST10. NO-OP FUNCTION

SEE IF A NO-OP FUNCTION CAN BE EXECUTED.

TST11. WRITE CHECK FUNCTION

SEE IF A WRITE CHECK FUNCTION CAUSES THE DRIVE TO HANG.

TST12. WRITE FUNCTION

SEE IF A WRITE FUNCTION CAUSES THE DRIVE TO HANG.

TST13. READ FUNCTION

SEE IF A READ FUNCTION CAUSES THE DRIVE TO HANG.

TST14. CLEAR FUNCTION

SEE IF A CLEAR FUNCTION CAN BE EXECUTED.

TST15. COMPOSITE ERROR BIT TEST

SEE IF EACH INDIVIDUAL ERROR BIT IN MLER CAUSES A COMPOSITE ERROR.

TST16. ATA BIT

TEST IF THE ATA BIT CAN BE SET AND CLEARED.

TST17. SEARCH FUNCTION

SEE IF A SEARCH FUNCTION CAN BE EXECUTED ON ALL PRESENT ARRAYS.

TST18. READ IN PRESET

TEST IF A READ IN PRESET FUNCTION SETS VOL V H.

TST19. ILLEGAL FUNCTION

SEE IF WRITING AN ILLEGAL FUNCTION TO CS1 CAN BE DETECTED AND THAT A TRANSFER IS NOT INITIATED.

TST20. REGISTER MODIFICATION REFUSED

TEST TO SEE IF WRITING TO SPECIFIC REGISTERS ARE ABORTED WHILE THE DRIVE IS ACTIVE. SEE IF WRITING TO NON-SPECIFIC REGISTERS ARE ALLOWED WHILE DRIVE IS ACTIVE.

TST21. INITIAL PROM TEST

TEST PROMS FOR EXISTENCE.

TST22. PROM 'OR' FUNCTION TEST

TEST THE PROM DATA ORING FUNC

TST23. UV ERROR TEST

TEST ABILITY OF UV ERR PROMS TO DETECT ALL POSSIBLE CHECK SUM
ERRORS.

TST24. INITIAL ARRAY TEST

TEST ARRAY TIMING AND CONTROL FOR EXISTENCE.

TST25. PROM SELECTION TEST

TEST FOR UNIQUE PROM SELECTION.

TST26. READ WRITE MEMORY ARRAY WITH PROM DATA (DIAG MODE)

SEE IF MEMORY CAN BE WRITTEN AND READ.

ALSO FIND ERROR FREE BLOCK OF MEMORY FOR FUTURE TESTS.

TST27. REFRESH TIMING

TEST TO SEE IF MEMORY CAN BE REFRESHED.

TST28. ADDRESS COUNTER

TEST THE ADDRESS COUNTER FOR ABILITY TO COUNT THROUGH ALL
POSSIBLE MEMORY ADDRESSES.

TST29. ARRAY MODULE SELECTION

TEST FOR UNIQUE ARRAY MODULE SELECTION

TST30. SEQUENCER EXISTENCE TEST

TEST TO SEE IF BASIC SEQUENCER TIMING EXISTS.

TST31. SYNC DATA BUS CONTINUITY/WRITE PATH

TEST SYNCHRONOUS DATA BUS WRITE PATH FOR CONTINUITY BY READ-
ING WRITING ONE'S AND ZERO'S.

TST32. SYNC DATA BUS CONTINUITY/READ PATH

TEST SYNCHRONOUS DATA BUS READ PATH FOR CONTINUITY BY READING
WRITING ONE'S AND ZEROES.

TST33. RAM-BUS ADDRESS COUNTER/WRITE PATH

TEST ABILITY OF THE RAM-BUS ADDRESS COUNTERS TO LOAD/UNLOAD
THE SKIP DURING WRITE FUNCTIONS.

TST34. RAM BUS ADRS COUNTER/READ PATH

TEST ABILITY OF RAM/BUS ADRS COUNTERS TO LOAD/UNLOAD THE SKIP
RAM DURING READ FUNCTIONS.

TST35. SYNC DATA BUS BIT UNIQUENESS/WRITE PATH

TEST SYNCHRONOUS DATA BUS FOR DATA BIT UNIQUENESS BY WRITING
SHIFTING PATTERNS OF ONE'S AND ZERO'S TO THE ML.

TST36. SYNC DAT BUS BIT UNIQUENESS/READ PATH

TEST SYNCHRONOUS DATA BUS READ PATH FOR DATA BIT UNIQUENESS
BY WRITING SHIFTING PATTERNS OF ONES AND ZEROES TO THE

TST37. ARRAY ADDRESS MUX

TEST FOR UNIQUE ROW AND COLUMN ADDRESSING

TST38. NIBBLE OFFSET

TEST NIBBLE OFFSET COUNTERS TO COUNT TO 14 NIBBLE DATA TO BE
SHIFTED ON DETECTION OF BAD NIBBLES.

TST39. CS1 FUNCTION ABORT

SEE IF A CLASS 'B' ERROR ABORTS A FUNCTION WHILE IN PROGRESS.

SEE IF A CLASS 'A' ERROR IS DETECTED BUT FUNCTION IS ALLOWED TO COMPLETE.

TST40. LAST BLOCK INDICATOR

TEST THE LAST BLOCK INDICATOR BIT FOR NOT SETTING BELOW THE LAST AND SETTING AND CLRING AT THE LAST BLOCK

TST41. INVALID ADDRESS TEST

FOR ALL ILLEGAL DSA ADDRESSES READ THE IAE BIT SET.

TST42. ADDRESS OVERFLOW

TEST FOR AOE ON TRANSFERS WHICH EXTEND BEYOND THE LAST BLOCK.

TST43. SYNC BUS PARITY

TEST FOR BAD PARITY DETECTION AND GOOD PARITY GENERATION.

TST44. WRITE READ MEMORY ARRAY (M-BUS BLOCK MODE)

WRITE READ MEMORY VIA M-BUS BLOCK MODE AT MAX SPEED.

TST45. PROM DATA TEST

VERIFY THAT CHECK SUM VALUES FOR ALL PROM LOCATIONS ARE CORRECT.

1
33
35
36
38
39
40
41
42
43
44
45
46
47
64
65
66
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
111
112
113
114
115
116
117
118

000000 002000
002000
002000
002122
002130
002152 000000
002152 000000
002154 000000
002156 000000
002160 000000
002162

.SBTTL PROGRAM HEADER AND TABLES

.ENABL ABS,AMA
= 2000

BGNMOD

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

POINTER ALL

HEADER ML11,A,0,1800.,0

:
: NAMES OF DEVICES SUPPORTED BY THIS PROGRAM
:

DEVTYP <ML-11>

:
: TEST DESCRIPTION
:

DESCRIPT <ML-11 LOGIC TEST>

:
: THE GLOBAL ERROR TABLE (INFORMATION
: USED IN A CALL TO THE MACRO "ERROR")
:

ERRTBL
ERRTYP:: .WORD 0
ERRNBR:: .WORD 0
ERRMSG:: .WORD 0
ERRBLK:: .WORD 0

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

DISPATCH 45

:++
: 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.
:--

```
119 002316          BGNHW  DFPTBL
120
130
131 002320 176400   .WORD 176400   ;RH ADDRESS
132 002322 000070   .WORD 70       ;RH TYPE
133 002324 000204   .WORD 204      ;RH VECTOR ADDRESS
134 002326 000020   .WORD 16.     ;NUMBER OF ARRAY MODULES
135 002330 000001   .WORD 1       ;IS DRVE OPTION ML11A, 1=16K, 0=64K
136 002332 000000   .WORD 0       ;ML-11 DRIVE NUMBER
137 002334 000000   .WORD 0       ;IS PARITY DISABLED, 1=YES, 0=NO
138
139 002336          ENDSW
```

```
;++
; 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.
:--
```

```
149
150 002336          BGNSW  SFPTBL
151
159
160 002340 000000   PRSN: .WORD 0       ;PRINT SERIAL NUMBER, 1=YES, 0=NO
161
162 002342          ENDSW
163
```

189
215
216
217
218
219
220
221
222
223
224
225
226
227
237
238
239
240
241
242
243
244
245
246
247
248
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
284
285
286
287
288
289
290
297
298
299

002342

002344
002354
002366
002400
002412
002420
002432

002440

002440
002454
002502
002525
002556
002610
002634

122 110
111 123
122 110
116 125
111 123
115 114
111 123

040
040
040
115
040
055
040

;++
: THE HARDWARE 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.
:--

BGNHRD

GPRMA MSGH1,0,0,0,177777,YES
GPRMD MSGH2,2,0,77,11,70,YES
GPRMD MSGH3,4,0,777,0,777,YES
GPRMD MSGH4,6,D,77,1,16.,YES
GPRML MSGH5,10,1,YES
GPRMD MSGH6,12,0,7,0,7,YES
GPRML MSGH7,14,1,YES

ENDHRD

;++
: 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.
:--

BGNSFT

GPRML MSGS1,0,1,YES
.EVEN

ENDSFT

120 122

111

MSGS1: .ASCIZ /PRINT SERIAL NO.?
.EVEN

;PRINT DRIVE SERIAL NUMBER?

300
301
302
303
304
305
306
307
308
309
310
311
312
326
327
328
329
336
337
338
339
352

+++
: THIS TABLE IS USED BY THE RUNTIME SERVICES
: TO PROTECT THE LOAD MEDIA.

BGNPROT

-1 ;OFFSET INTO P-TABLE FOR CSR ADDRESS
-1 ;OFFSET INTO P-TABLE FOR MASSBUS ADDRESS
-1 ;OFFSET INTO P-TABLE FOR DRIVE NUMBER

ENDPROT

\$PATCH::
.BLKW 20

ENDMOD

.SBTTL MISCELLANEOUS CODING SECTION

17-Oct-1980 11:31:46
29-Sep-1980 10:13:18

TOPS-20 Bliss-16 V2(206)
PA:<NEALE>ML3.BLI.2 (1)

```
1  
6 ;ML3  
7 :  
8  
9 : 0001 MODULE ML3 =  
10 : 0002 BEGIN  
11 : 0003  
12 : 0004 REQUIRE 'MACRO.REQ';  
13 : 0718  
14  
15 : 0719 !+  
16 : 0720 ! THE REPORT CODING SECTION CONTAINS THE  
17 : 0721 ! 'PRINTS' CALLS THAT GENERATE STATISTICAL REPORTS.  
18 : 0722 !-  
19 : 0723  
20 : 0724 BGNRPT;  
21 : 0725 RETURN;  
22 : 0726 ENDRPT;  
26  
27  
28  
32 004114 000207 LRPT: RTS PC ; 0716  
33  
34 ; Routine Size: 1 word  
35 ; Maximum stack depth per invocation: 0 words  
40  
41  
45  
49 004116 004767 177772 L$RPT:: JSR PC,LRPT ; 0725  
50 004122 104425 TRAP 25  
51 004124 000207 RTS PC  
52  
53 ; Routine Size: 4 words  
54 ; Maximum stack depth per invocation: 0 words  
62  
63  
64 ; 0727  
65  
66 : 0728 !+  
67 : 0729 ! THE AUTODROP CODE IS EXECUTED IMMEDIATELY AFTER THE INITIALIZE  
68 : 0730 ! CODE IF THE 'ADR' FLAG WAS SET. THE UNIT(S) UNDER TEST ARE  
69 : 0731 ! CHECKED TO SEE IF THEY WILL RESPOND. THOSE THAT DON'T ARE IMMEDIATELY  
70 : 0732 ! DROPPED FROM TESTING. ISSUE A 'DODU' FOR THOSE THAT DON'T RESPOND.  
71 : 0733 !-  
72 : 0734  
73 : 0735 BGNAUTO;  
74 : 0736 RETURN;  
75 : 0737 ENDAUTO;  
79  
83 004126 000207 LAUTO: RTS PC ; 0726  
84  
85 ; Routine Size: 1 word  
86 ; Maximum stack depth per invocation: 0 words  
91  
92  
96  
100 004130 004767 177772 L$AUTO::JSR PC,LAUTO ; 0736
```

101 004134 104461
102 004136 000207

TRAP 61
RTS PC

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

103
104
105
110
111 : 0738

112 :ML3
113 :

17-Oct-1980 11:31:46
29-Sep-1980 10:13:18

TOPS-20 Bliss-16 V2(206)
PA:<NEALE>ML3.BLI.2 (1)

114
115

116 : 0739 !+
117 : 0740 !

THE DROP-UNIT SECTION CONTAINS THE CODING THAT CAUSES A DEVICE
TO NO LONGER BE TESTED.

118 : 0741 !-
119 : 0742 !-

120 : 0743
121 : 0744

BGNDU;
RETURN;
ENDDU;

122 : 0745
123 : 0746

127
131 004140 000207

LDU: RTS PC ;

0737

132
133

; Routine Size: 1 word
; Maximum stack depth per invocation: 0 words

134
139
140

144
148 004142 004767 177772

L\$DU:: JSR PC,LDU ;

0745

149 004146 104453
150 004150 000207

TRAP 53
RTS PC

151
152

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

153
158
159

160 : 0747
161 :

162 : 0748 !+
163 : 0749 !

THE ADD-UNIT SECTION CONTAINS ANY CODE THE PROGRAMMER WISHES
TO BE EXECUTED IN CONJUNCTION WITH THE ADDING OF A UNIT BACK
TO THE TEST CYCLE.

164 : 0750 !-
165 : 0751 !-

166 : 0752 !-

167 :ML3
168 :

17-Oct-1980 11:31:46
29-Sep-1980 10:13:18

TOPS-20 Bliss-16 V2(206)
PA:<NEALE>ML3.BLI.2 (1)

169
170 : 0753

171 : 0754

BGNAU;
RETURN;
ENDAU;

172 : 0755
173 : 0756

177
181 004152 000207

LAU: RTS PC ;

0746

182
183

; Routine Size: 1 word
; Maximum stack depth per invocation: 0 words

184
189
190

194

198 004154 004767 177772
199 004160 104452
200 004162 000207

L\$AU:: JSR PC,LAU ;
TRAP 52
RTS PC

0755

201
202 ; Routine Size: 4 words
203 ; Maximum stack depth per invocation: 0 words
208

209
210 : 0757 END
211 : 0758
212 : 0759 ELUDOM

216
217
218
219
220 ;ML3 17-Oct-1980 11:31:46 TOPS
221 ; 29-Sep-1980 10:13:18 PA:<

222
223
224
225 : Size: 20 code + 0 data words
226 : Run Time: 00:01.9
227 : Elapsed Time: 00:07.1
228 : Memory Used: 12 pages
229 : Compilation Complete

230

.SBTTL HARDWARD TEST SECTION

22-Oct-1980 10:47:44 TOPS-20 Bliss-16 V2(206)
22-Oct-1980 10:45:32 PA:<NEALE>BL2ML4.BLI.2 (1)

```

1
6 :ML4
7 :
8
9 : 0001 MODULE ML4 =
10 : 0002 BEGIN
11 : 0003
12 : 0004 : PRETTY BLF COMMANDS
13 : 0005
14 : 0006 : <BLF/NOERROR>
15 : 0007 : <BLF/LOWERCASE_KEY>
16 : 0008
17 : 0009 : REQUIRE
18 : 0010
19 : 0011
20 : 0012 require 'BLSMAC.REQ'; !BLISS INTERFACE MODULE
21 : 1496
22 : 1497
23 : 1498 : CONSTANT LITERALS
24 : 1499
25 : 1500
26 : 1501 literal
27 : 1502 ONE = 1, !DATA BIT OF ONE
28 : 1503 ONES = %o'177777', !DATA PATTERN OF ONES
29 : 1504 ZERO = 0, !DATA BIT OF ZERO
30 : 1505 ZEROES = 0, !DATA PATTERN OF ZEROES
31 : 1506 NUM_OF_REG = 22, !NUMBER OF BLOCKS IN GLOBAL STORAGE 'ML-REG'
32 : 1507 FIELD_SIZ = 4, !FIELD SIZ FOR FIELD DECLARATIONS 'WORD_MAP'
33 : 1508
34 : 1509 : MLCS1 FUNCTION CODES
35 : 1510
36 : 1511 NOOP = 1, !NOOP FUNCTION
37 : 1512 DRV_CLR = %o'11', !DRIVE CLEAR FUNCTION
38 : 1513 RD_IN_PRE = %o'21', !READ IN PRESET FUNCTION
39 : 1514 SEARCH = %o'31', !SEARCH FUNCTION
40 : 1515 WRT_CHK = %o'51', !WRITE CHECK FUNCTION
41 : 1516 write = %o'61', !WRITE FUNCTION
42 : 1517 read = %o'71', !READ FUNCTION
43 : 1518
44 : 1519 : DELAY ARGUMENTS
45 : 1520
46 : 1521 ONE_US = 1, !ONE MICRO SECOND DELAY
47 : 1522 FRTY_US = 40, !FORTY MICRO SECOND DELAY
48 : 1523 TWO_TH_US = 2000; !TWO THOUSAND MICRO SECOND DELAY
49 : 1524
50 : 1525 :
51 : 1526 : FIELD DECLARATIONS
52 : 1527
53 : 1528
54 : 1529 field
55 : 1530 WORD_MAP = !MAPS GLOBAL STORAGE 'ML_REG' INTO REGISTER PERSONALITIES
56 : 1531 set
57 : 1532 REGISTER_ADD = [0, 0, 16, 0], !REGISTERS ADDRESS
58 : 1533 FORCE_HI = [1, 0, 16, 0], !REGISTERS FORCED HI BITS
59 : 1534 FORCE_LO = [2, 0, 16, 0], !REGISTERS FORCED LO BITS
60 : 1535 DONT_CARE = [3, 0, 16, 0] !REGISTERS IGNORE BITS

```

62 ;ML4
63 :
64 :
65 :
66 :
67 :
68 :
69 :
70 :
71 :
72 :
73 :
74 :
75 :
76 :
77 :
78 :
79 :
80 :

1536
1537
1538
1539
1540
1541
1542
1543
1544
1545
1546
1547
1548
1549
1550
1551

```
tes,  
NIB_MAP =  
set  
NIB_0 = [0, 0, 4, 0],  
NIB_1 = [0, 4, 4, 0],  
NIB_2 = [0, 8, 4, 0],  
NIB_3 = [0, 12, 4, 0],  
NIB_4 = [1, 0, 4, 0],  
NIB_5 = [1, 4, 4, 0],  
NIB_6 = [1, 8, 4, 0],  
NIB_7 = [1, 12, 4, 0],  
NIB_8 = [2, 8, 4, 0],  
NIB_9 = [2, 12, 3, 0]  
tes;
```

!<BLF/PAGE>

22-Oct-1980 10:47:44
22-Oct-1980 10:45:32

TOPS-20 Bliss-16 V2(206)
PA:<NEALE>BL2ML4.BLI.2 (1)

!MAPS OWN STORAGE NIB_SAVE INTO TEN FOUR BIT NIBBLES

- !NIBBLE 0 BITS <0:3>
- !NIBBLE 1 BITS <4:7>
- !NIBBLE 2 BITS <8:11>
- !NIBBLE 3 BITS <12:15>
- !NIBBLE 4 BITS <16:19>
- !NIBBLE 5 BITS <20:23>
- !NIBBLE 6 BITS <24:27>
- !NIBBLE 7 BITS <28:31>
- !NIBBLE 8 BITS <32:35>
- !NIBBLE 9 BITS <36:39>

82 :ML4
83 :
84 :
85 :
86 :
87 :
88 :
89 :
90 :
91 :
92 :
93 :
94 :
95 :
96 :
97 :
98 :
99 :
100 :
101 :
102 :
103 :
104 :
105 :
106 :
107 :
108 :
109 :
110 :
111 :
112 :
113 :
114 :

1552 :
1553 : OWN STORAGE
1554 :
1555 :
1556 : own
1557 : NIB_SAVE : block [3] field (NIB_MAP) volatile,
1558 :
1559 : HW OR TBL : vector [127] volatile,
1560 : PTBL_PTR : volatile,
1561 : OP_NUM_ARR : volatile,
1562 : ARR_INC : volatile,
1563 : GOOD_BLK : volatile,
1564 : PAR_DIS : volatile,
1565 : CHIP_SIZ : volatile,
1566 : LST_BLK : volatile,
1567 : ARR_16 : volatile,
1568 : LST_ARR : volatile,
1569 : IO_BUF : vector [256] volatile,
1570 : STK_OFF : vector [9, byte] volatile,
1571 : stack : vector [198, byte] volatile,
1572 : PD_TEMP : bitvector [16] volatile,
1573 : W_C_SIZE : volatile,
1574 : RAS_INC : volatile,
1575 : WT_DATA : volatile,
1576 : RD_DATA : volatile,
1577 : DRIVE_TYPE : volatile,
1578 : REG_INIT_FLG;
1579 :
1580 : EQUALS;
1581 : <BLF/PAGE>

22-Oct-1980 10:47:44
22-Oct-1980 10:45:32

TOPS-20 Bliss-16 V2(206)
PA:<NEALE>BL2ML4.BLI.2 (2)

! STORAGE LOCATION TO SAVE NIBBLE DATA READ DURING DIAG MODE
! STORES HARDWARE ORED PROM DATA DURING PROM OR FUNC TEST
! HARDWARE P-TABLE POINTER
! OPERATORS NUMBER OF ARRAY INPUTTED
! ARRAY SELECTION INCREMENT VALUE
! GOOD BLOCK ADRS
! PARITY DISABLE FLAG
! MOS RAM CHIP SIZE
! LAST ADDRESSABLE BLOCK
! MAX NUMBER OF ARRAY ALLOWED
! LAST ADDRESSABLE ARRAY
! INPUT OUTPUT BUFFER
! STACK OFFSET STORAGE LOCATION
! STACK OF 198 BYTE LOCATIONS
! PROM DATA STORAGE LOCATION DURING DIAG MODES
! STORES WORD COUNT FOR 16K OR 64K XFERS
! ROW ADRS STROBE INCREMENT
! SAVE WRITE DATA DURING REG READ WRITE TEST
! SAVE READ DATA DURING REG READ WRITE TEST
! DRIVE TYPE STORAGE LOCATION
! FLAG TO DETECT DOING REG INIT TEST

22-Oct-1980 10:47:44
22-Oct-1980 10:45:32

TOPS-20 Bliss-16 V2(206)
PA:<NEALE>BL2ML4.BLI.2 (3)

116 :ML4
117 :
118 :
119 :
120 :
121 :
122 :
123 :
124 :
125 :
126 :
127 :
128 :
129 :
130 :
131 :
132 :
133 :
134 :
135 :
136 :
137 :
138 :
139 :
140 :
141 :
142 :
143 :
144 :
145 :
146 :
147 :
148 :
149 :
150 :
151 :
152 :
153 :
154 :
155 :
156 :
157 :
158 :
159 :
160 :
161 :
162 :
163 :
164 :
165 :
166 :
167 :
168 :
169 :
170 :

1582 :
1583 : GLOBAL STORAGE
1584 :
1585 : <BLF/NOFORMAT>
1586 :

global

THIS STRUCTURE IS LOADED DURING THE INIT CODE AND
WILL CONTAIN THE ML11 REGISTER PERSONALITIES AND BUS ADRS.

THE REGISTERS FORCED HI AND FORCED LOW BITS ARE PRELOADED
INTO THE STRUCTURE USING THE ATTRIBUTE 'PRESET'.

ML_REG: blockvector [NUM_OF_REG, FIELD_SIZ] field(WORD_MAP)

```
preset (  
  [0, FORCE_HI] = %o'004000', !MLCS1  
  [0, FORCE_LO] = %o'173701',  
  [0, DONT_CARE] = %o'160200',  
  
  [5, FORCE_LO] = %o'25077', !MLDS  
  [5, FORCE_HI] = %o'010600',  
  [5, DONT_CARE] = %o'000100',  
  
  [6, FORCE_LO] = %o'014620', !MLER  
  
  [7, DONT_CARE] = %o'177400', !MLAS  
  
  [8, FORCE_LO] = %o'100000', !MLPA  
  
  [10, FORCE_LO] = %o'000020', !MLMR  
  [10, DONT_CARE] = %o'177400',  
  
  [11, FORCE_HI] = %o'000110', !MLDT  
  [11, FORCE_LO] = %o'177666',  
  [11, DONT_CARE] = %o'000001',  
  
  [13, FORCE_LO] = %o'140300', !MLE1  
  
  [14, FORCE_LO] = %o'100300', !MLE2  
  
  [17, FORCE_LO] = %o'010000', !MLEE  
  [21, DONT_CARE] = %o'000000', !MLCS2  
  ) volatile,
```

```
RH_ADD, !RH CONTROLLER BASE ADDRESS  
RH_TYP, !RH CONTROLLER TYPE  
RH_VEC, !RH CONTROLLER VECTOR ADDRESS  
ML_LUN, !ML LOGICAL UNIT NO.  
ML_DUT, !ML DRIVE NUMBER
```

1633 : <BLF/FORMAT>

22-Oct-1980 10:47:44 TOPS-20 Bliss-16 V2(206)
 22-Oct-1980 10:45:32 PA:<NEALE>BL2ML4.BLI.2 (3)

172 :ML4
 173 :
 174 :
 175 :
 176 :
 177 :
 178 :
 179 :
 180 :
 181 :
 182 :
 183 :
 184 :
 185 :
 186 :
 187 :
 188 :
 189 :
 190 :
 191 :
 192 :
 193 :
 194 :
 195 :
 196 :
 197 :
 198 :
 199 :
 200 :
 201 :
 202 :
 203 :
 204 :
 205 :
 206 :
 207 :
 208 :
 209 :
 210 :
 211 :
 212 :
 213 :
 214 :
 215 :
 216 :
 217 :
 218 :
 219 :
 220 :
 221 :
 222 :
 223 :
 224 :
 225 :
 226 :

```

1634 :
1635 : MACRO DEFINITIONS
1636 :
1637 :
1638 : macro
1639 :
1640 : REGISTER NAMES:
1641 :
1642 :     MLCS1 =
1643 :     .ML_REG [0,REGISTER_ADD]%,
1644 :     MLWC =
1645 :     .ML_REG [1,REGISTER_ADD]%,
1646 :     MLBA =
1647 :     .ML_REG [2,REGISTER_ADD]%,
1648 :     MLDA =
1649 :     .ML_REG [3,REGISTER_ADD]%,
1650 :     MLCS2 =
1651 :     .ML_REG [4,REGISTER_ADD]%,
1652 :     MLDS =
1653 :     .ML_REG [5,REGISTER_ADD]%,
1654 :     MLER =
1655 :     .ML_REG [6,REGISTER_ADD]%,
1656 :     MLAS =
1657 :     .ML_REG [7,REGISTER_ADD]%,
1658 :     MLLA =
1659 :     .ML_REG [8,REGISTER_ADD]%,
1660 :     MLPA =
1661 :     .ML_REG [8,REGISTER_ADD]%,
1662 :     MLDB =
1663 :     .ML_REG [9,REGISTER_ADD]%,
1664 :     MLMR =
1665 :     .ML_REG [10,REGISTER_ADD]%,
1666 :     MLDT =
1667 :     .ML_REG [11,REGISTER_ADD]%,
1668 :     MLSN =
1669 :     .ML_REG [12,REGISTER_ADD]%,
1670 :     MLE1 =
1671 :     .ML_REG [13,REGISTER_ADD]%,
1672 :     MLE2 =
1673 :     .ML_REG [14,REGISTER_ADD]%,
1674 :     MLD1 =
1675 :     .ML_REG [15,REGISTER_ADD]%,
1676 :     MLD2 =
1677 :     .ML_REG [16,REGISTER_ADD]%,
1678 :     MLEE =
1679 :     .ML_REG [17,REGISTER_ADD]%,
1680 :     MLEL =
1681 :     .ML_REG [18,REGISTER_ADD]%,
1682 :     MLPD =
1683 :     .ML_REG [19,REGISTER_ADD]%,
1684 :     MLBAE =
1685 :     .ML_REG [20,REGISTER_ADD]%,
  
```

```

!CONTROL AND STATUS REGISTER 1
!WORD COUNT REGISTER
!UNIBUS ADDRESS REGISTER
!DESIRED ADDRESS REGISTER
!CONTROL AND STATUS REGISTER 2
!DRIVE STATUS REGISTER
!ERROR REGISTER
!ATTENTION SUMMARY REGISTER
!LOOK AHEAD REGISTER
!PROM ADDRESS REGISTER
!DATA BUFFER REGISTER
!MAINTENANCE REGISTER
!DRIVE TYPE REGISTER
!SERIAL NUMBER REGISTER
!ECC CRC WORD REGISTER 1
!ECC CRC WORD REGISTER 2
!DATA DIAGNOSTIC REGISTER 1
!DATA DIAGNOSTIC REGISTER 2
!ECC ERROR REGISTER
!ECC ERROR LCOATION REGISTER
!PROM DATA REGISTER
!BUS ADDRESS EXTENSION REGISTER
  
```

22-Oct-1980 10:47:44 TOPS-20 Bliss-16 V2(206)
22-Oct-1980 10:45:32 PA:<NEALE>BL2ML4.BLI.2 (3)

```
228 :ML4
229 :
230 :
231 : M 1686      MLCS3 =
232 :      1687      .ML_REG [21,REGISTER_ADD]%,
233 :      1688      :
234 :      1689      : BIT ASSIGNMENTS:
235 :      1690      :
236 : M 1691      SC =
237 :      1692      (MLCS1)<15,1>%,
238 : M 1693      TRE =
239 :      1694      (MLCS1)<14,1>%,
240 : M 1695      MCPE =
241 :      1696      (MLCS1)<13,1>%,
242 : M 1697      DVA =
243 :      1698      (MLCS1)<11,1>%,
244 : M 1699      RDY =
245 :      1700      (MLCS1)<7,1>%,
246 : M 1701      IE =
247 :      1702      (MLCS1)<6,1>%,
248 : M 1703      GO =
249 :      1704      (MLCS1)<0,1>%,
250 : M 1705      ML_FUNC =
251 :      1706      (MLCS1)<0,6>%,
252 : M 1707      DLT =
253 :      1708      (MLCS2)<15,1>%,
254 : M 1709      WCE =
255 :      1710      (MLCS2)<14,1>%,
256 : M 1711      PE =
257 :      1712      (MLCS2)<13,1>%,
258 : M 1713      NED =
259 :      1714      (MLCS2)<12,1>%,
260 : M 1715      NEM =
261 :      1716      (MLCS2)<11,1>%,
262 : M 1717      PGE =
263 :      1718      (MLCS2)<10,1>%,
264 : M 1719      MXF =
265 :      1720      (MLCS2)<9,1>%,
266 : M 1721      MDPE =
267 :      1722      (MLCS2)<8,1>%,
268 : M 1723      ORDY =
269 :      1724      (MLCS2)<7,1>%,
270 : M 1725      IRDY =
271 :      1726      (MLCS2)<6,1>%,
272 : M 1727      CLR =
273 :      1728      (MLCS2)<5,1>%,
274 : M 1729      PAT =
275 :      1730      (MLCS2)<4,1>%,
276 : M 1731      BAI =
277 :      1732      (MLCS2)<3,1>%,
278 : M 1733      DRV_NUM =
279 :      1734      (MLCS2)<0,3>%,
280 : M 1735      ATTN =
281 :      1736      (MLDS)<15,1>%,
282 : M 1737      COMP_ERR =
```

!CONTROL AND STATUS REGISTER 3

!MLCS1 BIT ASSIGNMENTS

!MLCS2 BIT ASSIGNMENTS

!MLDS BIT ASSIGNMENTS

284 :ML4

285 :

286 :

287 : 1738 (MLDS)<14,1>%,

288 : M 1739 MOL =

289 : 1740 (MLDS)<12,1>%,

290 : M 1741 LBT =

291 : 1742 (MLDS)<10,1>%,

292 : M 1743 DPR =

293 : 1744 (MLDS)<8,1>%,

294 : M 1745 DRY =

295 : 1746 (MLDS)<7,1>%,

296 : M 1747 VV =

297 : 1748 (MLDS)<6,1>%,

298 : M 1749 DCK =

299 : 1750 (MLER)<15,1>%,

300 : M 1751 UNS =

301 : 1752 (MLER)<14,1>%,

302 : M 1753 OPI =

303 : 1754 (MLER)<13,1>%,

304 : M 1755 IAE =

305 : 1756 (MLER)<10,1>%,

306 : M 1757 AOE =

307 : 1758 (MLER)<9,1>%,

308 : M 1759 ECH =

309 : 1760 (MLER)<6,1>%,

310 : M 1761 DPAR =

311 : 1762 (MLER)<5,1>%,

312 : M 1763 CPAR =

313 : 1764 (MLER)<3,1>%,

314 : M 1765 RMR =

315 : 1766 (MLER)<2,1>%,

316 : M 1767 ILR =

317 : 1768 (MLER)<1,1>%,

318 : M 1769 ILF =

319 : 1770 (MLER)<0,1>%,

320 : M 1771 ARR_TYP =

321 : 1772 (MLMR)<10,1>%,

322 : M 1773 ML_NUM_ARR =

323 : 1774 (MLMR)<11,5>%,

324 : M 1775 REF_MAR =

325 : 1776 (MLMR)<7,1>%,

326 : M 1777 PROM_RW =

327 : 1778 (MLMR)<6,1>%,

328 : M 1779 PROM_DIS =

329 : 1780 (MLMR)<5,1>%,

330 : M 1781 DAT_CLK =

331 : 1782 (MLMR)<4,1>%,

332 : M 1783 DAT_DM =

333 : 1784 (MLMR)<3,1>%,

334 : M 1785 DCK_EN =

335 : 1786 (MLMR)<2,1>%,

336 : M 1787 ECC_DIS =

337 : 1788 (MLMR)<1,1>%,

338 : M 1789 ECC_DM =

22-Oct-1980 10:47:44
22-Oct-1980 10:45:32

TOPS-20 Bliss-16 v2(206)
PA:<NEALE>BL2ML4.BLI.2 (3)

!MLER BIT ASSIGNMENTS

!MLMR BIT ASSIGNMENTS

22-Oct-1980 10:47:44 TOPS-20 Bliss-16 V2(206)
22-Oct-1980 10:45:32 PA:<NEALE>BL2ML4.BLI.2 (3)

```

340 ;ML4
341 ;
342 ;
343 : 1790 (MLMR)<0,1>%,
344 : M 1791 DRV_TYP =
345 : 1792 (MLDT)<0,1>%, !MLDT BIT ASSIGNMENTS
346 : M 1793 UNC_ERR =
347 : 1794 (MLEE)<15,1>%, !MLEE BIT ASSIGNMENTS
348 : M 1795 SGL_ERR =
349 : 1796 (MLEE)<14,1>%,
350 : M 1797 CRC_ERR =
351 : 1798 (MLEE)<13,1>%,
352 : 1799
353 : 1800 ! MISCELLANEOUS MACRO DEFINITIONS:
354 : 1801 !
355 : M 1802 IS_SET =
356 : M 1803 !TEST IF BIT IS EQUAL 1
357 : 1804 eql 1%,
358 : M 1805 IS_NOT_SET =
359 : M 1806 !TEST IF BIT IS EQUAL 0
360 : 1807 eql 0%,
361 : M 1808 REG_ADRS =
362 : M 1809 !READS REGISTERS ADDRESS FROM PERSONALITY TABLE
363 : 1810 .ML_REG[.index,REGISTER_ADD]%,
364 : M 1811 HI =
365 : M 1812 !READS REGISTERS FORCED HI BITS FROM PERSONALITY TABLE
366 : 1813 ML_REG[.index,FORCE_HI]%,
367 : M 1814 LO =
368 : M 1815 !READS REGISTERS FORCED LO BITS FROM PERSONALITY TABLE
369 : 1816 ML_REG[.index,FORCE_LO]%,
370 : M 1817 IGNORE =
371 : M 1818 !READS REGISTERS DONT_CARE BITS FROM PERSONALITY TABLE
372 : 1819 ML_REG[.index,DONT_CARE]%,
373 : M 1820 MLE2_MASK =
374 : M 1821 !READS MLE2 DONT CARE MASK EITHER DATA DIAG OR ECC CIE REG
375 : 1822 ML_REG[14,DONT_CARE]%,
376 : M 1823 WRT_MASK =
377 : M 1824 !GENERATE MASK DATA PATTERN USING REGISTER FORCE LO, HI AND IGNORE B
378 : 1825 .IGNORE or ((not .LO) and (.HI or .TST_PAT))%,
379 : M 1826 CLR_MBUS =
380 : M 1827 !CLEAR MASS BUS RESTORE DRIVE NUMBER
381 : 1828 CLR = ONE; DRV_NUM = .ML_DUT%,
382 : 1829 !<BLF/SYNONYM IS_SET = EQL 1 * >
383 : 1830 !<BLF/SYNONYM IS_NOT_SET = EQL 0 * >
384 : 1831
385 : 1832 ! DIAGNOSTIC DATA REGISTER MACROS
386 : 1833 !
387 : M 1834 RD_LNG_WRD =
388 : M 1835 !READ DATA DIAG REGS INTO BIND LOCATIONS
389 : M 1836 D1_TEMP = .MLD1;
390 : M 1837 D2_TEMP = .MLD2;
391 : 1838 E2_TEMP = .MLE2%,
392 : M 1839 WRT_LNG_WRD =
393 : M 1840 !LOADS DATA DIAG REG WITH CONTENTS OF BIND LOCATIONS
394 : M 1841 MLD1 = .D1_TEMP;

```


22-Oct-1980 10:47:44
22-Oct-1980 10:45:32

TOPS-20 Bliss-16 V2(206)
PA:<NEALE>BL2ML4.BLI.2 (3)

396 :ML4
397 :
398 :
399 :
400 :
401 :
402 :
403 :ML4
404 :
405 :

M 1842 MLD2 = .D2_TEMP;
1843 MLE2 = .E2_TEMP%;
1844
1845 !<BLF/PAGE>

22-Oct-1980 10:47:44
22-Oct-1980 10:45:32

TOPS-20 Bliss-16 V2(206)
PA:<NEALE>BL2ML4.BLI.2 (4)

406 : 1846
407 : 1847
408 : 1848
409 : 1849
410 : 1850
411 : 1851
412 : 1852
413 : 1853
414 : 1854
415 : 1855
416 : 1856
417 : 1857
418 : 1858
419 : 1859
420 : 1860
421 : 1861
422 : 1862
423 : 1863
424 : 1864
425 : 1865
426 : 1866
427 : 1867
428 : 1868
429 : 1869
430 : 1870
431 : 1871
432 : 1872
433 : 1873
434 : 1874
435 : 1875
436 : 1876
437 : 1877
438 : 1878
439 : 1879
440 : 1880
441 : 1881
442 : 1882
443 : 1883
444 : 1884
445 : 1885
446 : 1886
447 : 1887
448 : 1888
449 : 1889
450 : 1890
451 : 1891
452 : 1892

! BIND DECLARATIONS

bind

! ERROR DATA MAPPING FORMATS

FMT_1 = uplit (%asciz'%AWROTE: %06%A READ: %06%N%N'),
FMT_2 = uplit (%asciz'%AGOOD DATA: %06%A BAD DATA: %06%A XOR: %06%N%N'),
FMT_3 = uplit (%asciz'%ADRIVE SN: %06%N%N'),
FMT_4 = uplit (%asciz'%ABIT IN ERROR: %06%N%N'),
FMT_5 = uplit (%asciz'%AGOOD NIB DATA: %02%A BAD NIB DATA: %02%A NIB POS: %04%N%N'),
FMT_6 = uplit (%asciz'%ANIB IN ERROR: %D4%N%N'),
FMT_7 = uplit (%asciz'%AFAILED AT: %06%N%N'),
FMT_8 = uplit (%asciz'%AREPLACE ARR MOD: %D2%N%N'),
FMT_9 = uplit (%asciz'%AFAILED AT DSA: %06%N%N'),
FMT_10 = uplit (%asciz'%ABIT<15:10>: %B6%A BIT<9:0>: %B10%N%N'),
FMT_11 = uplit (%asciz'%AFAILING REG ADRS: %06%N%N'),
FMT_12 = uplit (%asciz'%AFAILING FUNC: %06%N%N'),
FMT_13 = uplit (%asciz'%AOFF SET CNT FOR NIB : %D2 %A = %D2 %N%N'),
FMT_14 = uplit (%asciz'%AWROTE: %D2%A READ: %D2%N%N'),
FMT_15 = uplit (%asciz'%ANIBBLES XFERED BEFORE ERROR: %D3%N'),
FMT_16 = uplit (%asciz'%AFAILING REG: %06%A GOOD DATA: %06%A BAD DATA: %06%N%N'),
FMT_17 = uplit (%asciz'%N%ADIAGNOSING UNIT %01%N%N'),
FMT_18 = uplit (%asciz'%ATIMED OUT DURING MBUS %02%A FUNC%N%N'),

! ERROR MESSAGE MAPPING FORMATS

ONE_FMT = uplit (%asciz'%T%N'),
TWO_FMT = uplit (%asciz'%T%T%N'),
THR_FMT = uplit (%asciz'%T%T%T%N'),
FOR_FMT = uplit (%asciz'%T%T%T%T%N'),
FIV_FMT = uplit (%asciz'%T%T%T%T%T%N'),
SIX_FMT = uplit (%asciz'%T%T%T%T%T%T%N'),
SEV_FMT = uplit (%asciz'%T%T%T%T%T%T%T%N'),
EIG_FMT = uplit (%asciz'%T%T%T%T%T%T%T%T%N'),
NIN_FMT = uplit (%asciz'%T%T%T%T%T%T%T%T%T%N'),
TEN_FMT = uplit (%asciz'%T%T%T%T%T%T%T%T%T%T%N'),
ELV_FMT = uplit (%asciz'%T%T%T%T%T%T%T%T%T%T%T%N'),

! DIAGNOSTIC VOCABULARY

! WORDS

WRD_1 = uplit (%asciz' GQ'),

453 :	1893	WRD_2 = uplit (%asciz' DRV_RDY'),
454 :	1894	WRD_3 = uplit (%asciz' ILF'),
455 :	1895	WRD_4 = uplit (%asciz' OPI'),
456 :	1896	WRD_5 = uplit (%asciz' BAD'),
457 :	1897	WRD_6 = uplit (%asciz' GOOD'),

22-Oct-1980 10:47:44
22-Oct-1980 10:45:32

TOPS-20 Bliss-16 V2(206)
PA:<NEALE>BL2ML4.BLI.2 (4)

```
459 :ML4
460 :
461 :
462 : 1898 WRD_7 = uplit (%asciz' PARITY NOT'),
463 : 1899 WRD_8 = uplit (%asciz' GENERATED'),
464 : 1900 WRD_9 = uplit (%asciz' DETECTED'),
465 : 1901 WRD_10 = uplit (%asciz' ERRORS'),
466 : 1902 WRD_11 = uplit (%asciz' AFTER'),
467 : 1903 WRD_12 = uplit (%asciz' DURING'),
468 : 1904 WRD_13 = uplit (%asciz' AT'),
469 : 1905 WRD_14 = uplit (%asciz' FAILURE'),
470 : 1906 WRD_15 = uplit (%asciz' ATA'),
471 : 1907 WRD_16 = uplit (%asciz' ATTN'),
472 : 1908 WRD_17 = uplit (%asciz' WRITING'),
473 : 1909 WRD_18 = uplit (%asciz' VV'),
474 : 1910 WRD_19 = uplit (%asciz' FUNC'),
475 : 1911 WRD_20 = uplit (%asciz' TRE'),
476 : 1912 WRD_21 = uplit (%asciz' RMR'),
477 : 1913 WRD_22 = uplit (%asciz' EXCESSIVE'),
478 : 1914 WRD_23 = uplit (%asciz' MBUS'),
479 : 1915 WRD_24 = uplit (%asciz' DATA'),
480 : 1916 WRD_25 = uplit (%asciz' CONTINUITY'),
481 : 1917 WRD_26 = uplit (%asciz' AOE'),
482 : 1918 WRD_27 = uplit (%asciz' LBT'),
483 : 1919 WRD_29 = uplit (%asciz' PREMATURLY'),
484 : 1920 WRD_30 = uplit (%asciz' IAE'),
485 : 1921 WRD_31 = uplit (%asciz' INCREMENT'),
486 : 1922 WRD_32 = uplit (%asciz' WITH'),
487 : 1923 WRD_33 = uplit (%asciz' UV'),
488 : 1924 WRD_34 = uplit (%asciz' UNS'),
489 : 1925 WRD_35 = uplit (%asciz' PROM'),
490 : 1926 WRD_36 = uplit (%asciz' OR'),
491 : 1927 WRD_37 = uplit (%asciz' SELECT'),
492 : 1928 WRD_38 = uplit (%asciz' REG'),
493 : 1929 WRD_39 = uplit (%asciz' UNIQUE'),
494 : 1930 WRD_40 = uplit (%asciz' 14'),
495 : 1931 WRD_41 = uplit (%asciz' NIBBLE CNT'),
496 : 1932 WRD_42 = uplit (%asciz' GTR'),
497 : 1933 WRD_43 = uplit (%asciz' WHILE'),
498 : 1934 WRD_44 = uplit (%asciz' TRE'),
499 : 1935 WRD_45 = uplit (%asciz' INITIAL'),
500 : 1936 WRD_46 = uplit (%asciz' OFF SET'),
501 : 1937 WRD_47 = uplit (%asciz' COUNT'),
502 : 1938 WRD_48 = uplit (%asciz' DELAY'),
503 : 1939 WRD_49 = uplit (%asciz' TESTS'),
504 : 1940 WRD_50 = uplit (%asciz' ADRS'),
505 : 1941 WRD_51 = uplit (%asciz' COUNTER'),
506 : 1942 WRD_52 = uplit (%asciz' REG'),
507 : 1943 WRD_53 = uplit (%asciz' TESTED'),
508 : 1944 WRD_54 = uplit (%asciz' NIBBLE'),
509 : 1945 WRD_55 = uplit (%asciz' ALL'),
510 : 1946 WRD_56 = uplit (%asciz' TEST'),
511 : 1947 WRD_57 = uplit (%asciz' XFERED'),
512 : 1948 WRD_58 = uplit (%asciz' NIBBLES'),
513 : 1949 WRD_59 = uplit (%asciz' SC'),
```

22-Oct-1980 10:47:44
22-Oct-1980 10:45:32

TOPS-20 Bliss-16 V2(206)
PA:<NEALE>BL2ML4.BLI.2 (4)

515 :ML4
516 :
517 :
518 :
519 :
520 :
521 :
522 :
523 :
524 :
525 :
526 :
527 :
528 :
529 :
530 :
531 :
532 :
533 :
534 :
535 :
536 :
537 :
538 :
539 :
540 :
541 :
542 :
543 :
544 :
545 :
546 :
547 :
548 :
549 :
550 :
551 :
552 :
553 :
554 :
555 :
556 :
557 :
558 :
559 :
560 :
561 :
562 :
563 :
564 :
565 :
566 :
567 :
568 :
569 :

1950
1951
1952
1953
1954
1955
1956
1957
1958
1959
1960
1961
1962
1963
1964
1965
1966
1967
1968
1969
1970
1971
1972
1973
1974
1975
1976
1977
1978
1979
1980
1981
1982
1983
1984
1985
1986
1987
1988
1989
1990
1991
1992
1993
1994
1995
1996
1997
1998
1999
2000
2001

WRD_60 = uplit (%asciz' MULTIPLEXER'),
WRD_61 = uplit (%asciz' UNEXPECTED'),

PHRASES

PHR_1 = uplit (%asciz' BIT NOT SET'),
PHR_2 = uplit (%asciz' BIT NOT CLR'),
PHR_3 = uplit (%asciz' NO RESPONCE AFTER 1.5 US'),
PHR_4 = uplit (%asciz' DATA ERRORS'),
PHR_5 = uplit (%asciz' BIT SET'),
PHR_6 = uplit (%asciz' BIT CLR'),
PHR_7 = uplit (%asciz' OF OTHER DRIVES'),
PHR_8 = uplit (%asciz' CLASS A'),
PHR_9 = uplit (%asciz' CLASS B'),
PHR_10 = uplit (%asciz' TO FIND'),

FUNCTIONS

FNC_1 = uplit (%asciz' MEM SIZING'),
FNC_2 = uplit (%asciz' NOOP'),
FNC_3 = uplit (%asciz' DRV'),
FNC_4 = uplit (%asciz' WRITE CHECK'),
FNC_5 = uplit (%asciz' WRITE'),
FNC_6 = uplit (%asciz' READ'),
FNC_7 = uplit (%asciz' CLEAR'),
FNC_8 = uplit (%asciz' COMP ERROR'),
FNC_9 = uplit (%asciz' SYS CLR'),
FNC_10 = uplit (%asciz' SEARCH'),
FNC_11 = uplit (%asciz' READ-IN-PRESET'),
FNC_12 = uplit (%asciz' ILLEGAL'),
FNC_13 = uplit (%asciz' ABORT'),
FNC_14 = uplit (%asciz' ARR RD WRT'),
FNC_15 = uplit (%asciz' GOOD BLK'),
FNC_16 = uplit (%asciz' REFRESH'),
FNC_17 = uplit (%asciz' ARRAY'),
FNC_18 = uplit (%asciz' RAM-BUS'),
FNC_19 = uplit (%asciz' OVERFLOW'),
FNC_21 = uplit (%asciz' CHK_SUM'),
FNC_22 = uplit (%asciz' LAST BLK'),
FNC_23 = uplit (%asciz' INITIALIZE'),

REGISTERS

REG_1 = uplit (%asciz' MLCS1'),
REG_2 = uplit (%asciz' MLDS'),
REG_3 = uplit (%asciz' MLER'),
REG_4 = uplit (%asciz' MLMR'),
REG_5 = uplit (%asciz' MLAS'),
REG_6 = uplit (%asciz' MLDA'),
REG_7 = uplit (%asciz' MLDT'),
REG_8 = uplit (%asciz' MLPA'),
REG_9 = uplit (%asciz' MLSN'),

22-Oct-1980 10:47:44
22-Oct-1980 10:45:32

TOPS-20 Bliss-16 V2(206)
PA:<NEALE>BL2ML4.BLI.2 (4)

```
571 :ML4
572 :
573 :
574 : 2002 REG_10 = uplit (%asciz' MLE1'),
575 : 2003 REG_11 = uplit (%asciz' MLE2'),
576 : 2004 REG_12 = uplit (%asciz' MLD1'),
577 : 2005 REG_13 = uplit (%asciz' MLD2'),
578 : 2006 REG_14 = uplit (%asciz' MLEE'),
579 : 2007 REG_15 = uplit (%asciz' MLEL'),
580 : 2008 REG_16 = uplit (%asciz' MLPD'),
581 : 2009
582 : 2010 ! MODULES IN ERROR MESSAGES
583 : 2011
584 : 2012 ASYNC = uplit (%asciz'ASYNCHRONOUS MODULE FAILURE'),
585 : 2013 SYNC = uplit (%asciz'SYNCHRONOUS MODULE FAILURE'),
586 : 2014 ARR_DAT = uplit (%asciz'ARRAY DATA MODULE FAILURE'),
587 : 2015 MEM_ARR = uplit (%asciz'MEMORY ARRAY MODULE FAILURE'),
588 : 2016 INTER = uplit (%asciz'INTERMEDIATE DIAGNOSTIC MESSAGE'),
589 : 2017 TRBLE_LOOP = uplit (%asciz'TROUBLE SHOOT LOOP ERRORS'),
590 : 2018
591 : 2019 ! DATA DIAGNOSTIC REGISTER SAVE LOCATIONS
592 : 2020
593 : 2021 D1_TEMP = NIB_SAVE,
594 : 2022 D2_TEMP = NIB_SAVE [1, 0, 16, 0],
595 : 2023 E2_TEMP = NIR_SAVE [2, 0, 16, 0];
596 : 2024
597 : 2025 !<BLF/PAGE>
```

22-Oct-1980 10:47:44
22-Oct-1980 10:45:32

TOPS-20 Bliss-16 v2(206)
PA:<NEALE>BL2ML4.BLI.2 (6)

599 :ML4
600 :
601 :
602 :
603 :
604 :
605 :
606 :
607 :
608 :
609 :
610 :
611 :
612 :
613 :
614 :
615 :
616 :
617 :
618 :
619 :
620 :
621 :
622 :
623 :
624 :
625 :
626 :
627 :
628 :
629 :
630 :
631 :
632 :
633 :
634 :
635 :
636 :
637 :
638 :
639 :
640 :
641 :
642 :
643 :
644 :
645 :
646 :
647 :
648 :
649 :
650 :
651 :
652 :
653 :

```
2026 routine LOAD_STACK (STK_PTR, NIB_PTR) : novalue =
2027   begin
2028
2029   !++
2030   FUNCTIONAL DESCRIPTION:
2031     LOAD STACK TAKES GOOD NIBBLE DATA
2032     FOUND IN THE STRUCTURE 'NIB_SAVE'
2033     AND STORES IT INTO THE STRUCTURE
2034     'STACK' REWRITING ANY BAD NIBBLE
2035     'STACK' LOCATIONS WITH GOOD NIBBLE
2036     DATA
2037
2038   FORMAL PARAMETERS:
2039   STK_PTR
2040     POINTS TO PRESENT DEPTH OF THE
2041     'STACK' WHERE PRESENT GOOD NIBBLE
2042     DATA IS TO BE STORED.
2043
2044   NIB_PTR
2045     POINTS TO CURRENT NIBBLE POSITION BEING
2046     MANIPULATED.
2047
2048   IMPLICIT INPUTS:
2049   STACK
2050     VECTOR OF 198 BYTE LOCATIONS WHERE
2051     GOOD NIBBLE DATA IS STORED
2052     DURING DIAGNOSTIC MODE READS, AFTER
2053     BAD NIBBLE LOCATIONS HAVE BEEN
2054     STRIPPED AWAY.
2055
2056   STK_OFF
2057     vector of 9 byte LOCATIONS WHICH
2058     STORES AWAY A BAD NIBBLE OFF SET
2059     COUNT FOR EACH NIBBLE POSITION
2060
2061   NIB_SAVE
2062     BLOCK OF 3 WORDS TO STORE THE
2063     DATA FOUND IN MLD1, MLD2 AND
2064     MLE2 AFTER A DIAGNOSTIC MODE READ.
2065
2066   IMPLICIT OUTPUTS:
2067     'STACK' LOADED WITH GOOD NIBBLE
2068     DATA
2069
2070   COMPLETETION CODES:   NONE
2071
2072   SIDE EFFECTS:        NONE
2073
2074   --
2075
2076   case .NIB_PTR from 0 to 8 of
```

!SELECT NIBBLE DATA TO BE LOADED INTO THE STACK

22-Oct-1980 10:47:44 TOPS-20 Bliss-16 V2(206)
22-Oct-1980 10:45:32 PA:<NEALE>BL2ML4.BLI.2 (6)

```
655 :ML4
656 :
657 :
658 :      2078      set
659 :      2079
660 :      2080      [0] :
661 :      2081          stack [(.STK_PTR - (.STK_OFF [.NIB_PTR]))] = .NIB_SAVE [NIB 0];
662 :      2082          !LOAD NIBBLE DATA 0 INTO SELECTED STACK LOCATION
663 :      2083
664 :      2084      [1] :
665 :      2085          stack [(.STK_PTR - (.STK_OFF [.NIB_PTR]))] = .NIB_SAVE [NIB 1];
666 :      2086          !LOAD NIBBLE DATA 1 INTO SELECTED STACK LOCATION
667 :      2087
668 :      2088      [2] :
669 :      2089          stack [(.STK_PTR - (.STK_OFF [.NIB_PTR]))] = .NIB_SAVE [NIB 2];
670 :      2090          !LOAD NIBBLE DATA 2 INTO SELECTED STACK LOCATION
671 :      2091
672 :      2092      [3] :
673 :      2093          stack [(.STK_PTR - (.STK_OFF [.NIB_PTR]))] = .NIB_SAVE [NIB 3];
674 :      2094          !LOAD NIBBLE DATA 3 INTO SELECTED STACK LOCATION
675 :      2095
676 :      2096      [4] :
677 :      2097          stack [(.STK_PTR - (.STK_OFF [.NIB_PTR]))] = .NIB_SAVE [NIB 4];
678 :      2098          !LOAD NIBBLE DATA 4 INTO SELECTED STACK LOCATION
679 :      2099
680 :      2100      [5] :
681 :      2101          stack [(.STK_PTR - (.STK_OFF [.NIB_PTR]))] = .NIB_SAVE [NIB 5];
682 :      2102          !LOAD NIBBLE DATA 5 INTO SELECTED STACK LOCATION
683 :      2103
684 :      2104      [6] :
685 :      2105          stack [(.STK_PTR - (.STK_OFF [.NIB_PTR]))] = .NIB_SAVE [NIB 6];
686 :      2106          !LOAD NIBBLE DATA 6 INTO SELECTED STACK LOCATION
687 :      2107
688 :      2108      [7] :
689 :      2109          stack [(.STK_PTR - (.STK_OFF [.NIB_PTR]))] = .NIB_SAVE [NIB 7];
690 :      2110          !LOAD NIBBLE DATA 7 INTO SELECTED STACK LOCATION
691 :      2111
692 :      2112      [8] :
693 :      2113          stack [(.STK_PTR - (.STK_OFF [.NIB_PTR]))] = .NIB_SAVE [NIB 8];
694 :      2114          !LOAD NIBBLE DATA 8 INTO SELECTED STACK LOCATION
695 :      2115      tes;
696 :      2116
697 :      2117      end;
```

704	004164	045	101	127	P.AAA:	.ASCII	/%AW/
705	004167	122	117	124		.ASCII	/RCT/
706	004172	105	072	040		.ASCII	/E: /
707	004175	045	117	066		.ASCII	/%06/

```

709      :ML4
710      :
711
712 004200      045      101      040      .ASCII  /%A /
713 004203      040      040      040      .ASCII  / /
714 004206      122      105      101      .ASCII  /REA/
715 004211      104      072      040      .ASCII  /D: /
716 004214      045      117      066      .ASCII  /%06/
717 004217      045      116      045      .ASCII  /%N%/
718 004222      116      000      .ASCII  /N/<00>
719 004224      045      101      107      P.AAB: .ASCII  /%AG/
720 004227      117      117      104      .ASCII  /OOD/
721 004232      040      104      101      .ASCII  / DA/
722 004235      124      101      072      .ASCII  /TA:/
723 004240      040      045      117      .ASCII  / %O/
724 004243      066      045      101      .ASCII  /6%A/
725 004246      040      040      040      .ASCII  / /
726 004251      040      102      101      .ASCII  / BA/
727 004254      104      040      104      .ASCII  /D D/
728 004257      101      124      101      .ASCII  /ATA/
729 004262      072      040      045      .ASCII  /: %/
730 004265      117      066      045      .ASCII  /06%/
731 004270      101      040      040      .ASCII  /A /
732 004273      040      040      130      .ASCII  / X/
733 004276      117      122      072      .ASCII  /OR:/
734 004301      040      045      117      .ASCII  / %O/
735 004304      066      045      116      .ASCII  /6%N/
736 004307      045      116      000      .ASCII  /%N/<00>
737 004312      045      101      104      P.AAC: .ASCII  /%AD/
738 004315      122      111      126      .ASCII  /RIV/
739 004320      105      040      123      .ASCII  /E S/
740 004323      116      072      040      .ASCII  /N: /
741 004326      045      117      066      .ASCII  /%06/
742 004331      045      116      045      .ASCII  /%N%/
743 004334      116      000      .ASCII  /N/<00>
744 004336      045      101      102      P.AAD: .ASCII  /%AB/
745 004341      111      124      040      .ASCII  /IT /
746 004344      111      116      040      .ASCII  /IN /
747 004347      105      122      122      .ASCII  /ERR/
748 004352      117      122      072      .ASCII  /OR:/
749 004355      040      045      117      .ASCII  / %O/
750 004360      066      045      116      .ASCII  /6%N/
751 004363      045      116      000      .ASCII  /%N/<00>
752 004366      045      101      107      P.AAE: .ASCII  /%AG/
753 004371      117      117      104      .ASCII  /OOD/
754 004374      040      116      111      .ASCII  / NI/
755 004377      102      040      104      .ASCII  /B D/
756 004402      101      124      101      .ASCII  /ATA/
757 004405      072      040      045      .ASCII  /: %/
758 004410      117      062      045      .ASCII  /02%/
759 004413      101      040      040      .ASCII  /A /
760 004416      040      040      102      .ASCII  / B/
761 004421      101      104      040      .ASCII  /AD /
762 004424      116      111      102      .ASCII  /NIB/
763 004427      040      104      101      .ASCII  / DA/

```



```

765          :ML4
766          :
767
768 004432    124    101    072    .ASCII /TA:/
769 004435    040    045    117    .ASCII / %O/
770 004440    062    045    101    .ASCII /2%A/
771 004443    040    040    040    .ASCII / /
772 004446    040    116    111    .ASCII / NI/
773 004451    102    040    120    .ASCII /B P/
774 004454    117    123    072    .ASCII /OS:/
775 004457    040    045    117    .ASCII / %O/
776 004462    064    045    116    .ASCII /4%N/
777 004465    045    116    000    .ASCII /%N/<00>
778 004470    045    101    116    P.AAF: .ASCII /%AN/
779 004473    111    102    040    .ASCII /IB /
780 004476    111    116    040    .ASCII /IN /
781 004501    105    122    122    .ASCII /ERR/
782 004504    117    122    072    .ASCII /OR:/
783 004507    040    045    104    .ASCII / %D/
784 004512    064    045    116    .ASCII /4%N/
785 004515    045    116    000    .ASCII /%N/<00>
786 004520    045    101    106    P.AAG: .ASCII /%AF/
787 004523    101    111    114    .ASCII /AIL/
788 004526    105    104    040    .ASCII /ED /
789 004531    101    124    072    .ASCII /AT:/
790 004534    040    045    117    .ASCII / %O/
791 004537    066    045    116    .ASCII /6%N/
792 004542    045    116    000    .ASCII /%N/<00>
793 004545    000    .ASCII <00>
794 004546    045    101    122    P.AAH: .ASCII /%AR/
795 004551    105    120    114    .ASCII /EPL/
796 004554    101    103    105    .ASCII /ACE/
797 004557    040    101    122    .ASCII / AR/
798 004562    122    040    115    .ASCII /R M/
799 004565    117    104    072    .ASCII /OD:/
800 004570    040    045    104    .ASCII / %D/
801 004573    062    045    116    .ASCII /2%N/
802 004576    045    116    000    .ASCII /%N/<00>
803 004601    000    .ASCII <00>
804 004602    045    101    106    P.AAI: .ASCII /%AF/
805 004605    101    111    114    .ASCII /AIL/
806 004610    105    104    040    .ASCII /ED /
807 004613    101    124    040    .ASCII /AT /
808 004616    104    123    101    .ASCII /DSA/
809 004621    072    040    045    .ASCII /: %/
810 004624    117    066    045    .ASCII /06%/
811 004627    116    045    116    .ASCII /N%N/
812 004632    000    000    .ASCII <0C><00>
813 004634    045    101    102    P.AAJ: .ASCII /%AB/
814 004637    111    124    074    .ASCII /IT</
815 004642    061    065    072    .ASCII /15:/
816 004645    061    060    076    .ASCII /10>/
817 004650    072    040    045    .ASCII /: %/
818 004653    102    066    045    .ASCII /B6%/
819 004656    101    040    040    .ASCII /A /

```

```
821      ;ML4
822      ;
823
824 004661 040 040 102      .ASCII / B/
825 004664 111 124 074      .ASCII /IT</
826 004667 071 072 060      .ASCII /9:0/
827 004672 076 072 040      .ASCII />: /
828 004675 045 102 061      .ASCII /%B1/
829 004700 060 045 116      .ASCII /0%N/
830 004703 045 116 000      .ASCII /%N/<00>
831 004706 045 101 106 P.AAK: .ASCII /%AF/
832 004711 101 111 114      .ASCII /AIL/
833 004714 111 116 107      .ASCII /ING/
834 004717 040 122 105      .ASCII / RE/
835 004722 107 040 101      .ASCII /G A/
836 004725 104 122 123      .ASCII /DRS/
837 004730 072 040 045      .ASCII /: %/
838 004733 117 066 045      .ASCII /06%/
839 004736 116 045 116      .ASCII /N%N/
840 004741 000      .ASCII <00>
841 004742 045 101 106 P.AAL: .ASCII /%AF/
842 004745 101 111 114      .ASCII /AIL/
843 004750 111 116 107      .ASCII /ING/
844 004753 040 106 125      .ASCII / FU/
845 004756 116 103 072      .ASCII /NC:/
846 004761 040 045 117      .ASCII / %O/
847 004764 066 045 116      .ASCII /6%N/
848 004767 045 116 000      .ASCII /%N/<00>
849 004772 040 045 101 P.AAM: .ASCII / %A/
850 004775 117 106 106      .ASCII /OFF/
851 005000 137 123 105      .ASCII / SE/
852 005003 124 040 103      .ASCII /T C/
853 005006 116 124 040      .ASCII /NT /
854 005011 106 117 122      .ASCII /FOR/
855 005014 040 116 111      .ASCII / NI/
856 005017 102 040 072      .ASCII /B :/
857 005022 040 045 104      .ASCII / %D/
858 005025 062 040 045      .ASCII /2 %/
859 005030 101 040 075      .ASCII /A =/
860 005033 040 045 104      .ASCII / %D/
861 005036 062 040 045      .ASCII /2 %/
862 005041 116 045 116      .ASCII /N%N/
863 005044 000 000      .ASCII <00><00>
864 005046 045 101 127 P.AAN: .ASCII /%AW/
865 005051 122 117 124      .ASCII /ROT/
866 005054 105 072 040      .ASCII /E: /
867 005057 045 104 062      .ASCII /%D2/
868 005062 045 101 040      .ASCII /%A /
869 005065 040 040 040      .ASCII / /
870 005070 122 105 101      .ASCII /REA/
871 005073 104 072 040      .ASCII /D: /
872 005076 045 104 062      .ASCII /%D2/
873 005101 045 116 045      .ASCII /%N%/
874 005104 116 000      .ASCII /N/<00>
875 005106 045 101 116 P.AAO: .ASCII /%AN/
```

22-Oct-1980 10:47:44 TOPS
22-Oct-1980 10:45:32 PA:<

```
877      :ML4
878      :
879
880 005111 111 102 102 .ASCII /IBB/
881 005114 114 105 123 .ASCII /LES/
882 005117 040 130 106 .ASCII /XF/
883 005122 105 122 105 .ASCII /ERE/
884 005125 104 040 102 .ASCII /D B/
885 005130 105 106 117 .ASCII /EFO/
886 005133 122 105 040 .ASCII /RE /
887 005136 105 122 122 .ASCII /ERR/
888 005141 117 122 072 .ASCII /OR:/
889 005144 040 045 104 .ASCII /%D/
890 005147 063 045 116 .ASCII /3%N/
891 005152 000 000 .ASCII <00><00>
892 005154 045 101 106 P.AAP: .ASCII /%AF/
893 005157 101 111 114 .ASCII /AIL/
894 005162 111 116 107 .ASCII /ING/
895 005165 040 122 105 .ASCII /RE/
896 005170 107 072 040 .ASCII /G: /
897 005173 045 117 066 .ASCII /%06/
898 005176 045 101 040 .ASCII /%A /
899 005201 107 117 117 .ASCII /GOO/
900 005204 104 040 104 .ASCII /D D/
901 005207 101 124 101 .ASCII /ATA/
902 005212 072 040 045 .ASCII /: %/
903 005215 117 066 045 .ASCII /06%/
904 005220 101 040 102 .ASCII /A B/
905 005223 101 104 040 .ASCII /AD /
906 005226 104 101 124 .ASCII /DAT/
907 005231 101 072 040 .ASCII /A: /
908 005234 045 117 066 .ASCII /%06/
909 005237 045 116 045 .ASCII /%N%/
910 005242 116 000 .ASCII /N/<00>
911 005244 045 116 045 P.AAQ: .ASCII /%N%/
912 005247 101 104 111 .ASCII /ADI/
913 005252 101 107 116 .ASCII /AGN/
914 005255 117 123 111 .ASCII /OSI/
915 005260 116 107 040 .ASCII /NG /
916 005263 125 116 111 .ASCII /UNI/
917 005266 124 040 045 .ASCII /T %/
918 005271 117 061 045 .ASCII /01%/
919 005274 116 045 116 .ASCII /N%N/
920 005277 000 .ASCII <00>
921 005300 045 101 124 P.AAR: .ASCII /%AT/
922 005303 111 115 105 .ASCII /IME/
923 005306 104 040 117 .ASCII /D O/
924 005311 125 124 040 .ASCII /UT /
925 005314 104 125 122 .ASCII /DUR/
926 005317 111 116 107 .ASCII /ING/
927 005322 040 115 102 .ASCII / MB/
928 005325 125 123 040 .ASCII /US /
929 005330 045 117 062 .ASCII /%02/
930 005333 045 101 040 .ASCII /%A /
931 005336 106 125 116 .ASCII /FUN/
```

22-Oct-1980 10:47:44 TOPS
22-Oct-1980 10:45:32 PA:<

```
933          :ML4
934          :
935
936 005341    103    045    116          .ASCII /C%N/
937 005344    045    116    000          .ASCII /%N/<00>
938 005347    000          .ASCII <00>
939 005350    045    124    045 P.AAS: .ASCII /%T%/
940 005353    116    000    000          .ASCII /N/<00><00>
941 005356    045    124    045 P.AAT: .ASCII /%T%/
942 005361    124    045    116          .ASCII /T%N/
943 005364    000    000          .ASCII <00><00>
944 005366    045    124    045 P.AAU: .ASCII /%T%/
945 005371    124    045    124          .ASCII /T%T/
946 005374    045    116    000          .ASCII /%N/<00>
947 005377    000          .ASCII <00>
948 005400    045    124    045 P.AAV: .ASCII /%T%/
949 005403    124    045    124          .ASCII /T%T/
950 005406    045    124    045          .ASCII /%T%/
951 005411    116    000    000          .ASCII /N/<00><00>
952 005414    045    124    045 P.AAW: .ASCII /%T%/
953 005417    124    045    124          .ASCII /T%T/
954 005422    045    124    045          .ASCII /%T%/
955 005425    124    045    116          .ASCII /T%N/
956 005430    000    000          .ASCII <00><00>
957 005432    045    124    045 P.AAX: .ASCII /%T%/
958 005435    124    045    124          .ASCII /T%T/
959 005440    045    124    045          .ASCII /%T%/
960 005443    124    045    124          .ASCII /T%T/
961 005446    045    116    000          .ASCII /%N/<00>
962 005451    000          .ASCII <00>
963 005452    045    124    045 P.AAY: .ASCII /%T%/
964 005455    124    045    124          .ASCII /T%T/
965 005460    045    124    045          .ASCII /%T%/
966 005463    124    045    124          .ASCII /T%T/
967 005466    045    124    045          .ASCII /%T%/
968 005471    116    000    000          .ASCII /N/<00><00>
969 005474    045    124    045 P.AAZ: .ASCII /%T%/
970 005477    124    045    124          .ASCII /T%T/
971 005502    045    124    045          .ASCII /%T%/
972 005505    124    045    124          .ASCII /T%T/
973 005510    045    124    045          .ASCII /%T%/
974 005513    124    045    116          .ASCII /T%N/
975 005516    000    000          .ASCII <00><00>
976 005520    045    124    045 P.ABA: .ASCII /%T%/
977 005523    124    045    124          .ASCII /T%T/
978 005526    045    124    045          .ASCII /%T%/
979 005531    124    045    124          .ASCII /T%T/
980 005534    045    124    045          .ASCII /%T%/
981 005537    124    045    124          .ASCII /T%T/
982 005542    045    116    000          .ASCII /%N/<00>
983 005545    000          .ASCII <00>
984 005546    045    124    045 P.ABB: .ASCII /%T%/
985 005551    124    045    124          .ASCII /T%T/
986 005554    045    124    045          .ASCII /%T%/
987 005557    124    045    124          .ASCII /T%T/
```

```

989          :ML4
990          :
991
992 005562    045    124    045          .ASCII  /%T%/
993 005565    124    045    124          .ASCII  /T%T/
994 005570    045    124    045          .ASCII  /%T%/
995 005573    116    000    000          .ASCII  /N/<00><00>
996 005576    045    124    045 P.ABC:   .ASCII  /%T%/
997 005601    124    045    124          .ASCII  /T%T/
998 005604    045    124    045          .ASCII  /%T%/
999 005607    124    045    124          .ASCII  /T%T/
1000 005612   045    124    045          .ASCII  /%T%/
1001 005615   124    045    124          .ASCII  /T%T/
1002 005620   045    124    045          .ASCII  /%T%/
1003 005623   124    045    116          .ASCII  /T%N/
1004 005626   000    000          .ASCII  <00><00>
1005 005630   040    107    117 P.ABD:   .ASCII  / GO/
1006 005633   000          .ASCII  <00>
1007 005634   040    104    122 P.ABE:   .ASCII  / DR/
1008 005637   126    137    122          .ASCII  /V R/
1009 005642   104    131    000          .ASCII  /D%/<00>
1010 005645   000          .ASCII  <00>
1011 005646   040    111    114 P.ABF:   .ASCII  / IL/
1012 005651   106    000    000          .ASCII  /F/<00><00>
1013 005654   040    117    120 P.ABG:   .ASCII  / OP/
1014 005657   111    000    000          .ASCII  /I/<00><00>
1015 005662   040    102    101 P.ABH:   .ASCII  / BA/
1016 005665   104    000    000          .ASCII  /D/<00><00>
1017 005670   040    107    117 P.ABI:   .ASCII  / GO/
1018 005673   117    104    000          .ASCII  /OD/<00>
1019 005676   040    120    101 P.ABJ:   .ASCII  / PA/
1020 005701   122    111    124          .ASCII  /RIT/
1021 005704   131    040    116          .ASCII  /Y N/
1022 005707   117    124    000          .ASCII  /OT/<00>
1023 005712   040    107    105 P.ABK:   .ASCII  / GE/
1024 005715   116    105    122          .ASCII  /NER/
1025 005720   101    124    105          .ASCII  /ATE/
1026 005723   104    000    000          .ASCII  /D/<00><00>
1027 005726   040    104    105 P.ABL:   .ASCII  / DE/
1028 005731   124    105    103          .ASCII  /TEC/
1029 005734   124    105    104          .ASCII  /TED/
1030 005737   000          .ASCII  <00>
1031 005740   040    105    122 P.ABM:   .ASCII  / ER/
1032 005743   122    117    122          .ASCII  /ROR/
1033 005746   123    000          .ASCII  /S/<00>
1034 005750   040    101    106 P.ABN:   .ASCII  / AF/
1035 005753   124    105    122          .ASCII  /TER/
1036 005756   000    000          .ASCII  <00><00>
1037 005760   040    104    125 P.ABO:   .ASCII  / DU/
1038 005763   122    111    116          .ASCII  /RIN/
1039 005766   107    000          .ASCII  /G/<00>
1040 005770   040    101    124 P.ABP:   .ASCII  / AT/
1041 005773   000          .ASCII  <00>
1042 005774   040    106    101 P.ABQ:   .ASCII  / FA/
1043 005777   111    114    125          .ASCII  /ILU/
  
```

```

1045      ;ML4
1046      ;
1047
1048 006002 122 105 000 .ASCII /RE/<00>
1049 006005 000 .ASCII <00>
1050 006006 040 101 124 P.ABR: .ASCII / AT/
1051 006011 101 000 000 .ASCII /A/<00><00>
1052 006014 040 101 124 P.ABS: .ASCII / AT/
1053 006017 124 116 000 .ASCII /TN/<00>
1054 006022 040 127 122 P.ABT: .ASCII / WR/
1055 006025 111 124 111 .ASCII /ITI/
1056 006030 116 107 000 .ASCII /NG/<00>
1057 006033 000 .ASCII <00>
1058 006034 040 126 126 P.ABU: .ASCII / VV/
1059 006037 000 .ASCII <00>
1060 006040 040 106 125 P.ABV: .ASCII / FU/
1061 006043 116 103 000 .ASCII /NC/<00>
1062 006046 040 124 122 P.ABW: .ASCII / TR/
1063 006051 105 000 000 .ASCII /E/<00><00>
1064 006054 040 122 115 P.ABX: .ASCII / RM/
1065 006057 122 000 000 .ASCII /R/<00><00>
1066 006062 040 105 130 P.ABY: .ASCII / EX/
1067 006065 103 105 123 .ASCII /CES/
1068 006070 123 111 126 .ASCII /SIV/
1069 006073 105 000 000 .ASCII /E/<00><00>
1070 006076 040 115 102 P.ABZ: .ASCII / MB/
1071 006101 125 123 000 .ASCII /US/<00>
1072 006104 040 104 101 P.ACA: .ASCII / DA/
1073 006107 124 101 000 .ASCII /TA/<00>
1074 006112 040 103 117 P.ACB: .ASCII / CO/
1075 006115 116 124 111 .ASCII /NTI/
1076 006120 116 125 111 .ASCII /NUI/
1077 006123 124 131 000 .ASCII /TY/<00>
1078 006126 040 101 117 P.ACC: .ASCII / AO/
1079 006131 105 000 000 .ASCII /E/<00><00>
1080 006134 040 114 102 P.ACD: .ASCII / LB/
1081 006137 124 000 000 .ASCII /T/<00><00>
1082 006142 040 120 122 P.ACE: .ASCII / PR/
1083 006145 105 115 101 .ASCII /EMA/
1084 006150 124 125 122 .ASCII /TUR/
1085 006153 114 131 000 .ASCII /LY/<00>
1086 006156 040 111 101 P.ACF: .ASCII / IA/
1087 006161 105 000 000 .ASCII /E/<00><00>
1088 006164 040 111 116 P.ACG: .ASCII / IN/
1089 006167 103 122 105 .ASCII /CRE/
1090 006172 115 105 116 .ASCII /MEN/
1091 006175 124 000 000 .ASCII /T/<00><00>
1092 006200 040 127 111 P.ACH: .ASCII / WI/
1093 006203 124 110 000 .ASCII /TH/<00>
1094 006206 040 125 126 P.ACI: .ASCII / UV/
1095 006211 000 .ASCII <00>
1096 006212 040 125 116 P.ACJ: .ASCII / UN/
1097 006215 123 000 000 .ASCII /S/<00><00>
1098 006220 040 120 122 P.ACK: .ASCII / PR/
1099 006223 117 115 000 .ASCII /OM/<00>

```

```

1101          ;ML4
1102          ;
1103
1104 006226   040   117   122 P.ACL:  .ASCII / OR/
1105 006231   000                .ASCII <00>
1106 006232   040   123   105 P.ACM:  .ASCII / SE/
1107 006235   114   105   103     .ASCII /LEC/
1108 006240   124   000     .ASCII /T/<00>
1109 006242   040   122   105 P.ACN:  .ASCII / RE/
1110 006245   107   000   000     .ASCII /G/<00><00>
1111 006250   040   125   116 P.ACO:  .ASCII / UN/
1112 006253   111   121   125     .ASCII /IQU/
1113 006256   105   000     .ASCII /E/<00>
1114 006260   040   061   064 P.ACP:  .ASCII / 14/
1115 006263   000                .ASCII <00>
1116 006264   040   116   111 P.ACQ:  .ASCII / NI/
1117 006267   102   102   114     .ASCII /BBL/
1118 006272   105   040   103     .ASCII /E C/
1119 006275   116   124   000     .ASCII /NT/<00>
1120 006300   040   107   124 P.ACR:  .ASCII / GT/
1121 006303   122   000   000     .ASCII /R/<00><00>
1122 006306   040   127   110 P.ACS:  .ASCII / WH/
1123 006311   111   114   105     .ASCII /ILE/
1124 006314   000   000     .ASCII <00><00>
1125 006316   040   124   122 P.ACT:  .ASCII / TR/
1126 006321   105   000   000     .ASCII /E/<00><00>
1127 006324   040   111   116 P.ACU:  .ASCII / IN/
1128 006327   111   124   111     .ASCII /ITI/
1129 006332   101   114   000     .ASCII /AL/<00>
1130 006335   000                .ASCII <00>
1131 006336   040   117   106 P.ACV:  .ASCII / OF/
1132 006341   106   137   123     .ASCII /F S/
1133 006344   105   124   000     .ASCII /ET/<00>
1134 006347   000                .ASCII <00>
1135 006350   040   103   117 P.ACW:  .ASCII / CO/
1136 006353   125   116   124     .ASCII /UNT/
1137 006356   000   000     .ASCII <00><00>
1138 006360   040   104   105 P.ACX:  .ASCII / DE/
1139 006363   114   101   131     .ASCII /LAY/
1140 006366   000   000     .ASCII <00><00>
1141 006370   040   124   105 P.ACY:  .ASCII / TE/
1142 006373   123   124   123     .ASCII /STS/
1143 006376   000   000     .ASCII <00><00>
1144 006400   040   101   104 P.ACZ:  .ASCII / AD/
1145 006403   122   123   000     .ASCII /RS/<00>
1146 006406   040   103   117 P.ADA:  .ASCII / CO/
1147 006411   125   116   124     .ASCII /UNT/
1148 006414   105   122   000     .ASCII /ER/<00>
1149 006417   000                .ASCII <00>
1150 006420   040   122   105 P.ADB:  .ASCII / RE/
1151 006423   107   000   000     .ASCII /G/<00><00>
1152 006426   040   124   105 P.ADC:  .ASCII / TE/
1153 006431   123   124   105     .ASCII /STE/
1154 006434   104   000     .ASCII /D/<00>
1155 006436   040   116   111 P.ADD:  .ASCII / NI/
  
```

```

1157          ;ML4
1158          ;
1159
1160 006441    102    102    114      .ASCII /BBL/
1161 006444    105    000          .ASCII /E/<00>
1162 006446    040    101    114 P.ADE: .ASCII / AL/
1163 006451    114    000    000      .ASCII /L/<00><00>
1164 006454    040    124    105 P.ADF: .ASCII / TE/
1165 006457    123    124    000      .ASCII /ST/<00>
1166 006462    040    130    106 P.ADG: .ASCII / XF/
1167 006465    105    122    105      .ASCII /ERE/
1168 006470    104    000          .ASCII /D/<00>
1169 006472    040    116    111 P.ADH: .ASCII / NI/
1170 006475    102    102    114      .ASCII /BBL/
1171 006500    105    123    000      .ASCII /ES/<00>
1172 006503    000          .ASCII <00>
1173 006504    040    123    103 P.ADI: .ASCII / SC/
1174 006507    000          .ASCII <00>
1175 006510    040    115    125 P.ADJ: .ASCII / MU/
1176 006513    114    124    111      .ASCII /LTI/
1177 006516    120    114    105      .ASCII /PLE/
1178 006521    130    105    122      .ASCII /XER/
1179 006524    000    000          .ASCII <00><00>
1180 006526    040    125    116 P.ADK: .ASCII / UN/
1181 006531    105    130    120      .ASCII /EXP/
1182 006534    105    103    124      .ASCII /ECT/
1183 006537    105    104    000      .ASCII /ED/<00>
1184 006542    040    102    111 P.ADL: .ASCII / BI/
1185 006545    124    040    116      .ASCII /T N/
1186 006550    117    124    040      .ASCII /OT /
1187 006553    123    105    124      .ASCII /SET/
1188 006556    000    000          .ASCII <00><00>
1189 006560    040    102    111 P.ADM: .ASCII / BI/
1190 006563    124    040    116      .ASCII /T N/
1191 006566    117    124    040      .ASCII /OT /
1192 006571    103    114    122      .ASCII /CLR/
1193 006574    000    000          .ASCII <00><00>
1194 006576    040    116    117 P.ADN: .ASCII / NO/
1195 006601    040    122    105      .ASCII / RE/
1196 006604    123    120    117      .ASCII /SPO/
1197 006607    116    103    105      .ASCII /NCE/
1198 006612    040    101    106      .ASCII / AF/
1199 006615    124    105    122      .ASCII /TER/
1200 006620    040    061    056      .ASCII / 1./
1201 006623    065    040    125      .ASCII /5 U/
1202 006626    123    000          .ASCII /S/<00>
1203 006630    040    104    101 P.ADO: .ASCII / DA/
1204 006633    124    101    040      .ASCII /TA /
1205 006636    105    122    122      .ASCII /ERR/
1206 006641    117    122    123      .ASCII /ORS/
1207 006644    000    000          .ASCII <00><00>
1208 006646    040    102    111 P.ADP: .ASCII / BI/
1209 006651    124    040    123      .ASCII /T S/
1210 006654    105    124    000      .ASCII /ET/<00>
1211 006657    000          .ASCII <00>

```



```

1213          ;ML4
1214          ;
1215
1216 006660   040   102   111 P.ADQ: .ASCII / BI/
1217 006663   124   040   103 .ASCII /T C/
1218 006666   114   122   000 .ASCII /LR/<00>
1219 006671   000           .ASCII <00>
1220 006672   040   117   106 P.ADR: .ASCII / OF/
1221 006675   040   117   124 .ASCII / OT/
1222 006700   110   105   122 .ASCII /HER/
1223 006703   040   104   122 .ASCII / DR/
1224 006706   111   126   105 .ASCII /IVE/
1225 006711   123   000   000 .ASCII /S/<00><00>
1226 006714   040   103   114 P.ADS: .ASCII / CL/
1227 006717   101   123   123 .ASCII /ASS/
1228 006722   040   101   000 .ASCII / A/<00>
1229 006725   000           .ASCII <00>
1230 006726   040   103   114 P.ADT: .ASCII / CL/
1231 006731   101   123   123 .ASCII /ASS/
1232 006734   040   102   000 .ASCII / B/<00>
1233 006737   000           .ASCII <00>
1234 006740   040   124   117 P.ADU: .ASCII / TO/
1235 006743   040   106   111 .ASCII / FI/
1236 006746   116   104   000 .ASCII /ND/<00>
1237 006751   000           .ASCII <00>
1238 006752   040   115   105 P.ADV: .ASCII / ME/
1239 006755   115   040   123 .ASCII /M S/
1240 006760   111   132   111 .ASCII /IZI/
1241 006763   116   107   000 .ASCII /NG/<00>
1242 006766   040   116   117 P.ADW: .ASCII / NO/
1243 006771   117   120   000 .ASCII /OP/<00>
1244 006774   040   104   122 P.ADX: .ASCII / DR/
1245 006777   126   000   000 .ASCII /V/<00><00>
1246 007002   040   127   122 P.ADY: .ASCII / WR/
1247 007005   111   124   105 .ASCII /ITE/
1248 007010   040   103   110 .ASCII / CH/
1249 007013   105   103   113 .ASCII /ECK/
1250 007016   000   000           .ASCII <00><00>
1251 007020   040   127   122 P.ADZ: .ASCII / WR/
1252 007023   111   124   105 .ASCII /ITE/
1253 007026   000   000           .ASCII <00><00>
1254 007030   040   122   105 P.AEA: .ASCII / RE/
1255 007033   101   104   000 .ASCII /AD/<00>
1256 007036   040   103   114 P.AEB: .ASCII / CL/
1257 007041   105   101   122 .ASCII /EAR/
1258 007044   000   000           .ASCII <00><00>
1259 007046   040   103   117 P.AEC: .ASCII / CO/
1260 007051   115   120   040 .ASCII /MP /
1261 007054   105   122   122 .ASCII /ERR/
1262 007057   117   122   000 .ASCII /OR/<00>
1263 007062   040   123   131 P.AED: .ASCII / SY/
1264 007065   123   040   103 .ASCII /S C/
1265 007070   114   122   000 .ASCII /LR/<00>
1266 007073   000           .ASCII <00>
1267 007074   040   123   105 P.AEE: .ASCII / SE/
  
```

```

1269          ;ML4
1270          ;
1271
1272 007077    101    122    103      .ASCII /ARC/
1273 007102    110    000          .ASCII /H/<00>
1274 007104    040    122    105 P.AEF: .ASCII / RE/
1275 007107    101    104    055      .ASCII /AD-/
1276 007112    111    116    055      .ASCII /IN-/
1277 007115    120    122    105      .ASCII /PRE/
1278 007120    123    105    124      .ASCII /SET/
1279 007123    000          .ASCII <00>
1280 007124    040    111    114 P.AEG: .ASCII / IL/
1281 007127    114    105    107      .ASCII /LEG/
1282 007132    101    114    000      .ASCII /AL/<00>
1283 007135    000          .ASCII <00>
1284 007136    040    101    102 P.AEH: .ASCII / AB/
1285 007141    117    122    124      .ASCII /ORT/
1286 007144    000    000          .ASCII <00><00>
1287 007146    040    101    122 P.AEI: .ASCII / AR/
1288 007151    122    040    122      .ASCII /R R/
1289 007154    104    137    127      .ASCII /D W/
1290 007157    122    124    000      .ASCII /RT/<00>
1291 007162    040    107    117 P.AEJ: .ASCII / GO/
1292 007165    117    104    040      .ASCII /OD /
1293 007170    102    114    113      .ASCII /BLK/
1294 007173    000          .ASCII <00>
1295 007174    040    122    105 P.AEK: .ASCII / RE/
1296 007177    106    122    105      .ASCII /FRE/
1297 007202    123    110    000      .ASCII /SH/<00>
1298 007205    000          .ASCII <00>
1299 007206    040    101    122 P.AEL: .ASCII / AR/
1300 007211    122    101    131      .ASCII /RAY/
1301 007214    000    000          .ASCII <00><00>
1302 007216    040    122    101 P.AEM: .ASCII / RA/
1303 007221    115    055    102      .ASCII /M-B/
1304 007224    125    123    000      .ASCII /US/<00>
1305 007227    000          .ASCII <00>
1306 007230    040    117    126 P.AEN: .ASCII / OV/
1307 007233    105    122    106      .ASCII /ERF/
1308 007236    114    117    127      .ASCII /LOW/
1309 007241    000          .ASCII <00>
1310 007242    040    103    110 P.AEO: .ASCII / CH/
1311 007245    113    137    123      .ASCII /K S/
1312 007250    125    115    000      .ASCII /UM/<00>
1313 007253    000          .ASCII <00>
1314 007254    040    114    101 P.AEP: .ASCII / LA/
1315 007257    123    124    040      .ASCII /ST /
1316 007262    102    114    113      .ASCII /BLK/
1317 007265    000          .ASCII <00>
1318 007266    040    111    116 P.AEQ: .ASCII / IN/
1319 007271    111    124    111      .ASCII /ITI/
1320 007274    101    114    111      .ASCII /ALI/
1321 007277    132    105    000      .ASCII /ZE/<00>
1322 007302    040    115    114 P.AER: .ASCII / ML/
1323 007305    103    123    061      .ASCII /CS1/

```

1325					:ML4		
1326					:		
1327							
1328	007310	000	000			.ASCII	<00><00>
1329	007312	040	115	114	P.AES:	.ASCII	/ ML/
1330	007315	104	123	000		.ASCII	/DS/<00>
1331	007320	040	115	114	P.AET:	.ASCII	/ ML/
1332	007323	105	122	000		.ASCII	/ER/<00>
1333	007326	040	115	114	P.AEU:	.ASCII	/ ML/
1334	007331	115	122	000		.ASCII	/MR/<00>
1335	007334	040	115	114	P.AEV:	.ASCII	/ ML/
1336	007337	101	123	000		.ASCII	/AS/<00>
1337	007342	040	115	114	P.AEW:	.ASCII	/ ML/
1338	007345	104	101	000		.ASCII	/DA/<00>
1339	007350	040	115	114	P.AEX:	.ASCII	/ ML/
1340	007353	104	124	000		.ASCII	/DT/<00>
1341	007356	040	115	114	P.AEY:	.ASCII	/ ML/
1342	007361	120	101	000		.ASCII	/PA/<00>
1343	007364	040	115	114	P.AEZ:	.ASCII	/ ML/
1344	007367	123	116	000		.ASCII	/SN/<00>
1345	007372	040	115	114	P.AFA:	.ASCII	/ ML/
1346	007375	105	061	000		.ASCII	/E1/<00>
1347	007400	040	115	114	P.AFB:	.ASCII	/ ML/
1348	007403	105	062	000		.ASCII	/E2/<00>
1349	007406	040	115	114	P.AFC:	.ASCII	/ ML/
1350	007411	104	061	000		.ASCII	/D1/<00>
1351	007414	040	115	114	P.AFD:	.ASCII	/ ML/
1352	007417	104	062	000		.ASCII	/D2/<00>
1353	007422	040	115	114	P.AFE:	.ASCII	/ ML/
1354	007425	105	105	000		.ASCII	/EE/<00>
1355	007430	040	115	114	P.AFF:	.ASCII	/ ML/
1356	007433	105	114	000		.ASCII	/EL/<00>
1357	007436	040	115	114	P.AFG:	.ASCII	/ ML/
1358	007441	120	104	000		.ASCII	/PD/<00>
1359	007444	101	123	131	P.AFH:	.ASCII	/ASY/
1360	007447	116	103	110		.ASCII	/NCH/
1361	007452	122	117	116		.ASCII	/RON/
1362	007455	117	125	123		.ASCII	/OUS/
1363	007460	040	115	117		.ASCII	/ MO/
1364	007463	104	125	114		.ASCII	/DUL/
1365	007466	105	040	106		.ASCII	/E F/
1366	007471	101	111	114		.ASCII	/AIL/
1367	007474	125	122	105		.ASCII	/URE/
1368	007477	000				.ASCII	<00>
1369	007500	123	131	116	P.AFI:	.ASCII	/SYN/
1370	007503	103	110	122		.ASCII	/CHR/
1371	007506	117	116	117		.ASCII	/ONO/
1372	007511	125	123	040		.ASCII	/US /
1373	007514	115	117	104		.ASCII	/MOD/
1374	007517	125	114	105		.ASCII	/ULE/
1375	007522	040	106	101		.ASCII	/ FA/
1376	007525	111	114	125		.ASCII	/ILU/
1377	007530	122	105	000		.ASCII	/RE/<00>
1378	007533	000				.ASCII	<00>
1379	007534	101	122	122	P.AFJ:	.ASCII	/ARR/

22-Oct-1980 10:47:44 TOPS
22-Oct-1980 10:45:32 PA:<

```

1381          ;ML4
1382          ;
1383
1384 007537    101    131    040          .ASCII /AY /
1385 007542    104    101    124          .ASCII /DAT/
1386 007545    101    040    115          .ASCII /A M/
1387 007550    117    104    125          .ASCII /ODU/
1388 007553    114    105    040          .ASCII /LE /
1389 007556    106    101    111          .ASCII /FAI/
1390 007561    114    125    122          .ASCII /LUR/
1391 007564    105    000          .ASCII /E/<00>
1392 007566    115    105    115 P.AFK: .ASCII /MEM/
1393 007571    117    122    131          .ASCII /ORY/
1394 007574    040    101    122          .ASCII / AR/
1395 007577    122    101    131          .ASCII /RAY/
1396 007602    040    115    117          .ASCII / MO/
1397 007605    104    125    114          .ASCII /DUL/
1398 007610    105    040    106          .ASCII /E F/
1399 007613    101    111    114          .ASCII /AIL/
1400 007616    125    122    105          .ASCII /URE/
1401 007621    000          .ASCII <00>
1402 007622    111    116    124 P.AFL: .ASCII /INT/
1403 007625    105    122    115          .ASCII /ERM/
1404 007630    105    104    111          .ASCII /EDI/
1405 007633    101    124    105          .ASCII /ATE/
1406 007636    040    104    111          .ASCII / DI/
1407 007641    101    107    116          .ASCII /AGN/
1408 007644    117    123    124          .ASCII /OST/
1409 007647    111    103    040          .ASCII /IC /
1410 007652    115    105    123          .ASCII /MES/
1411 007655    123    101    107          .ASCII /SAG/
1412 007660    105    000          .ASCII /E/<00>
1413 007662    124    122    117 P.AFM: .ASCII /TRO/
1414 007665    125    102    114          .ASCII /UBL/
1415 007670    105    040    123          .ASCII /E S/
1416 007673    110    117    117          .ASCII /HOO/
1417 007676    124    040    114          .ASCII /T L/
1418 007701    117    117    120          .ASCII /OOP/
1419 007704    040    105    122          .ASCII / ER/
1420 007707    122    117    122          .ASCII /ROR/
1421 007712    123    000          .ASCII /S/<00>
1422
1423
1424
1425 007714          NIB.SAVE:
1426 007714          .BLKW 3
1427 007722          HW.OR.TBL:
1428 007722          .BLKW 177
1429 010320          PTBL.PTR:
1430 010320          .BLKW 1
1431 010322          OP.NUM.ARR:
1432 010322          .BLKW 1
1433 010324          ARR.INC: .BLKW 1
1434 010326          GOOD.BLK:

```

22-Oct-1980 10:47:44 TOPS
22-Oct-1980 10:45:32 PA:<

```
1436 ;ML4
1437 ;
1438
1439 010326 .BLKW 1
1440 010330 PAR.DIS:.BLKW 1
1441 010332 CHIP.SIZ:
1442 010332 .BLKW 1
1443 010334 LST.BLK:.BLKW 1
1444 010336 ARR.16:.BLKW 1
1445 010340 LST.ARR:.BLKW 1
1446 010342 IO.BUF:.BLKW 400
1447 011342 STK.OFF:.BLKB 11
1448 .EVEN
1449 011354 STACK:.BLKW 143
1450 011662 PD.TEMP:.BLKW 1
1451 011664 W.C.SIZE:
1452 011664 .BLKW 1
1453 011666 RAS.INC:.BLKW 1
1454 011670 WT.DATA:.BLKW 1
1455 011672 RD.DATA:.BLKW 1
1456 011674 DRIVE.TYPE:
1457 011674 .BLKW 1
1458 011676 REG.INIT.FLG:
1459 011676 .BLKW 1
1460
1461
1462
1463 011700 000 ML.REG: .BYTE 0
1464 011701 000 .BYTE 0
1465 011702 004000 .WORD 4000
1466 011704 173701 .WORD -4077
1467 011706 160200 .WORD -17600
1468 011710 000 .BYTE 0
1469 011711 000 .BYTE 0
1470 011712 000 .BYTE 0
1471 011713 000 .BYTE 0
1472 011714 000 .BYTE 0
1473 011715 000 .BYTE 0
1474 011716 000 .BYTE 0
1475 011717 000 .BYTE 0
1476 011720 000 .BYTE 0
1477 011721 000 .BYTE 0
1478 011722 000 .BYTE 0
1479 011723 000 .BYTE 0
1480 011724 000 .BYTE 0
1481 011725 000 .BYTE 0
1482 011726 000 .BYTE 0
1483 011727 000 .BYTE 0
1484 011730 000 .BYTE 0
1485 011731 000 .BYTE 0
1486 011732 000 .BYTE 0
1487 011733 000 .BYTE 0
1488 011734 000 .BYTE 0
1489 011735 000 .BYTE 0
```

```
1491      :ML4
1492      :
1493
1494 011736      000      .BYTE      0
1495 011737      000      .BYTE      0
1496 011740      000      .BYTE      0
1497 011741      000      .BYTE      0
1498 011742      000      .BYTE      0
1499 011743      000      .BYTE      0
1500 011744      000      .BYTE      0
1501 011745      000      .BYTE      0
1502 011746      000      .BYTE      0
1503 011747      000      .BYTE      0
1504 011750      000      .BYTE      0
1505 011751      000      .BYTE      0
1506 011752      010600    .WORD     10600
1507 011754      025077    .WORD     25077
1508 011756      000100    .WORD      100
1509 011760      000      .BYTE      0
1510 011761      000      .BYTE      0
1511 011762      000      .BYTE      0
1512 011763      000      .BYTE      0
1513 011764      014620    .WORD    14620
1514 011766      000      .BYTE      0
1515 011767      000      .BYTE      0
1516 011770      000      .BYTE      0
1517 011771      000      .BYTE      0
1518 011772      000      .BYTE      0
1519 011773      000      .BYTE      0
1520 011774      000      .BYTE      0
1521 011775      000      .BYTE      0
1522 011776      177400    .WORD    -400
1523 012000      000      .BYTE      0
1524 012001      000      .BYTE      0
1525 012002      000      .BYTE      0
1526 012003      000      .BYTE      0
1527 012004      100000    .WORD   -100000
1528 012006      000      .BYTE      0
1529 012007      000      .BYTE      0
1530 012010      000      .BYTE      0
1531 012011      000      .BYTE      0
1532 012012      000      .BYTE      0
1533 012013      000      .BYTE      0
1534 012014      000      .BYTE      0
1535 012015      000      .BYTE      0
1536 012016      000      .BYTE      0
1537 012017      000      .BYTE      0
1538 012020      000      .BYTE      0
1539 012021      000      .BYTE      0
1540 012022      000      .BYTE      0
1541 012023      000      .BYTE      0
1542 012024      000020    .WORD      20
1543 012026      177400    .WORD    -400
1544 012030      000      .BYTE      0
1545 012031      000      .BYTE      0
```

22-Oct-1980 10:47:44 TOPS
22-Oct-1980 10:45:32 PA:<

```
1547 ;ML4
1548 ;
1549
1550 012032 000110 .WORD 110
1551 012034 177666 .WORD -112
1552 012036 000001 .WORD 1
1553 012040 000 .BYTE 0
1554 012041 000 .BYTE 0
1555 012042 000 .BYTE 0
1556 012043 000 .BYTE 0
1557 012044 000 .BYTE 0
1558 012045 000 .BYTE 0
1559 012046 000 .BYTE 0
1560 012047 000 .BYTE 0
1561 012050 000 .BYTE 0
1562 012051 000 .BYTE 0
1563 012052 000 .BYTE 0
1564 012053 000 .BYTE 0
1565 012054 140300 .WORD -37500
1566 012056 000 .BYTE 0
1567 012057 000 .BYTE 0
1568 012060 000 .BYTE 0
1569 012061 000 .BYTE 0
1570 012062 000 .BYTE 0
1571 012063 000 .BYTE 0
1572 012064 100300 .WORD -77500
1573 012066 000 .BYTE 0
1574 012067 000 .BYTE 0
1575 012070 000 .BYTE 0
1576 012071 000 .BYTE 0
1577 012072 000 .BYTE 0
1578 012073 000 .BYTE 0
1579 012074 000 .BYTE 0
1580 012075 000 .BYTE 0
1581 012076 000 .BYTE 0
1582 012077 000 .BYTE 0
1583 012100 000 .BYTE 0
1584 012101 000 .BYTE 0
1585 012102 000 .BYTE 0
1586 012103 000 .BYTE 0
1587 012104 000 .BYTE 0
1588 012105 000 .BYTE 0
1589 012106 000 .BYTE 0
1590 012107 000 .BYTE 0
1591 012110 000 .BYTE 0
1592 012111 000 .BYTE 0
1593 012112 000 .BYTE 0
1594 012113 000 .BYTE 0
1595 012114 010000 .WORD 10000
1596 012116 000 .BYTE 0
1597 012117 000 .BYTE 0
1598 012120 000 .BYTE 0
1599 012121 000 .BYTE 0
1600 012122 000 .BYTE 0
1601 012123 000 .BYTE 0
```

22-Oct-1980 10:47:44 TOPS
22-Oct-1980 10:45:32 PA:<

```
1603          :ML4
1604          :
1605
1606 012124    000          .BYTE 0
1607 012125    000          .BYTE 0
1608 012126    000          .BYTE 0
1609 012127    000          .BYTE 0
1610 012130    000          .BYTE 0
1611 012131    000          .BYTE 0
1612 012132    000          .BYTE 0
1613 012133    000          .BYTE 0
1614 012134    000          .BYTE 0
1615 012135    000          .BYTE 0
1616 012136    000          .BYTE 0
1617 012137    000          .BYTE 0
1618 012140    000          .BYTE 0
1619 012141    000          .BYTE 0
1620 012142    000          .BYTE 0
1621 012143    000          .BYTE 0
1622 012144    000          .BYTE 0
1623 012145    000          .BYTE 0
1624 012146    000          .BYTE 0
1625 012147    000          .BYTE 0
1626 012150    000          .BYTE 0
1627 012151    000          .BYTE 0
1628 012152    000          .BYTE 0
1629 012153    000          .BYTE 0
1630 012154    000          .BYTE 0
1631 012155    000          .BYTE 0
1632 012156    000000     .WORD 0
1633 012160          RH.ADD:::BLKW 1
1634 012162          RH.TYP:::BLKW 1
1635 012164          RH.VEC:::BLKW 1
1636 012166          ML.LUN:::BLKW 1
1637 012170          ML.DUT:::BLKW 1
1638
1639
1640          100000     BIT15== -100000
1641          040000     BIT14== 40000
1642          020000     BIT13== 20000
1643          010000     BIT12== 10000
1644          004000     BIT11== 4000
1645          002000     BIT10== 2000
1646          001000     BIT09== 1000
1647          000400     BIT08== 400
1648          000200     BIT07== 200
1649          000100     BIT06== 100
1650          000040     BIT05== 40
1651          000020     BIT04== 20
1652          000010     BIT03== 10
1653          000004     BIT02== 4
1654          000002     BIT01== 2
1655          000001     BIT00== 1
1656          001000     BIT9== 1000
1657          000400     BIT8== 400
```


1659		:ML4	
1660		:	
1661			
1662	000200	BIT7==	200
1663	000100	BIT6==	100
1664	000040	BIT5==	40
1665	000020	BIT4==	20
1666	000010	BIT3==	10
1667	000004	BIT2==	4
1668	000002	BIT1==	2
1669	000001	BIT0==	1
1670	000040	EF.START==	40
1671	000037	EF.RESTART==	37
1672	000036	EF.CONTINUE==	36
1673	000035	EF.NEW==	35
1674	000034	EF.PWR==	34
1675	000340	PRI07==	340
1676	000300	PRI06==	300
1677	000240	PRI05==	240
1678	000200	PRI04==	200
1679	000140	PRI03==	140
1680	000100	PRI02==	100
1681	000040	PRI01==	40
1682	000000	PRI00==	0
1683	000004	FVL==	4
1684	000010	LOT==	10
1685	000020	ADR==	20
1686	000040	IDU==	40
1687	000100	ISR==	100
1688	000200	UAM==	200
1689	000400	BOE==	400
1690	001000	PNT==	1000
1691	002000	PRI==	2000
1692	004000	IXE==	4000
1693	010000	IBE==	10000
1694	020000	IER==	20000
1695	040000	LOE==	40000
1696	100000	HOE==	-100000
1697	004164	FMT.1=	P.AAA
1698	004224	FMT.2=	P.AAB
1699	004312	FMT.3=	P.AAC
1700	004336	FMT.4=	P.AAD
1701	004366	FMT.5=	P.AAE
1702	004470	FMT.6=	P.AAF
1703	004520	FMT.7=	P.AAG
1704	004546	FMT.8=	P.AAH
1705	004602	FMT.9=	P.AAI
1706	004634	FMT.10=	P.AAJ
1707	004706	FMT.11=	P.AAK
1708	004742	FMT.12=	P.AAL
1709	004772	FMT.13=	P.AAM
1710	005046	FMT.14=	P.AAN
1711	005106	FMT.15=	P.AAO
1712	005154	FMT.16=	P.AAP
1713	005244	FMT.17=	P.AAQ

1715		;ML4	
1716		:	
1717			
1718	005300	FMT.18=	P.AAR
1719	005350	ONE.FMT=	P.AAS
1720	005356	TWO.FMT=	P.AAT
1721	005366	THR.FMT=	P.AAU
1722	005400	FOR.FMT=	P.AAV
1723	005414	FIV.FMT=	P.AAW
1724	005432	SIX.FMT=	P.AAX
1725	005452	SEV.FMT=	P.AAY
1726	005474	EIG.FMT=	P.AAZ
1727	005520	NIN.FMT=	P.ABA
1728	005546	TEN.FMT=	P.ABB
1729	005576	ELV.FMT=	P.ABC
1730	005630	WRD.1=	P.ABD
1731	005634	WRD.2=	P.ABE
1732	005646	WRD.3=	P.ABF
1733	005654	WRD.4=	P.ABG
1734	005662	WRD.5=	P.ABH
1735	005670	WRD.6=	P.ABI
1736	005676	WRD.7=	P.ABJ
1737	005712	WRD.8=	P.ABK
1738	005726	WRD.9=	P.ABL
1739	005740	WRD.10=	P.ABM
1740	005750	WRD.11=	P.ABN
1741	005760	WRD.12=	P.ABO
1742	005770	WRD.13=	P.ABP
1743	005774	WRD.14=	P.ABQ
1744	006006	WRD.15=	P.ABR
1745	006014	WRD.16=	P.ABS
1746	006022	WRD.17=	P.ABT
1747	006034	WRD.18=	P.ABU
1748	006040	WRD.19=	P.ABV
1749	006046	WRD.20=	P.ABW
1750	006054	WRD.21=	P.ABX
1751	006062	WRD.22=	P.ABY
1752	006076	WRD.23=	P.ABZ
1753	006104	WRD.24=	P.ACA
1754	006112	WRD.25=	P.ACB
1755	006126	WRD.26=	P.ACC
1756	006134	WRD.27=	P.ACD
1757	006142	WRD.29=	P.ACE
1758	006156	WRD.30=	P.ACF
1759	006164	WRD.31=	P.ACG
1760	006200	WRD.32=	P.ACH
1761	006206	WRD.33=	P.ACI
1762	006212	WRD.34=	P.ACJ
1763	006220	WRD.35=	P.ACK
1764	006226	WRD.36=	P.ACL
1765	006232	WRD.37=	P.ACM
1766	006242	WRD.38=	P.ACN
1767	006250	WRD.39=	P.ACO
1768	006260	WRD.40=	P.ACP
1769	006264	WRD.41=	P.ACQ

1771		:ML4	
1772		:	
1773			
1774	006300	WRD.42=	P.ACR
1775	006306	WRD.43=	P.ACS
1776	006316	WRD.44=	P.ACT
1777	006324	WRD.45=	P.ACU
1778	006336	WRD.46=	P.ACV
1779	006350	WRD.47=	P.ACW
1780	006360	WRD.48=	P.ACX
1781	006370	WRD.49=	P.ACY
1782	006400	WRD.50=	P.ACZ
1783	006406	WRD.51=	P.ADA
1784	006420	WRD.52=	P.ADB
1785	006426	WRD.53=	P.ADC
1786	006436	WRD.54=	P.ADD
1787	006446	WRD.55=	P.ADE
1788	006454	WRD.56=	P.ADF
1789	006462	WRD.57=	P.ADG
1790	006472	WRD.58=	P.ADH
1791	006504	WRD.59=	P.ADI
1792	006510	WRD.60=	P.ADJ
1793	006526	WRD.61=	P.ADK
1794	006542	PHR.1=	P.ADL
1795	006560	PHR.2=	P.ADM
1796	006576	PHR.3=	P.ADN
1797	006630	PHR.4=	P.ADO
1798	006646	PHR.5=	P.ADP
1799	006660	PHR.6=	P.ADQ
1800	006672	PHR.7=	P.ADR
1801	006714	PHR.8=	P.ADS
1802	006726	PHR.9=	P.ADT
1803	006740	PHR.10=	P.ADU
1804	006752	FNC.1=	P.ADV
1805	006766	FNC.2=	P.ADW
1806	006774	FNC.3=	P.ADX
1807	007002	FNC.4=	P.ADY
1808	007020	FNC.5=	P.ADZ
1809	007030	FNC.6=	P.AEA
1810	007036	FNC.7=	P.AEB
1811	007046	FNC.8=	P.AEC
1812	007062	FNC.9=	P.AED
1813	007074	FNC.10=	P.AEE
1814	007104	FNC.11=	P.AEF
1815	007124	FNC.12=	P.AEG
1816	007136	FNC.13=	P.AEH
1817	007146	FNC.14=	P.AEI
1818	007162	FNC.15=	P.AEJ
1819	007174	FNC.16=	P.AEK
1820	007206	FNC.17=	P.AEL
1821	007216	FNC.18=	P.AEM
1822	007230	FNC.19=	P.AEN
1823	007242	FNC.21=	P.AEO
1824	007254	FNC.22=	P.AEP
1825	007266	FNC.23=	P.AEQ

22-Oct-1980 10:47:44 TOPS
22-Oct-1980 10:45:32 PA:<

```

1827      ;ML4
1828      ;
1829
1830      007302      REG.1=      P.AER
1831      007312      REG.2=      P.AES
1832      007320      REG.3=      P.AET
1833      007326      REG.4=      P.AEU
1834      007334      REG.5=      P.AEV
1835      007342      REG.6=      P.AEW
1836      007350      REG.7=      P.AEX
1837      007356      REG.8=      P.AEY
1838      007364      REG.9=      P.AEZ
1839      007372      REG.10=     P.AFA
1840      007400      REG.11=     P.AFB
1841      007406      REG.12=     P.AFC
1842      007414      REG.13=     P.AFD
1843      007422      REG.14=     P.AFE
1844      007430      REG.15=     P.AFF
1845      007436      REG.16=     P.AFG
1846      007444      ASYNC=      P.AFH
1847      007500      SYNC=      P.AFI
1848      007534      ARR.DAT=   P.AFJ
1849      007566      MEM.ARR=   P.AFK
1850      007622      INTER=    P.AFL
1851      007662      TRBLE.LOOP= P.AFM
1852      007714      D1.TEMP=  NIB.SAVE
1853      007716      D2.TEMP=  NIB.SAVE+2
1854      007720      E2.TEMP=  NIB.SAVE+4
1855
1856
1857

```

```

1861 012172      LOAD.STACK:
1862 012172 004167 171624      JSR      R1,$SAVE3      ;
1863 012176 016601 000012      MOV      12(SP),R1      ; NIB.PTR,*
1864 012202 012702 011342      MOV      #STK.OFF,R2
1865 012206 060102      ADD      R1,R2
1866 012210 016601 000014      MOV      14(SP),R1      ; STK.PTR,*
1867 012214 016600 000012      MOV      12(SP),R0      ; NIB.PTR,*
1868 012220 006300
1869 012222 066007 012226      ASL      R0
1870 012226 000022      ADC      1$(R0),PC
1871 012230 000040      1$:      .WORD  2$-1$
1872 012232 000056      .WORD  3$-1$
1873 012234 000074      .WORD  4$-1$
1874 012236 000112      .WORD  5$-1$
1875 012240 000130      .WORD  6$-1$
1876 012242 000156      .WORD  7$-1$
1877 012244 000174      .WORD  8$-1$
1878 012246 000240      .WORD  9$-1$
1879 012250 005003      .WORD 10$-1$
1879 012250 005003      .WORD 14$-1$
1879 012250 005003      2$:      CLR      R3      ;

```

2026
2081
2077
2081

Line No.	Address	Label	Instruction	Comments	Line No.
1881		:ML4			
1882		:			
1883					
1884	012252	151203	BISB	(R2),R3	
1885	012254	010100	MOV	R1,R0	
1886	012256	160300	SUB	R3,R0	
1887	012260	016703	MOV	NIB.SAVE,R3	175430
1888	012264	000471	BR	13\$	
1889	012266	005003	3\$: CLR	R3	2085
1890	012270	151203	BISB	(R2),R3	
1891	012272	010100	MOV	R1,R0	
1892	012274	160300	SUB	R3,R0	
1893	012276	016703	MOV	NIB.SAVE,R3	175412
1894	012302	000433	BR	8\$	
1895	012304	005003	4\$: CLR	R3	2089
1896	012306	151203	BISB	(R2),R3	
1897	012310	010100	MOV	R1,R0	
1898	012312	160300	SUB	R3,R0	
1899	012314	016703	MOV	NIB.SAVE,R3	175374
1900	012320	000452	BR	12\$	
1901	012322	005003	5\$: CLR	R3	2093
1902	012324	151203	BISB	(R2),R3	
1903	012326	010100	MOV	R1,R0	
1904	012330	160300	SUB	R3,R0	
1905	012332	016703	MOV	NIB.SAVE,R3	175356
1906	012336	000437	BR	11\$	
1907	012340	005003	6\$: CLR	R3	2097
1908	012342	151203	BISB	(R2),R3	
1909	012344	010100	MOV	R1,R0	
1910	012346	160300	SUB	R3,R0	
1911	012350	016703	MOV	NIB.SAVE+2,R3	175342
1912	012354	000435	BR	13\$	
1913	012356	005003	7\$: CLR	R3	2101
1914	012360	151203	BISB	(R2),R3	
1915	012362	010100	MOV	R1,R0	
1916	012364	160300	SUB	R3,R0	
1917	012366	016703	MOV	NIB.SAVE+2,R3	175324
1918	012372	006203	8\$: ASR	R3	
1919	012374	006203	ASR	R3	
1920	012376	006203	ASR	R3	
1921	012400	006203	ASR	R3	
1922	012402	000422	BR	13\$	
1923	012404	005003	9\$: CLR	R3	2105
1924	012406	151203	BISB	(R2),R3	
1925	012410	010100	MOV	R1,R0	
1926	012412	160300	SUB	R3,R0	
1927	012414	016703	MOV	NIB.SAVE+2,R3	175276
1928	012420	000412	BR	12\$	
1929	012422	005003	10\$: CLR	R3	2109
1930	012424	151203	BISB	(R2),R3	
1931	012426	010100	MOV	R1,R0	
1932	012430	160300	SUB	R3,R0	
1933	012432	016703	MOV	NIB.SAVE+2,R3	175260
1934	012436	006203	11\$: ASR	R3	
1935	012440	006203	ASR	R3	

1937
1938
1939
1940 012442 006203
1941 012444 006203
1942 012446 000303
1943 012450 042703 177760
1944 012454 105060 011354
1945 012460 150360 011354
1946 012464 000207
1947 012466 005003
1948 012470 151203
1949 012472 160301
1950 012474 016703 175220
1951 012500 000303
1952 012502 042703 177760
1953 012506 105061 011354
1954 012512 150361 011354
1955 012516 000207
1956
1957
1958
1963
1964

;ML4
;

12\$:
13\$:

14\$:

ASR R3
ASR R3
SWAB R3
BIC #177760,R3
CLRB STACK(R0)
BISB R3,STACK(R0)
RTS PC
CLR R3
BISB (R2),R3
SUB R3,R1
MOV NIB.SAVE+4,R3
SWAB R3
BIC #177760,R3
CLRB STACK(R1)
BISB R3,STACK(R1)
RTS PC

22-Oct-1980 10:47:44 TOPS
22-Oct-1980 10:45:32 PA:<

2077
2113

2026

; Routine Size: 107 words
; Maximum stack depth per invocation: 4 words

22-Oct-1980 10:47:44
22-Oct-1980 10:45:32

TOPS-20 Bliss-16 V2(206)
PA:<NEALE>BL2ML4.BLI.2 (7)

1966 :ML4
1967 :
1968 :
1969 :
1970 :
1971 :
1972 :
1973 :
1974 :
1975 :
1976 :
1977 :
1978 :
1979 :
1980 :
1981 :
1982 :
1983 :
1984 :
1985 :
1986 :
1987 :
1988 :
1989 :
1993 :
1997 012520
1998 012520
1999 012524
2000 012532
2001 012540
2002 :
2003 :
2004 :
2009 :
2010 :

2118 routine FIRST_BLK_XFER : novalue =

2119
2120
2121
2122
2123
2124
2125
2126
2127
2128
2129
2130
2131
2132
2133
2134
2135
2136
2137
2138

```

!++
FUNCTIONAL DESCRIPTION:
  A REPEATEDLY CALLED SEQUENCE OF
  ASSIGNMENT EXPRESSION TO LOAD
  THE DSA, BUS ADRS AND WORD COUNT
  REGISTERS WITH APPROPRIATE INFORMATION
  BEFORE MASS BUS TRANSFERS CAN
  COMMENCE.

LOADS A MASS_BUS BLOCK XFERR AT
BLOCK ZERO.
!--

```

```

begin
MLDA = ZEROES;
MLBA = IO_BUF;
MLWC = not 255;
end;

```

```

!LOAD THE DSA REG WITH SECTOR ZERO
!LOAD THE BUS ADDRESS REG WITH IO BUF ADRS
!LOAD WORD COUNT REG WITH COMPLIMENT 256

```

FIRST.BLK.XFER:

```

CLR @ML.REG+30
MOV #IO.BUF,@ML.REG+20
MOV #-400,@ML.REG+10
RTS PC

```

2135
2136
2137
2118

```

; Routine Size: 9 words
; Maximum stack depth per invocation: 0 words

```

22-Oct-1980 10:47:44
22-Oct-1980 10:45:32

TOPS-20 Bliss-16 V2(206)
PA:<NEALE>BL2ML4.BLI.2 (8)

2012 :ML4
2013 :
2014 :
2015 :
2016 :
2017 :
2018 :
2019 :
2020 :
2021 :
2022 :
2023 :
2024 :
2025 :
2026 :
2027 :
2028 :
2029 :
2030 :
2031 :
2032 :
2033 :
2034 :
2035 :
2036 :
2037 :
2038 :
2042 :
2046 012542
2047 012542
2048 012550
2049 012556
2050 012564
2051 012572
2052 :
2053 :
2054 :
2059 :
2060 :

2139 routine GD_BLK_XFER : novalue =
2140
2141 :
2142 :
2143 :
2144 :
2145 :
2146 :
2147 :
2148 :
2149 :
2150 :
2151 :
2152 :
2153 :
2154 :
2155 :
2156 :
2157 :
2158 :
2159 :
2160 :
2161 :
2162 :

!++
FUNCTIONAL DESCRIPTION:
A REPEATEDLY CALLED SEQUENCE OF
ASSIGNMENT EXPRESSIONS TO LOAD
THE DSA, BUS ADRS AND WORD
COUNT REGISTERS WITH APPROPRIATE
INFORMATION BEFORE A MASS BUS
TRANSFERS CAN COMMENCE

LOADS A MASS BUS BLOCK XFERR
AT THE GOOD BLOCK ADRS FOUND
BY THE READ WRITE ARRAYS WITH
PROM DATA TEST

!--

begin
ECC DIS = ONE;
MLDA = .GOOD_BLK;
MLBA = IO_BUF;
MLWC = not 255;
end;

!DISABLE ERROR CORRECTION
!LOAD DSA REG WITH THE GOOD BLOCK ADRS
!LOAD BUS ADRS REG WITH IO_BUF ADRS
!LOAD WORD COUNT REG WITH COMPLIMENT 256

GD.BLK.XFER:

BISB #2,@ML.REG+120 ;
MOV GOOD.BLK,@ML.REG+30 ;
MOV #IO.BUF,@ML.REG+20 ;
MOV #-400,@ML.REG+10 ;
RTS PC ;

2158
2159
2160
2161
2139

; Routine Size: 13 words
; Maximum stack depth per invocation: 0 words

22-Oct-1980 10:47:44
22-Oct-1980 10:45:32

TOPS-20 Bliss-16 V2(206)
PA:<NEALE>BL2ML4.BLI.2 (9)

2062 :ML4
2063 :
2064 :
2065 : 2163 routine LAST_BLK_XFER : novalue =
2066 : 2164

2067 : 2165 !++
2068 : 2166 ! FUNCTIONAL DESCRIPTION:
2069 : 2167 ! A REPEATEDLY CALLED SEQUENCE OF
2070 : 2168 ! ASSIGNMENT EXPRESSIONS TO LOAD
2071 : 2169 ! THE DSA, BUS ADRS AND WORD
2072 : 2170 ! COUNT REGISTERS WITH APPROPRIATE
2073 : 2171 ! INFORMATION BEFORE A MBUS
2074 : 2172 ! TRANSFER CAN COMMENCE
2075 : 2173
2076 : 2174 ! LOADS A MASS BUS BLOCK
2077 : 2175 ! TRANSFER AT THE LAST BLOCK
2078 : 2176 ! ADDRESS
2079 : 2177 !--

2080 : 2178
2081 : 2179
2082 : 2180 begin
2083 : 2181 ECC DIS = ONE;
2084 : 2182 MLDA = .LST BLK;
2085 : 2183 MLBA = IO BUF;
2086 : 2184 MLWC = not 255;
2087 : 2185 end;

!DISABLE ERROR CORRECTION
!LOAD DSA REG WITH THE LAST BLOCK ADRS
!LOAD BUS ADRS REG WITH THE IO BUF ADRS
!LOAD WORD COUNT REG WITH COMPLIMENT 256

2091
2095 012574 LAST.BLK.XFER:
2096 012574 152777 000002 177216 BISB #2,@ML.REG+120 : 2181
2097 012602 016777 175526 177120 MOV LST.BLK,@ML.REG+30 : 2182
2098 012610 012777 010342 177102 MOV #IO.BUF,@ML.REG+20 : 2183
2099 012616 012777 177400 177064 MOV #-400,@ML.REG+10 : 2184
2100 012624 000207 RTS PC : 2163

2101
2102 ; Routine Size: 13 words
2103 ; Maximum stack depth per invocation: 0 words
2108
2109

22-Oct-1980 10:47:44
22-Oct-1980 10:45:32

TOPS-20 Bliss-16 V2(206)
PA:<NEALE>BL2ML4.BLI.2 (10)

2111 :ML4
2112 :
2113 :
2114 :
2115 :
2116 :
2117 :
2118 :
2119 :
2120 :
2121 :
2122 :
2123 :
2124 :
2125 :
2126 :
2127 :
2128 :
2129 :
2130 :
2131 :
2132 :
2133 :
2134 :
2135 :
2136 :
2137 :
2141 :
2145 012626
2146 012626
2147 012634
2148 012642
2149 012650
2150 012656
2151 :
2152 :
2153 :
2158 :
2159 :

2186 routine DAT_DM_XFER : novalue =

2187
2188
2189
2190
2191
2192
2193
2194
2195
2196
2197
2198
2199
2200
2201
2202
2203
2204
2205
2206
2207
2208
2209

!++
FUNCTIONAL DESCRIPTION:
A REPEATEDLY CALLED SEQUENCE
OF ASSIGNMENT EXPRESSIONS TO
LOAD THE DSA, BUS ADRS AND WORD
COUNT REGISTERS WITH APPROPRIATE
INFORMATION BEFORE A MASS BUS
TRANSFER CAN COMMENCE
LOADS A MASS BUS BLOCK TRANSFER,
IN DIAGNOSTIC MODE, AT THE GOOD
BLOCK ADRS.

!--

begin
DAT_DM = ONE;
MLDA = .GOOD_BLK;
MLBA = IO_BUF;
MLWC = not 255;
end;

!SET DATA DIAG MODE
!LOAD DSA REG WITH THE GOOD BLOCK ADRS
!LOAD BUS ADRS REG WITH THE IO_BUF ADRS
!LOAD WORD COUNT REG WITH COMPLIMENT 256

DAT_DM.XFER:

BISB #10,@ML.REG+120
MOV GOOD.BLK,@ML.REG+30
MOV #IO.BUF,@ML.REG+20
MOV #-400,@ML.REG+10
RTS PC

2205
2206
2207
2208
2186

; Routine Size: 13 words
; Maximum stack depth per invocation: 0 words

22-Oct-1980 10:47:44
 22-Oct-1980 10:45:32

TOPS-20 Bliss-16 V2(206)
 PA:<NEALE>BL2ML4.BLI.2 (11)

2161 :ML4
 2162 :
 2163 :
 2164 :
 2165 :
 2166 :
 2167 :
 2168 :
 2169 :
 2170 :
 2171 :
 2172 :
 2173 :
 2174 :
 2175 :
 2176 :
 2177 :
 2178 :
 2179 :
 2180 :
 2181 :
 2182 :
 2183 :
 2184 :
 2185 :
 2186 :
 2187 :
 2188 :
 2189 :
 2190 :
 2191 :
 2192 :
 2193 :
 2194 :
 2195 :
 2196 :
 2197 :
 2198 :
 2199 :
 2200 :
 2201 :
 2202 :
 2203 :
 2204 :
 2205 :
 2206 :
 2207 :
 2208 :
 2209 :
 2210 :
 2211 :
 2212 :
 2213 :
 2214 :
 2215 :

```

2210 routine TST_LNG_WRD (NIB_NUM, NIB_PAT, ERR_FLG) : novalue =
2211   begin
2212
2213   ++
2214   FUNCTIONAL DESCRIPTION:
2215   COMPARES THE CURRENT NIBBLE
2216   POSITION IN 'NIB_SAVE' WITH THE
2217   CURRENT TEST PATTERN. IF THE
2218   TWO VALUES ARE NOT EQUAL AN
2219   ERROR FLG IS SET WHICH THE
2220   CALLER CAN INTERIGATE
2221
2222   FORMAL PARAMETERS:
2223   NIB_NUM
2224   CASE SELECT EXPRESSION TO SELECT THE
2225   CURRENT NIBBLE TO BE EXAMINED
2226
2227   NIB_PAT
2228   CURRENT NIBBLE PATTERN TO BE
2229   COMPARED
2230
2231   ERR_FLG
2232   CONTAINS THE ADDRESS (PASSED BY REF)
2233   OF THE CALLERS ERROR FLG
2234   TO ENABLE THE CALLER TO EXAMINE
2235   THE ERROR STATUS OF THE ROUTINE CALL
2236
2237   IMPLICIT INPUTS:
2238   NIB_SAVE
2239   BLOCK OF 3 WORDS TO STORE
2240   THE DATA FOUND IN MLD1, MLD2
2241   AND MLE2 AFTER A DIAGNOSTIC MODE
2242   READ
2243
2244   IMPLICIT OUTPUTS:   NONE
2245   --
2246
2247   .ERR_FLG = ZERO;                                !CLEAR THE ERROR FLAG BACK IN THE CALLING ROUTINE
2248
2249   case .(NIB_NUM) from 0 to 9 of                    !SELECT THE NIBBLE TO BE TESTED
2250     set
2251
2252     [0] :
2253
2254         if .NIB_SAVE [NIB_0] neq .(NIB_PAT)<0, 4> then .ERR_FLG = ONE;
2255
2256         !TEST NIBBLE 0 AND SET ERR FLG IF NEQ
2257
2258     [1] :
2259
2260         if .NIB_SAVE [NIB_1] neq .(NIB_PAT)<0, 4> then .ERR_FLG = ONE;
2261

```

2217 :ML4
2218 :
2219 :
2220 :
2221 :
2222 :
2223 :
2224 :
2225 :
2226 :
2227 :
2228 :
2229 :
2230 :
2231 :
2232 :
2233 :
2234 :
2235 :
2236 :
2237 :
2238 :
2239 :
2240 :
2241 :
2242 :
2243 :
2244 :
2245 :
2246 :
2247 :
2248 :
2249 :
2250 :
2251 :
2252 :
2253 :
2254 :
2255 :
2256 :
2257 :
2258 :
2259 :
2260 :
2261 :
2262 :
2263 :
2264 :
2265 :
2266 :
2267 :
2268 :
2269 :
2270 :
2271 :

2262
2263
2264
2265
2266
2267
2268
2269
2270
2271
2272
2273
2274
2275
2276
2277
2278
2279
2280
2281
2282
2283
2284
2285
2286
2287
2288
2289
2290
2291
2292
2293
2294
2295
2296
2297
2298
2299
2300
2301
2302
2303
2304
2305
2306
2307
2308
2309
2310
2311
2312
2313

[2] :

if .NIB_SAVE [NIB_2] neq .(NIB_PAT)<0, 4> then .ERR_FLG = ONE;

!TEST NIBBLE 1 AND SET ERR FLG IF NEQ

!TEST NIBBLE 2 AND SET ERR FLG IF NEQ

[3] :

if .NIB_SAVE [NIB_3] neq .(NIB_PAT)<0, 4> then .ERR_FLG = ONE;

!TEST NIBBLE 3 AND SET ERR FLG IF NEQ

[4] :

if .NIB_SAVE [NIB_4] neq .(NIB_PAT)<0, 4> then .ERR_FLG = ONE;

!TEST NIBBLE 4 AND SET ERR FLG IF NEQ

[5] :

if .NIB_SAVE [NIB_5] neq .(NIB_PAT)<0, 4> then .ERR_FLG = ONE;

!TEST NIBBLE 5 AND SET ERR FLG IF NEQ

[6] :

if .NIB_SAVE [NIB_6] neq .(NIB_PAT)<0, 4> then .ERR_FLG = ONE;

!TEST NIBBLE 6 AND SET ERR FLG IF NEQ

[7] :

if .NIB_SAVE [NIB_7] neq .(NIB_PAT)<0, 4> then .ERR_FLG = ONE;

!TEST NIBBLE 7 AND SET ERR FLG IF NEQ

[8] :

if .NIB_SAVE [NIB_8] neq .(NIB_PAT)<0, 4> then .ERR_FLG = ONE;

!TEST NIBBLE 8 AND SET ERR FLG IF NEQ

[9] :

if .NIB_SAVE [NIB_9] neq .(NIB_PAT)<0, 3>
then
 .ERR_FLG = ONE

!TEST NIBBLE 9 AND SET ERR FLG IF NEQ

tes;

end;

22-Oct-1980 10:47:44 TOPS-20 Bliss-16 V2(206)
 22-Oct-1980 10:45:32 PA:<NEALE>BL2ML4.BLI.2 (11)

2273	:	ML4						
2274	:							
2275	:							
2279	:							
2283		012660			TST.LNG.WRD:			
2284		012660	004167	171122	JSR	R1,\$SAVE2		2210
2285		012664	016600	000010	MOV	10(SP),R0	: ERR.FLG,*	2247
2286		012670	005010		CLR	(R0)		
2287		012672	016601	000014	MOV	14(SP),R1	: NIB.NUM,*	2249
2288		012676	006301		ASL	R1		
2289		012700	066107	012704	ADD	1\$(R1),PC		
2290		012704	000024		1\$: .WORD	2\$-1\$		
2291		012706	000052		.WORD	3\$-1\$		
2292		012710	000110		.WORD	4\$-1\$		
2293		012712	000140		.WORD	5\$-1\$		
2294		012714	000200		.WORD	6\$-1\$		
2295		012716	000226		.WORD	7\$-1\$		
2296		012720	000264		.WORD	8\$-1\$		
2297		012722	000314		.WORD	9\$-1\$		
2298		012724	000354		.WORD	10\$-1\$		
2299		012726	000404		.WORD	11\$-1\$		
2300		012730	016602	000012	2\$: MOV	12(SP),R2	: NIB.PAT,*	2254
2301		012734	042702	177760	BIC	#177760,R2		
2302		012740	016701	174750	MOV	NIB.SAVE,R1		
2303		012744	042701	177760	BIC	#177760,R1		
2304		012750	020102		CMP	R1,R2		
2305		012752	001577		BEQ	13\$		
2306		012754	000574		BR	12\$		
2307		012756	016601	000012	3\$: MOV	12(SP),R1	: NIB.PAT,*	2260
2308		012762	042701	177760	BIC	#177760,R1		
2309		012766	016702	174722	MOV	NIB.SAVE,R2		
2310		012772	006202		ASR	R2		
2311		012774	006202		ASR	R2		
2312		012776	006202		ASR	R2		
2313		013000	006202		ASR	R2		
2314		013002	042702	177760	BIC	#177760,R2		
2315		013006	020201		CMP	R2,R1		
2316		013010	001560		BEQ	13\$		
2317		013012	000555		BR	12\$		
2318		013014	016601	000012	4\$: MOV	12(SP),R1	: NIB.PAT,*	2266
2319		013020	042701	177760	BIC	#177760,R1		
2320		013024	016702	174664	MOV	NIB.SAVE,R2		
2321		013030	000302		SWAB	R2		
2322		013032	042702	177760	BIC	#177760,R2		
2323		013036	020201		CMP	R2,R1		
2324		013040	001544		BEQ	13\$		
2325		013042	000541		BR	12\$		
2326		013044	016601	000012	5\$: MOV	12(SP),R1	: NIB.PAT,*	2272

```

2328          ;ML4
2329          ;
2330
2331 013050 042701 177760      BIC      #177760,R1
2332 013054 016702 174634      MOV      NIB.SAVE,R2
2333 013060 006202              ASR      R2
2334 013062 006202              ASR      R2
2335 013064 006202              ASR      R2
2336 013066 006202              ASR      R2
2337 013070 000302              SWAB     R2
2338 013072 042702 177760      BIC      #177760,R2
2339 013076 020201              CMP      R2,R1
2340 013100 001524              BEQ      13$
2341 013102 000521              BR       12$
2342 013104 016601 000012      6$:     MOV      12(SP),R1          ; NIB.PAT,*          2278
2343 013110 042701 177760      BIC      #177760,R1
2344 013114 016702 174576      MOV      NIB.SAVE+2,R2
2345 013120 042702 177760      BIC      #177760,R2
2346 013124 020201              CMP      R2,R1
2347 013126 001511              BEQ      13$
2348 013130 000506              BR       12$
2349 013132 016601 000012      7$:     MOV      12(SP),R1          ; NIB.PAT,*          2284
2350 013136 042701 177760      BIC      #177760,R1
2351 013142 016702 174550      MOV      NIB.SAVE+2,R2
2352 013146 006202              ASR      R2
2353 013150 006202              ASR      R2
2354 013152 006202              ASR      R2
2355 013154 006202              ASR      R2
2356 013156 042702 177760      BIC      #177760,R2
2357 013162 020201              CMP      R2,R1
2358 013164 001472              BEQ      13$
2359 013166 000467              BR       12$
2360 013170 016601 000012      8$:     MOV      12(SP),R1          ; NIB.PAT,*          2290
2361 013174 042701 177760      BIC      #177760,R1
2362 013200 016702 174512      MOV      NIB.SAVE+2,R2
2363 013204 000302              SWAB     R2
2364 013206 042702 177760      BIC      #177760,R2
2365 013212 020201              CMP      R2,R1
2366 013214 001456              BEQ      13$
2367 013216 000453              BR       12$
2368 013220 016601 000012      9$:     MOV      12(SP),R1          ; NIB.PAT,*          2296
2369 013224 042701 177760      BIC      #177760,R1
2370 013230 016702 174462      MOV      NIB.SAVE+2,R2
2371 013234 006202              ASR      R2
2372 013236 006202              ASR      R2
2373 013240 006202              ASR      R2
2374 013242 006202              ASR      R2
2375 013244 000302              SWAB     R2
2376 013246 042702 177760      BIC      #177760,R2
2377 013252 020201              CMP      R2,R1
2378 013254 001436              BEQ      13$
2379 013256 000433              BR       12$
2380 013260 016601 000012      10$:    MOV      12(SP),R1          ; NIB.PAT,*          2302
2381 013264 042701 177760      BIC      #177760,R1
2382 013270 016702 174424      MOV      NIB.SAVE+4,R2

```

```
2384 ;ML4
2385 ;
2386
2387 013274 000302 SWAB R2
2388 013276 042702 177760 BIC #177760,R2
2389 013302 020201 CMP R2,R1
2390 013304 001422 BEQ 13$
2391 013306 000417 BR 12$
2392 013310 016601 000012 11$: MOV 12(SP),R1 ; NIB.PAT,* 2308
2393 013314 042701 177770 BIC #177770,R1
2394 013320 016702 174374 MOV NIB.SAVE+4,R2
2395 013324 006202 ASR R2
2396 013326 006202 ASR R2
2397 013330 006202 ASR R2
2398 013332 006202 ASR R2
2399 013334 000302 SWAB R2
2400 013336 042702 177770 BIC #177770,R2
2401 013342 020201 CMP R2,R1
2402 013344 001402 BEQ 13$
2403 013346 012710 000001 12$: MOV #1,(R0) ; 2310
2404 013352 000207 13$: RTS PC ; 2210
2405
2406 ; Routine Size: 158 words
2407 ; Maximum stack depth per invocation: 3 words
2412
2413
```

22-Oct-1980 10:47:44
 22-Oct-1980 10:45:32

TOPS-20 Bliss-16 V2(206)
 PA:<NEALE>BL2ML4.BLI.2 (12)

2415 :ML4
 2416 :
 2417 :
 2418 :
 2419 :
 2420 :
 2421 :
 2422 :
 2423 :
 2424 :
 2425 :
 2426 :
 2427 :
 2428 :
 2429 :
 2430 :
 2431 :
 2432 :
 2433 :
 2434 :
 2435 :
 2436 :
 2437 :
 2438 :
 2439 :
 2440 :
 2441 :
 2442 :
 2443 :
 2444 :
 2445 :
 2446 :
 2447 :
 2448 :
 2449 :
 2450 :
 2451 :
 2452 :
 2453 :
 2454 :
 2455 :
 2456 :
 2457 :
 2458 :
 2459 :
 2460 :
 2461 :
 2462 :
 2463 :
 2464 :
 2465 :
 2466 :
 2467 :
 2468 :
 2469 :

```

2314 routine XOR_LNG_WRD (NIB_NUM, NIB_PAT, RESULT) : novalue =
2315     begin
2316
2317     !++
2318     FUNCTIONAL DESCRIPTION:
2319     EXCLUSIVE ORS THE CURRENT
2320     NIBBLE POSITION IN 'NIB_SAVE'
2321     WITH THE CURRENT TEST PATTERN
2322     AND ASSIGNS THE RESULTS TO THE
2323     FORMAL PARAMETER 'RESULT'.
2324
2325     FORMAL PARAMETERS:
2326     NIB_NUM
2327     CASE SELECT EXPRESSION TO
2328     SELECT THE CURRENT NIBBLE TO BE
2329     EXAMINED
2330
2331     NIB_PAT
2332     CURRENT NIBBLE PATTERN TO BE
2333     XOR'ED
2334
2335     RESULT
2336     CONTAINS THE ADDRESS (PASSED BY REF)
2337     OF AN OWN STORAGE LOCATION TO
2338     ENABLE THE CALLER TO EXAMINE THE XOR RESULTS.
2339
2340     IMPLICIT INPUTS:
2341     NIB_SAVE
2342     BLOCK OF 3 WORDS TO STORE
2343     THE DATA FOUND IN MLD1
2344     MLD2 AND MLE2 AFTER A
2345     DIAGNOSTIC MODE READ.
2346
2347     IMPLICIT OUTPUTS:  NONE
2348     --
2349
2350     case .(NIB_NUM) from 0 to 9 of
2351     set
2352
2353     [0] :
2354         .RESULT = .NIB_SAVE [NIB_0] xor .NIB_PAT;
2355         !XOR NIBBLE 0 AND STORE RESULTS IN THE ADRS CONTAINED IN .RESULTS
2356
2357     [1] :
2358         .RESULT = .NIB_SAVE [NIB_1] xor .NIB_PAT;
2359         !XOR NIBBLE 1 AND STORE RESULTS IN THE ADRS CONTAINED IN .RESULTS
2360
2361     [2] :
2362         .RESULT = .NIB_SAVE [NIB_2] xor .NIB_PAT;
2363         !XOR NIBBLE 2 AND STORE RESULTS IN THE ADRS CONTAINED IN .RESULTS
2364
2365     [3] :
  
```


2471 :ML4
2472 :
2473 :
2474 : 2366
2475 : 2367
2476 : 2368
2477 : 2369
2478 : 2370
2479 : 2371
2480 : 2372
2481 : 2373
2482 : 2374
2483 : 2375
2484 : 2376
2485 : 2377
2486 : 2378
2487 : 2379
2488 : 2380
2489 : 2381
2490 : 2382
2491 : 2383
2492 : 2384
2493 : 2385
2494 : 2386
2495 : 2387
2496 : 2388
2497 : 2389
2498 : 2390
2499 : 2391
2500 : 2392
2501 : 2393
2502 : 2394

```
.RESULT = .NIB_SAVE [NIB_3] xor .NIB_PAT;
!XOR NIBBLE 3 AND STORE RESULTS IN THE ADRS CONTAINED IN .RESULTS

[4] : .RESULT = .NIB_SAVE [NIB_4] xor .NIB_PAT;
!XOR NIBBLE 4 AND STORE RESULTS IN THE ADRS CONTAINED IN .RESULTS

[5] : .RESULT = .NIB_SAVE [NIB_5] xor .NIB_PAT;
!XOR NIBBLE 5 AND STORE RESULTS IN THE ADRS CONTAINED IN .RESULTS

[6] : .RESULT = .NIB_SAVE [NIB_6] xor .NIB_PAT;
!XOR NIBBLE 6 AND STORE RESULTS IN THE ADRS CONTAINED IN .RESULTS

[7] : .RESULT = .NIB_SAVE [NIB_7] xor .NIB_PAT;
!XOR NIBBLE 7 AND STORE RESULTS IN THE ADRS CONTAINED IN .RESULTS

[8] : .RESULT = .NIB_SAVE [NIB_8] xor .NIB_PAT;
!XOR NIBBLE 8 AND STORE RESULTS IN THE ADRS CONTAINED IN .RESULTS

[9] : .RESULT = .NIB_SAVE [NIB_9] xor .NIB_PAT;
!XOR NIBBLE 9 AND STORE RESULTS IN THE ADRS CONTAINED IN .RESULTS
```

tes;

end;

2510 013354
2511 013354 004167 170460
2512 013360 016601 000016
2513 013364 016602 000014
2514 013370 016600 000020
2515 013374 006300
2516 013376 066007 013402
2517 013402 000024
2518 013404 000036
2519 013406 000062
2520 013410 000070
2521 013412 000076
2522 013414 000120
2523 013416 000136
2524 013420 000144
2525
2526
2527
2528 013422 000162
2529 013424 000176
2530 013426 016712 174262
2531 013432 042712 177760
2532 013436 000472
2533 013440 016703 174250

XOR.LNG.WRD:

```
JSR R1,$SAVE4  
MOV 16(SP),R1  
MOV 14(SP),R2  
MOV 20(SP),R0  
ASL R0  
ADD 1$(R0),PC  
1$: .WORD 2$-1$  
.WORD 3$-1$  
.WORD 4$-1$  
.WORD 5$-1$  
.WORD 6$-1$  
.WORD 8$-1$  
.WORD 9$-1$  
.WORD 10$-1$
```

:ML4
:

```
.WORD 12$-1$  
.WORD 15$-1$  
2$: MOV NIB_SAVE,(R2)  
BIC #177760,(R2)  
BR 17$  
3$: MOV NIB_SAVE,R3
```

```
: NIB.PAT,*  
: RESULT,*  
: NIB.NUM,*
```

2314
2354
2350

2354
2358

2534	013444	006203		ASR	R3		
2535	013446	006203		ASR	R3		
2536	013450	006203		ASR	R3		
2537	013452	006203		ASR	R3		
2538	013454	042703	177760	BIC	#177760,R3		
2539	013460	010312		MOV	R3,(R2)		
2540	013462	000412		BR	7\$		
2541	013464	016704	174224	4\$: MOV	NIB.SAVE,R4	:	2362
2542	013470	000437		BR	13\$		
2543	013472	016704	174216	5\$: MOV	NIB.SAVE,R4	:	2366
2544	013476	000425		BR	11\$		
2545	013500	016712	174212	6\$: MOV	NIB.SAVE+2,(R2)	:	2370
2546	013504	042712	177760	BIC	#177760,(R2)		
2547	013510	010104		7\$: MOV	R1,R4		
2548	013512	041204		BIC	(R2),R4		
2549	013514	040112		BIC	R1,(R2)		
2550	013516	050412		BIS	R4,(R2)		
2551	013520	000207		RTS	PC	:	2350
2552	013522	016704	174170	8\$: MOV	NIB.SAVE+2,R4	:	2374
2553	013526	006204		ASR	R4		
2554	013530	006204		ASR	R4		
2555	013532	006204		ASR	R4		
2556	013534	006204		ASR	R4		
2557	013536	000415		BR	14\$		
2558	013540	016704	174152	9\$: MOV	NIB.SAVE+2,R4	:	2378
2559	013544	000411		BR	13\$		
2560	013546	016704	174144	10\$: MOV	NIB.SAVE+2,R4	:	2382
2561	013552	006204		11\$: ASR	R4		
2562	013554	006204		ASR	R4		
2563	013556	006204		ASR	R4		
2564	013560	006204		ASR	R4		
2565	013562	000402		BR	13\$		
2566	013564	016704	174130	12\$: MOV	NIB.SAVE+4,R4	:	2386
2567	013570	000304		13\$: SWAB	R4		
2568	013572	042704	177760	14\$: BIC	#177760,R4		
2569	013576	000411		BR	16\$		
2570	013600	016704	174114	15\$: MOV	NIB.SAVE+4,R4	:	2390
2571	013604	006204		ASR	R4		
2572	013606	006204		ASR	R4		
2573	013610	006204		ASR	R4		
2574	013612	006204		ASR	R4		
2575	013614	000304		SWAB	R4		
2576	013616	042704	177770	BIC	#177770,R4		
2577	013622	010412		16\$: MOV	R4,(R2)		
2578	013624	010103		17\$: MOV	R1,R3		
2579	013626	041203		BIC	(R2),R3		
2580				:ML4			
2581				:			
2582							
2583	013630	040112		BIC	R1,(R2)		
2584	013632	050312		BIS	R3,(R2)		
2585	013634	000207		RTS	PC	:	2314
2590							
2591							

22-Oct-1980 10:47:44 TOPS
22-Oct-1980 10:45:32 PA:<

22-Oct-1980 10:47:44
22-Oct-1980 10:45:32

TOPS-20 Bliss-16 V2(206)
PA:<NEALE>BL2ML4.BLI.2 (13)

2593 :ML4
2594 :
2595 :
2596 :
2597 :
2598 :
2599 :
2600 :
2601 :
2602 :
2603 :
2604 :
2605 :
2606 :
2607 :
2608 :
2609 :
2610 :
2611 :
2612 :
2613 :
2614 :
2615 :
2616 :
2617 :
2618 :
2619 :
2620 :
2621 :
2622 :
2623 :
2624 :
2625 :
2626 :
2627 :
2628 :
2629 :
2630 :
2631 :
2632 :
2633 :
2634 :
2635 :
2636 :
2637 :
2638 :
2639 :
2640 :
2641 :
2642 :
2643 :
2644 :
2645 :
2646 :
2647 :

```
2395 routine LD_LNG_WRD (NIB_NUM, NIB_PAT) : novalue =
2396   begin
2397
2398   !++
2399   FUNCTIONAL DESCRIPTION:
2400     LOADS 'NIB_SAVE' WITH UNIQUE
2401     NIBBLE PATTERNS PRIOR TO WRITING
2402     TO MLD1, MLD2 AND MLE2
2403     DATA DIAGNOSTIC REGISTERS.
2404
2405   FORMAL PARAMETERS:
2406     NIB_NUM
2407     CASE SELECT EXPRESSION TO SELECT
2408     THE CURRENT NIBBLE TO BE LOADED
2409
2410     NIB_PAT
2411     CURRENT NIBBLE PATTERN TO BE
2412     LOADED
2413
2414   IMPLICIT INPUTS:
2415     NIB_SAVE
2416     BLOCK OF 3 WORDS TO STORE
2417     THE DATA TO BE WRITTEN
2418     INTO MLD1 MLD2 MLE2
2419
2420   IMPLICIT OUTPUTS:
2421     NIB_SAVE IS LOADED WITH
2422     THE CURRENT NIBBLE PATTERN
2423   --
2424
2425   case .(NIB_NUM) from 0 to 9 of
2426     set
2427
2428     [0] :
2429       NIB_SAVE [NIB_0] = .NIB_PAT;
2430
2431     [1] :
2432       NIB_SAVE [NIB_1] = .NIB_PAT;
2433
2434     [2] :
2435       NIB_SAVE [NIB_2] = .NIB_PAT;
2436
2437     [3] :
2438       NIB_SAVE [NIB_3] = .NIB_PAT;
2439
2440     [4] :
2441       NIB_SAVE [NIB_4] = .NIB_PAT;
2442
2443     [5] :
2444       NIB_SAVE [NIB_5] = .NIB_PAT;
2445
2446     [6] :
```

!SELECT THE NIBBLE LOCATION IN NIB_SAVE TO BE LOADED

!LOAD NIBBLE 0 WITH NIB_PAT

!LOAD NIBBLE 1 WITH NIB_PAT

!LOAD NIBBLE 2 WITH NIB_PAT

!LOAD NIBBLE 3 WITH NIB_PAT

!LOAD NIBBLE 4 WITH NIB_PAT

!LOAD NIBBLE 5 WITH NIB_PAT

22-Oct-1980 10:47:44 TOPS-20 Bliss-16 V2(206)
22-Oct-1980 10:45:32 PA:<NEALE>BL2ML4.BLI.2 (13)

```

2649 :ML4
2650 :
2651 :
2652 :      2447      NIB_SAVE [NIB_6] = .NIB_PAT;      !LOAD NIBBLE 6 WITH NIB_PAT
2653 :      2448
2654 :      2449      [7] :      NIB_SAVE [NIB_7] = .NIB_PAT;      !LOAD NIBBLE 7 WITH NIB_PAT
2655 :      2450
2656 :      2451      [8] :      NIB_SAVE [NIB_8] = .NIB_PAT;      !LOAD NIBBLE 8 WITH NIB_PAT
2657 :      2452
2658 :      2453      [9] :      NIB_SAVE [NIB_9] = .NIB_PAT;      !LOAD NIBBLE 9 WITH NIB_PAT
2659 :      2454
2660 :      2455
2661 :      2456
2662 :      2457
2663 :      2458
2664 :      2459
2668 :
2672 013636      end;

```

```

LD.LNG.WRD:
2673 013636      010146      MOV      R1,-(SP)      ;
2674 013640      016600      000004      MOV      4(SP),R0      ; NIB.PAT,*
2675 013644      016601      000006      MOV      6(SP),R1      ; NIB.NUM,*
2676 013650      006301      ASL      R1
2677 013652      066107      013656      ADD      1$(R1),PC
2678 013656      000024      1$:      .WORD   2$-1$
2679 013660      000040      .WORD   3$-1$
2680 013662      000064      .WORD   4$-1$
2681 013664      000102      .WORD   5$-1$
2682 013666      000134      .WORD   7$-1$
2683 013670      000150      .WORD   8$-1$
2684 013672      000174      .WORD   9$-1$
2685 013674      000212      .WORD  10$-1$
2686 013676      000244      .WORD  12$-1$
2687 013700      000262      .WORD  13$-1$
2688 013702      042700      177760      2$:      BIC      #177760,R0      ;
2689 013706      142767      000017      174000      BICB     #17,NIB.SAVE
2690 013714      000433      BR      6$
2691 013716      006300      3$:      ASL      R0      ;
2692 013720      006300      ASL      R0
2693 013722      006300      ASL      R0
2694 013724      006300      ASL      R0
2695 013726      042700      177417      BIC      #177417,R0
2696 013732      142767      000360      173754      BICB     #360,NIB.SAVE
2697 013740      000421      BR      6$
2698 013742      000300      4$:      SWAB     R0      ;
2699 013744      042700      170377      BIC      #170377,R0
2700 013750      042767      007400      173736      BIC      #7400,NIB.SAVE
2701 013756      000412      BR      6$
2702 013760      000300      5$:      SWAB     R0      ;

```

```

2704      ;ML4
2705      ;
2706
2707 013762 006300      ASL      R0
2708 013764 006300      ASL      R0
2709 013766 006300      ASL      R0
2710 013770 006300      ASL      R0
2711 013772 042700 007777 BIC      #7777,R0
2712 013776 042767 170000 173710 BIC      #170000,NIB.SAVE
2713 014004 050067 173704 6$: BIS      R0,NIB.SAVE
2714 014010 000467      BR      15$      ; 2425
2715 014012 042700 177760 7$: BIC      #177760,R0      ; 2441
2716 014016 142767 000017 173672 BICB     #17,NIB.SAVE+2
2717 014024 000433      BR      11$
2718 014026 006300 8$: ASL      R0      ; 2444
2719 014030 006300      ASL      R0
2720 014032 006300      ASL      R0
2721 014034 006300      ASL      R0
2722 014036 042700 177417 BIC      #177417,R0
2723 014042 142767 000360 173646 BICB     #360,NIB.SAVE+2
2724 014050 000421      BR      11$
2725 014052 000300 9$: SWAB     R0      ; 2447
2726 014054 042700 170377 BIC      #170377,R0
2727 014060 042767 007400 173630 BIC      #7400,NIB.SAVE+2
2728 014066 000412      BR      11$
2729 014070 000300 10$: SWAB     R0      ; 2450
2730 014072 006300      ASL      R0
2731 014074 006300      ASL      R0
2732 014076 006300      ASL      R0
2733 014100 006300      ASL      R0
2734 014102 042700 007777 BIC      #7777,R0
2735 014106 042767 170000 173602 BIC      #170000,NIB.SAVE+2
2736 014114 050067 173576 11$: BIS      R0,NIB.SAVE+2
2737 014120 000423      BR      15$      ; 2425
2738 014122 000300 12$: SWAB     R0      ; 2453
2739 014124 042700 170377 BIC      #170377,R0
2740 014130 042767 007400 173562 BIC      #7400,NIB.SAVE+4
2741 014136 000412      BR      14$
2742 014140 000300 13$: SWAB     R0      ; 2456
2743 014142 006300      ASL      R0
2744 014144 006300      ASL      R0
2745 014146 006300      ASL      R0
2746 014150 006300      ASL      R0
2747 014152 042700 107777 BIC      #107777,R0
2748 014156 042767 070000 173534 BIC      #70000,NIB.SAVE+4
2749 014164 050067 173530 14$: BIS      R0,NIB.SAVE+4
2750 014170 012601 15$: MOV      (SP)+,R1      ; 2395
2751 014172 000207      RTS      PC
2752
2753
2754

```

```

; Routine Size: 111 words
; Maximum stack depth per invocation: 2 words

```

22-Oct-1980 10:47:44
 22-Oct-1980 10:45:32

TOPS-20 Bliss-16 V2(206)
 PA:<NEALE>BL2ML4.BLI.2 (14)

2760 ;ML4
 2761 ;
 2762 ;
 2763 ;
 2764 ;
 2765 ;
 2766 ;
 2767 ;
 2768 ;
 2769 ;
 2770 ;
 2771 ;
 2772 ;
 2773 ;
 2774 ;
 2775 ;
 2776 ;
 2777 ;
 2778 ;
 2779 ;
 2780 ;
 2781 ;
 2782 ;
 2783 ;
 2787 ;
 2791 014174
 2792 014200
 2793 014204
 2794 014206
 2795 014210
 2796 014212
 2797 014214
 2798 014220
 2799 014224
 2800 014230
 2801 014234
 2802 014236
 2803 014242
 2804
 2805
 2806
 2811
 2812

```

2460 routine WRT_CS1 (TST_PAT, index) : novalue =
2461   begin
2462
2463   !++
2464   FUNCTIONAL DESCRIPTION:
2465
2466   LOADS THE CONTROL & STATUS REGISTER 1 WITH A DATA PATTERN
2467   GENERATED BY THE MACRO WRT_MASK.
2468
2469   FORMAL PARAMETERS:
2470   TST_PAT
2471   CURRENT DATA PATTERN TO BE LOADED IN THE REGISTER.
2472
2473   INDEX
2474   USED BY THE MACRO WRT_MASK TO SELECT THE CURRENT REGISTERS ADDRESS,
2475   FORCED HI, FORCED LO AND DON'T CARE MASK INFORMATION.
2476
2477   !--
2478
2479   MLCS1 = WRT_MASK;
2480   end;
  
```

!LOAD GENERATED WRT_MASK PATTERN INTO MLCS1

```

WRT_CS1:JSR   R1,$SAVE2           ;
           MOV   10(SP),R0       ; INDEX,*
           ASL   R0
           ASL   R0
           ASL   R0
           MOV   R0,R1
           MOV   ML.REG+2(R1),R0
           BIS   12(SP),R0       ; TST.PAT,*
           BIC   ML.REG+4(R1),R0
           MOV   ML.REG+6(R1),R2
           BIS   R0,R2
           MOV   R2,@ML.REG
           RIS   PC
  
```

; Routine Size: 20 words
 ; Maximum stack depth per invocation: 3 words

2460
 2479
 2460

22-Oct-1980 10:47:44
22-Oct-1980 10:45:32

TOPS-20 Bliss-16 V2(206)
PA:<NEALE>BL2ML4.BLI.2 (15)

2814 :ML4
2815 :
2816 :
2817 :
2818 :
2819 :
2820 :
2821 :
2822 :
2823 :
2824 :
2825 :
2826 :
2827 :
2828 :
2829 :
2830 :
2831 :
2832 :
2833 :
2834 :
2835 :
2836 :
2837 :
2838 :
2839 :
2840 :
2841 :
2842 :
2843 :
2844 :
2845 :
2846 :
2847 :
2848 :
2849 :
2850 :
2851 :
2852 :
2853 :
2854 :
2855 :
2856 :
2857 :
2858 :
2859 :
2860 :
2861 :
2862 :
2863 :
2864 :
2865 :
2866 :
2867 :
2868 :

```
2481 routine RD_CS1 (TST_PAT, index, ERR_FLG) : novalue =
2482   begin
2483
2484   ++
2485   FUNCTIONAL DESCRIPTION:
2486
2487   COMPARES THE CONTENTS OF THE
2488   CONTROL & STATUS REGISTER 1
2489   WITH THE MASKED DATA PATTERN
2490   GENERATED BY THE MACRO 'WRT_MASK'
2491
2492   IF THE COMPARE IS NOT EQUAL THEN
2493   THE FORMAL PARAMETER 'ERR_FLG' IS
2494   ASSIGNED A ONE TO INDICATE THE
2495   ERROR.
2496
2497   FORMAL PARAMETERS:
2498
2499   TST_PAT
2500   DATA PATTERN TO BE MASKED AND
2501   COMPARED AGAINST THE CONTENTS
2502   OF THE REGISTER UNDER TEST.
2503
2504   INDEX
2505   USED BY THE MACRO WRT_MASK TO
2506   SELECT THE CURRENT REGISTER ADDRESS,
2507   FORCED HI, FORCED LO AND DON'T CARE
2508   MASK INFORMATION.
2509
2510   ERR_FLG
2511   CONTAINS THE ADDRESS (PASSED BY REF)
2512   OF THE CALLERS ERROR FLG TO ENABLE THE
2513   CALLER TO EXAMINE THE ERROR STATUS
2514   OF THE ROUTINE CALL.
2515
2516   IMPLICIT INPUTS:
2517   WT_DATA
2518   GETS LOADED WITH THE GENERATED
2519   WRT_MASK DATA PATTERN THUS ALLOWING
2520   CALLER TO PRINT FAILING GOOD DATA.
2521
2522   RD_DATA
2523   GETS LOADED WITH DATA READ FROM THE
2524   REGISTER THUS ALLOWING CALLER
2525   TO PRINT FAILING BAD DATA.
2526
2527   IMPLICIT OUTPUTS:
2528   GLOBAL LOCATION WR_DATA
2529   AND RD_DATA LOADED WITH GOOD
2530   AND BAD REGISTER DATA
2531
2532   --
```

2870 ;ML4
2871 ;
2872 ;
2873 ; 2533
2874 ; 2534
2875 ; 2535
2876 ; 2536
2877 ; 2537
2878 ; 2538
2879 ; 2539
2880 ; 2540

22-Oct-1980 10:47:44 TOPS-20 Bliss-16 V2(206)
22-Oct-1980 10:45:32 PA:<NEALE>BL2ML4.BLI.2 (15)

```
.ERR_FLG = ZERO;           !CLEAR ERROR FLAG
WT_DATA = WRT_MASK;        !SAVE THE DATA WRITTEN TO THE REGISTER
RD_DATA = .MLCS1 or .IGNORE; !READ AND SAVE THE REGISTER

if .WT_DATA neq .RD_DATA then .ERR_FLG = ONE;    !READ THE REG FOR WRT_MASK

end;                               !SET ERROR FLAG IF NEQ
```

2884									
2888	014244	004167	167536	RD.CS1:	JSR	R1,\$SAVE2			2481
2889	014250	005076	000010		CLR	@10(SP)		; ERR.FLG	2533
2890	014254	016600	000012		MOV	12(SP),R0		; INDEX,*	2534
2891	014260	006300			ASL	R0			
2892	014262	006300			ASL	R0			
2893	014264	006300			ASL	R0			
2894	014266	010001			MOV	R0,R1			
2895	014270	016100	011702		MOV	ML.REG+2(R1),R0			
2896	014274	056600	000014		BIS	14(SP),R0		; TST.PAT,*	
2897	014300	046100	011704		BIC	ML.REG+4(R1),R0			
2898	014304	016102	011706		MOV	ML.REG+6(R1),R2			
2899	014310	050002			BIS	R0,R2			
2900	014312	010267	175352		MOV	R2,WT_DATA			
2901	014316	017702	175356		MOV	@ML.REG,R2			2535
2902	014322	056102	011706		BIS	ML.REG+6(R1),R2			
2903	014326	010267	175340		MOV	R2,RD_DATA			
2904	014332	026767	175332	175332	CMP	WT_DATA,RD_DATA			2537
2905	014340	001403			BEQ	1\$			
2906	014342	012776	000001	000010	MOV	#1,@10(SP)		; *,ERR.FLG	
2907	014350	000207		1\$:	RTS	PC			2481

2908
2909
2910
2915
2916

```
; Routine Size: 35 words
; Maximum stack depth per invocation: 3 words
```


2918 :ML4
 2919 :
 2920 :
 2921 :
 2922 :
 2923 :
 2924 :
 2925 :
 2926 :
 2927 :
 2928 :
 2929 :
 2930 :
 2931 :
 2932 :
 2933 :
 2934 :
 2935 :
 2936 :
 2937 :
 2938 :
 2939 :
 2940 :
 2944 :

2541 routine WRT_ER (TST_PAT, index) : novalue =
 2542 begin
 2543
 2544 !++
 2545 FUNCTIONAL DESCRIPTION:
 2546
 2547 LOADS THE ERROR REGISTER WITH A DATA PATTERN GENERATED BY THE MACRO WRT_MASK
 2548
 2549 FORMAL PARAMETERS:
 2550
 2551 TST_PAT
 2552 CURRENT DATA PATTERN TO BE LOADED IN THE REGISTER.
 2553
 2554 INDEX
 2555 USED BY THE MACRO WRT_MASK TO SELECT THE CURRENT REGISTERS ADDRESS,
 2556 FORCED HI, FORCED LO AND DON'T CARE MASK INFORMATION.
 2557 !--
 2558
 2559 MLER = WRT_MASK;
 2560 end;

22-Oct-1980 10:47:44
 22-Oct-1980 10:45:32

TOPS-20 Bliss-16 V2(206)
 PA:<NEALE>BL2ML4.BLI.2 (16)

!LOAD GENERATE WRT_MASK PATTERN INTO MLER

2948	014352	004167	167430	WRT.ER:	JSR	R1,\$SAVE2	:	2541
2949	014356	016600	000010		MOV	10(SP),R0	; INDEX,*	2559
2950	014362	006300			ASL	R0		
2951	014364	006300			ASL	R0		
2952	014366	006300			ASL	R0		
2953	014370	010001			MOV	R0,R1		
2954	014372	016100	011702		MOV	ML.REG+2(R1),R0		
2955	014376	056600	000012		BIS	12(SP),R0	; TST.PAT,*	
2956	014402	046100	011704		BIC	ML.REG+4(R1),R0		
2957	014406	016102	011706		MOV	ML.REG+6(R1),R2		
2958	014412	050002			BIS	R0,R2		
2959	014414	010277	175340		MOV	R2,@ML.REG+60		
2960	014420	000207			RTS	PC	:	2541
2961								
2962								
2963								
2968								
2969								

; Routine Size: 20 words
 ; Maximum stack depth per invocation: 3 words

2971 :ML4
2972 :
2973 :
2974 :
2975 :
2976 :
2977 :
2978 :
2979 :
2980 :
2981 :
2982 :
2983 :
2984 :
2985 :
2986 :
2987 :
2988 :
2989 :
2990 :
2991 :
2992 :
2993 :
2994 :
2995 :
2996 :
2997 :
2998 :
2999 :
3000 :
3001 :
3002 :
3003 :
3004 :
3005 :
3006 :
3007 :
3008 :
3009 :
3010 :
3011 :
3012 :
3013 :
3014 :
3015 :
3016 :
3017 :
3018 :
3019 :
3020 :
3021 :
3022 :
3023 :
3024 :
3025 :

22-Oct-1980 10:47:44
22-Oct-1980 10:45:32

TOPS-20 Bliss-16 V2(206)
PA:<NEALE>BL2ML4.BLI.2 (17)

```
2561 routine RD_ER (TST_PAT, index, ERR_FLG) : novalue =  
2562     begin  
2563  
2564     ++  
2565     FUNCTIONAL DESCRIPTION:  
2566  
2567     COMPARES THE CONTENTS OF THE  
2568     ERROR REGISTER WITH  
2569     THE MASKED DATA PATTERN  
2570     GENERATED BY THE MACRO 'WRT_MASK'.  
2571  
2572     IF THE COMPARE IS NOT EQUAL THEN  
2573     THE FORMAL PARAMETER 'ERR_FLG' IS  
2574     ASSIGNED A ONE TO INDICATE THE  
2575     ERROR.  
2576  
2577     FORMAL PARAMETERS:  
2578  
2579     TST_PAT  
2580     DATA PATTERN TO BE MASKED AND  
2581     COMPARED AGAINST THE CONTENTS  
2582     OF THE REGISTER UNDER TEST.  
2583  
2584     INDEX  
2585     USED BY THE MACRO WRT_MASK TO  
2586     SELECT THE CURRENT REGISTERS ADDRESS,  
2587     FORCED HI, FORCED LO AND DON'T CARE  
2588     MASK INFORMATION.  
2589  
2590     ERR_FLG  
2591     CONTAINS THE ADDRESS (PASSED BY REF)  
2592     OF THE CALLERS ERROR FLG TO ENABLE THE  
2593     CALLER TO EXAMINE THE ERROR STATUS  
2594     OF THE ROUTINE CALL.  
2595  
2596     IMPLICIT INPUTS:  
2597     WT_DATA  
2598     GETS LOADED WITH THE GENERATED  
2599     WRT_MASK DATA PATTERN THUS ALLOWING  
2600     CALLER TO PRINT FAILING GOOD DATA.  
2601  
2602     RD_DATA  
2603     GETS LOADED WITH DATA READ FROM THE  
2604     REGISTER THUS ALLOWING CALLER  
2605     TO PRINT FAILING BAD DATA.  
2606  
2607     IMPLICIT OUTPUTS:  
2608     GLOBAL LOCATION WR_DATA  
2609     AND RD_DATA LOADED WITH GOOD  
2610     AND BAD REGISTER DATA  
2611  
2612     --
```

22-Oct-1980 10:47:44 TOPS-20 Bliss-16 V2(206)
 22-Oct-1980 10:45:32 PA:<NEALE>BL2ML4.BLI.2 (17)

3027 :ML4
 3028 :
 3029 :
 3030 : 2613
 3031 : 2614
 3032 : 2615
 3033 : 2616
 3034 : 2617
 3035 : 2618
 3036 : 2619
 3037 : 2620
 3038 : 2621

```

.ERR_FLG = ZERO;           !CLEAR THE ERROR FLAG
WT_DATA = WRT_MASK;       !SAVE THE DATA WRITTEN TO THE REGISTER
RD_DATA = .MLER or .IGNORE; !READ AND SAVE THE REGISTER

if .WT_DATA neq .RD_DATA then .ERR_FLG = ONE;   !READ THE REG FOR WRT_MASK

end;                          !SET ERROR FLAG IF NEQ

```

```

3042
3046 014422 004167 167360      RD.ER: JSR      R1,$SAVE2           ;
3047 014426 005076 000010      CLR      @10(SP)           ; ERR.FLG
3048 014432 016600 000012      MOV      12(SP),R0        ; INDEX,*
3049 014436 006300              ASL      R0
3050 014440 006300              ASL      R0
3051 014442 006300              ASL      R0
3052 014444 010001              MOV      R0,R1
3053 014446 016100 011702      MOV      ML.REG+2(R1),R0
3054 014452 056600 000014      BIS      14(SP),R0        ; TST.PAT,*
3055 014456 046100 011704      BIC      ML.REG+4(R1),R0
3056 014462 016102 011706      MOV      ML.REG+6(R1),R2
3057 014466 050002              BIS      R0,R2
3058 014470 010267 175174      MOV      R2,WT.DATA
3059 014474 017702 175260      MOV      @ML.REG+60,R2    ;
3060 014500 056102 011706      BIS      ML.REG+6(R1),R2
3061 014504 010267 175162      MOV      R2,RD.DATA
3062 014510 026767 175154 175154  CMP      WT.DATA,RD.DATA  ;
3063 014516 001403              BEQ      1$
3064 014520 012776 000001 000010  MOV      #1,@10(SP)      ; *,ERR.FLG
3065 014526 000207      1$: RTS      PC           ;

```

```

; Routine Size: 35 words
; Maximum stack depth per invocation: 3 words

```

3066
 3067
 3068
 3073
 3074

22-Oct-1980 10:47:44 TOPS-20 Bliss-16 V2(206)
22-Oct-1980 10:45:32 PA:<NEALE>BL2ML4.BLI.2 (18)

3076 :ML4
3077 :
3078 :
3079 :
3080 :
3081 :
3082 :
3083 :
3084 :
3085 :
3086 :
3087 :
3088 :
3089 :
3090 :
3091 :
3092 :
3093 :
3094 :
3095 :
3096 :
3097 :
3098 :
3102 :
3106 :
3107 :
3108 :
3109 :
3110 :
3111 :
3112 :
3113 :
3114 :
3115 :
3116 :
3117 :
3118 :
3119 :
3120 :
3121 :
3126 :
3127 :

```

2622 routine WRT_DA (TST_PAT, index) : novalue =
2623   begin
2624
2625 !++
2626 ! FUNCTIONAL DESCRIPTION:
2627 !
2628 !   LOADS THE DESIRED SECTOR WITH A DATA PATTERN GENERATED BY THE MACRO WRT_MASK
2629 !
2630 ! FORMAL PARAMETERS:
2631 !
2632 !   TST_PAT
2633 !   CURRENT DATA PATTERN TO BE LOADED IN THE REGISTER.
2634 !
2635 !   INDEX
2636 !   USED BY THE MACRO WRT_MASK TO SELECT THE CURRENT REGISTERS ADDRESS,
2637 !   FORCED HI, FORCED LO AND DON'T CARE MASK INFORMATION.
2638 !--
2639
2640 MLDA = WRT_MASK;
2641 end;

```

!LOAD MLDA WITH GENERATED WRT_MASK PATTERN

```

WRT.DA: JSR    R1,$SAVE2           ;
        MOV    10(SP),R0          ; INDEX,*
        ASL   R0
        ASL   R0
        ASL   R0
        MOV   R0,R1
        MOV   ML.REG+2(R1),R0
        BIS   12(SP),R0           ; TST.PAT,*
        BIC   ML.REG+4(R1),R0
        MOV   ML.REG+6(R1),R2
        BIS   R0,R2
        MOV   R2,@ML.REG+30
        RTS   PC

```

; Routine Size: 20 words
; Maximum stack depth per invocation: 3 words

3129 :ML4
3130 :
3131 :
3132 :
3133 :
3134 :
3135 :
3136 :
3137 :
3138 :
3139 :
3140 :
3141 :
3142 :
3143 :
3144 :
3145 :
3146 :
3147 :
3148 :
3149 :
3150 :
3151 :
3152 :
3153 :
3154 :
3155 :
3156 :
3157 :
3158 :
3159 :
3160 :
3161 :
3162 :
3163 :
3164 :
3165 :
3166 :
3167 :
3168 :
3169 :
3170 :
3171 :
3172 :
3173 :
3174 :
3175 :
3176 :
3177 :
3178 :
3179 :
3180 :
3181 :
3182 :
3183 :

2642
2643
2644
2645
2646
2647
2648
2649
2650
2651
2652
2653
2654
2655
2656
2657
2658
2659
2660
2661
2662
2663
2664
2665
2666
2667
2668
2669
2670
2671
2672
2673
2674
2675
2676
2677
2678
2679
2680
2681
2682
2683
2684
2685
2686
2687
2688
2689
2690
2691
2692
2693

routine RD_DA (TST_PAT, index, ERR_FLG) : novalue =
begin

!++
FUNCTIONAL DESCRIPTION:

COMPARES THE CONTENTS OF THE
DESIRED SECTOR ADDRESS REGISTER
WITH THE MASKED DATA PATTERN
GENERATED BY THE MACRO 'WRT_MASK'

IF THE COMPARE IS NOT EQUAL THEN
THE FORMAL PARAMETER 'ERR_FLG' IS
ASSIGNED A ONE TO INDICATE THE
ERROR

FORMAL PARAMETERS:

TST_PAT
DATA PATTERN TO BE MASKED AND
COMPARED AGAINST THE CONTENTS
OF THE REGISTER UNDER TEST

INDEX
USED BY THE MACRO WRT_MASK TO
SELECT THE CURRENT REGISTER'S ADDRESS,
FORCED HI, FORCED LO AND DON'T CARE
MASK INFORMATION

ERR_FLG
CONTAINS THE ADDRESS (PASSED BY REF)
OF THE CALLER'S ERROR_FLG TO ENABLE THE
CALLER TO EXAMINE THE ERROR STATUS
OF THE ROUTINE CALL.

IMPLICIT INPUTS:

WT_DATA
GETS LOADED WITH THE GENERATED
WRT_MASK DATA PATTERN THUS ALLOWING
CALLER TO PRINT FAILING GOOD DATA.

RD_DATA
GETS LOADED WITH DATA READ FROM THE
REGISTER THUS ALLOWING CALLER
TO PRINT FAILING BAD DATA.

IMPLICIT OUTPUTS:

GLOBAL LOCATION WR_DATA
AND RD_DATA LOADED WITH GOOD
AND BAD REGISTER DATA

--

22-Oct-1980 10:47:44
22-Oct-1980 10:45:32

TOPS-20 Bliss-16 V2(206)
PA:<NEALE>BL2ML4.BLI.2 (19)

22-Oct-1980 10:47:44 TOPS-20 Bliss-16 V2(206)
22-Oct-1980 10:45:32 PA:<NEALE>BL2ML4.BLI.2 (19)

```

3185 ;ML4
3186 ;
3187 ;
3188 :      2694      .ERR_FLG = ZERO;          !CLEAR THE ERROR FLAG
3189 :      2695      WT_DATA = WRT_MASK;      !SAVE THE DATA WRITTEN TO THE REGISTER
3190 :      2696      RD_DATA = .MLDA or .IGNORE; !READ AND SAVE THE REGISTER
3191 :      2697
3192 :      2698      if .WT_DATA neq .RD_DATA then .ERR_FLG = ONE;    !READ THE REG FOR WRT_MASK
3193 :      2699
3194 :      2700      !SET ERROR FLAG IF NEQ
3195 :      2701
3199      end;

3203 014600 004167 167202      RD.DA: JSR      R1,$SAVE2          ;
3204 014604 005076 000010      CLR      @10(SP)          ; ERR.FLG
3205 014610 016600 000012      MOV      12(SP),R0      ; INDEX,*
3206 014614 006300              ASL      R0
3207 014616 006300              ASL      R0
3208 014620 006300              ASL      R0
3209 014622 010001              MOV      R0,R1
3210 014624 016100 011702      MOV      ML.REG+2(R1),R0
3211 014630 056600 000014      BIS      14(SP),R0      ; TST.PAT,*
3212 014634 046100 011704      BIC      ML.REG+4(R1),R0
3213 014640 016102 011706      MOV      ML.REG+6(R1),R2
3214 014644 050002              BIS      R0,R2
3215 014646 010267 175016      MOV      R2,WT_DATA
3216 014652 017702 175052      MOV      @ML.REG+30,R2  ;
3217 014656 056102 011706      BIS      ML.REG+6(R1),R2
3218 014662 010267 175004      MOV      R2,RD_DATA
3219 014666 026767 174776 174776  CMP      WT_DATA,RD_DATA ;
3220 014674 001403              BEQ      1$
3221 014676 012776 000001 000010  MOV      #1,@10(SP)    ; *.ERR.FLG
3222 014704 000207      1$: RTS      PC          ;
3223
3224      ; Routine Size: 35 words
3225      ; Maximum stack depth per invocation: 3 words
3230
3231

```

22-Oct-1980 10:47:44 TOPS-20 Bliss-16 V2(206)
22-Oct-1980 10:45:32 PA:<NEALE>BL2ML4.BLI.2 (20)

3233 ;ML4
3234 ;
3235 ;
3236 ;
3237 ;
3238 ;
3239 ;
3240 ;
3241 ;
3242 ;
3243 ;
3244 ;
3245 ;
3246 ;
3247 ;
3248 ;
3249 ;
3250 ;
3251 ;
3252 ;
3253 ;
3254 ;
3255 ;
3259 ;

```

2702 routine WRT_MR (TST_PAT, index) : novalue =
2703     begin
2704
2705     !++
2706     FUNCTIONAL DESCRIPTION:
2707
2708         LOADS THE MAINTENANCE REGISTER WITH A DATA PATTERN GENERATED BY THE MACRO WRT_MASK
2709
2710     FORMAL PARAMETERS:
2711
2712         TST_PAT
2713         CURRENT DATA PATTERN TO BE LOADED IN THE REGISTER.
2714
2715         INDEX
2716         USED BY THE MACRO WRT_MASK TO SELECT THE CURRENT REGISTERS ADDRESS,
2717         FORCED HI, FORCED LO AND DON'T CARE MASK INFORMATION.
2718     !--
2719
2720     MLMR = WRT_MASK;
2721     end;
    
```

!LOAD MLMR WITH GENERATED WRT_MASK PATTERN

3263	014706	004167	167074	WRT.MR:	JSR	R1,\$SAVE2	:		2702
3264	014712	016600	000010		MOV	10(SP),R0	:	INDEX,*	2720
3265	014716	006300			ASL	R0			
3266	014720	006300			ASL	R0			
3267	014722	006300			ASL	R0			
3268	014724	010001			MOV	R0,R1			
3269	014726	016100	011702		MOV	ML.REG+2(R1),R0			
3270	014732	056600	000012		BIS	12(SP),R0	:	TST.PAT,*	
3271	014736	046100	011704		BIC	ML.REG+4(R1),R0			
3272	014742	016102	011706		MOV	ML.REG+6(R1),R2			
3273	014746	050002			BIS	R0,R2			
3274	014750	010277	175044		MOV	R2,@ML.REG+120			
3275	014754	000207			RTS	PC	:		2702

; Routine Size: 20 words
; Maximum stack depth per invocation: 3 words

3276
3277
3278
3283
3284

22-Oct-1980 10:47:44
22-Oct-1980 10:45:32

TOPS-20 Bliss-16 V2(206)
PA:<NEALE>BL2ML4.BLI.2 (21)

```
3286 :ML4
3287 :
3288 :
3289 : 2722 routine RD_MR (TST_PAT, index, ERR_FLG) : novalue =
3290 : 2723   begin
3291 : 2724
3292 : 2725 !++
3293 : 2726 FUNCTIONAL DESCRIPTION:
3294 : 2727
3295 : 2728   COMPARES THE CONTENTS OF THE
3296 : 2729   MAINTENANCE REGISTER WITH THE
3297 : 2730   MASKED DATA PATTERN
3298 : 2731   GENERATED BY THE MACRO 'WRT_MASK'.
3299 : 2732
3300 : 2733   IF THE COMPARE IS NOT EQUAL THEN
3301 : 2734   THE FORMAL PARAMETER 'ERR_FLG' IS
3302 : 2735   ASSIGNED A ONE TO INDICATE THE
3303 : 2736   ERROR.
3304 : 2737
3305 : 2738 FORMAL PARAMETERS:
3306 : 2739
3307 : 2740   TST_PAT
3308 : 2741   DATA PATTERN TO BE MASKED AND
3309 : 2742   COMPARED AGAINST THE CONTENTS
3310 : 2743   OF THE REGISTER UNDER TEST.
3311 : 2744
3312 : 2745   INDEX
3313 : 2746   USED BY THE MACRO WRT_MASK TO
3314 : 2747   SELECT THE CURRENT REGISTER'S ADDRESS,
3315 : 2748   FORCED HI, FORCED LO AND DON'T CARE
3316 : 2749   MASK INFORMATION.
3317 : 2750
3318 : 2751   ERR_FLG
3319 : 2752   CONTAINS THE ADDRESS (PASSED BY REF)
3320 : 2753   OF THE CALLER'S ERROR_FLG TO ENABLE THE
3321 : 2754   CALLER TO EXAMINE THE ERROR STATUS
3322 : 2755   OF THE ROUTINE CALL.
3323 : 2756
3324 : 2757 IMPLICIT INPUTS:
3325 : 2758   WT_DATA
3326 : 2759   GETS LOADED WITH THE GENERATED
3327 : 2760   WRT_MASK DATA PATTERN THUS ALLOWING
3328 : 2761   CALLER TO PRINT FAILING GOOD DATA.
3329 : 2762
3330 : 2763   RD_DATA
3331 : 2764   GETS LOADED WITH DATA READ FROM THE
3332 : 2765   REGISTER THUS ALLOWING CALLER
3333 : 2766   TO PRINT FAILING BAD DATA.
3334 : 2767
3335 : 2768 IMPLICIT OUTPUTS:
3336 : 2769   GLOBAL LOCATION WR_DATA
3337 : 2770   AND RD_DATA LOADED WITH GOOD
3338 : 2771   AND BAD REGISTER DATA
3339 : 2772
3340 : 2773 !--
```


22-Oct-1980 10:47:44 TOPS-20 Bliss-16 V2(206)
22-Oct-1980 10:45:32 PA:<NEALE>BL2ML4.BLI.2 (21)

3342 :ML4
3343 :
3344 :
3345 : 2774
3346 : 2775
3347 : 2776
3348 : 2777
3349 : 2778
3350 : 2779
3351 : 2780
3352 : 2781
3353 : 2782

```
.ERR_FLG = ZERO;           !CLEAR THE ERROR FLAG  
WT_DATA = WRT_MASK;       !SAVE THE DATA WRITTEN TO THE REGISTER  
RD_DATA = .MLMR or .IGNORE; !READ AND SAVE THE REGISTER  
  
if .WT_DATA neq .RD_DATA then .ERR_FLG = ONE;   !READ THE REG FOR WRT_MASK  
  
end;                                     !SET ERROR FLAG IF NEQ
```

```
3357  
3361 014756 004167 167024      RD.MR: JSR    R1,$SAVE2           ; 2722  
3362 014762 005076 000010      CLR    @10(SP)                ; ERR.FLG 2775  
3363 014766 016600 000012      MOV    12(SP),R0              ; INDEX,* 2776  
3364 014772 006300              ASL    R0  
3365 014774 006300              ASL    R0  
3366 014776 006300              ASL    R0  
3367 015000 010001              MOV    R0,R1  
3368 015002 016100 011702      MOV    ML.REG+2(R1),R0  
3369 015006 056600 000014      BIS    14(SP),R0              ; TST.PAT,*  
3370 015012 046100 011704      BIC    ML.REG+4(R1),R0  
3371 015016 016102 011706      MOV    ML.REG+6(R1),R2  
3372 015022 050002              BIS    R0,R2  
3373 015024 010267 174640      MOV    R2,WT_DATA  
3374 015030 017702 174764      MOV    @ML.REG+120,R2         ; 2777  
3375 015034 056102 011706      BIS    ML.REG+6(R1),R2  
3376 015040 010267 174626      MOV    R2,RD_DATA  
3377 015044 026767 174620 174620  CMP    WT_DATA,RD_DATA       ; 2779  
3378 015052 001403              BEQ    1$  
3379 015054 012776 000001 000010  MOV    #1,@10(SP)           ; *,ERR.FLG  
3380 015062 000207      1$: RTS    PC                ; 2722
```

3381
3382 ; Routine Size: 35 words
3383 ; Maximum stack depth per invocation: 3 words
3388
3389

22-Oct-1980 10:47:44 TOPS-20 Bliss-16 V2(206)
22-Oct-1980 10:45:32 PA:<NEALE>BL2ML4.BLI.2 (22)

3391 :ML4
3392 :
3393 :
3394 :
3395 :
3396 :
3397 :
3398 :
3399 :
3400 :
3401 :
3402 :
3403 :
3404 :
3405 :
3406 :
3407 :
3408 :
3409 :
3410 :
3411 :
3415 :
3419 :
3420 :
3421 :
3422 :
3423 :
3424 :
3425 :
3426 :
3427 :
3428 :
3429 :
3430 :
3431 :
3432 :
3433 :
3434 :
3435 :
3436 :
3441 :
3442 :

```

2783 routine WRT_PA (TST_PAT, index) : novalue =
2784     begin
2785
2786     !++
2787     !FUNCTIONAL DESCRIPTION:
2788     !LOADS THE PROM ADDRESS REGISTER WITH A DATA PATTERN GENERATED BY THE MACRO WRT_MASK
2789     !FORMAL PARAMETERS:
2790     !TST PAT
2791     !CURRENT DATA PATTERN TO BE LOADED IN THE REGISTER.
2792     !INDEX
2793     !USED BY THE MACRO WRT_MASK TO SELECT THE CURRENT REGISTERS ADDRESS,
2794     !FORCED HI, FORCED LO AND DON'T CARE MASK INFORMATION.
2795     !--
2796
2797     PROM_DIS = ONE;           !SET PROM DISABLE BIT
2798     MLPA = WRT_MASK;         !LOAD MLPA WITH GENERATED WRT_MASK PATTERN
2799     PROM_DIS = ZERO;        !CLEAR PROM DISABLE BIT
2800     end;

```

```

WRT.PA: JSR    R1,$SAVE2           ;
        BISB  #40,@ML.REG+120    ;
        MOV   10(SP),R0          ; INDEX,*
        ASL  R0
        ASL  R0
        ASL  R0
        MOV  R0,R1
        MOV  ML.REG+2(R1),R0
        BIS  12(SP),R0           ; TST.PAT,*
        BIC  ML.REG+4(R1),R0
        MOV  ML.REG+6(R1),R2
        BIS  R0,R2
        MOV  R2,@ML.REG+100
        BICB #40,@ML.REG+120    ;
        RTS   PC                 ;

```

; Routine Size: 26 words
; Maximum stack depth per invocation: 3 words

2783
2797
2798
2799
2783
2783

3444 :ML4
3445 :
3446 :
3447 :
3448 :
3449 :
3450 :
3451 :
3452 :
3453 :
3454 :
3455 :
3456 :
3457 :
3458 :
3459 :
3460 :
3461 :
3462 :
3463 :
3464 :
3465 :
3466 :
3467 :
3468 :
3469 :
3470 :
3471 :
3472 :
3473 :
3474 :
3475 :
3476 :
3477 :
3478 :
3479 :
3480 :
3481 :
3482 :
3483 :
3484 :
3485 :
3486 :
3487 :
3488 :
3489 :
3490 :
3491 :
3492 :
3493 :
3494 :
3495 :
3496 :
3497 :
3498 :

```
2801 routine RD_PA (TST_PAT, index, ERR_FLG) : novalue =
2802   begin
2803
2804   !++
2805   FUNCTIONAL DESCRIPTION:
2806
2807   COMPARES THE CONTENTS OF THE
2808   FROM ADDRESS REGISTER
2809   WITH THE MASKED DATA PATTERN
2810   GENERATED BY THE MACRO 'WRT_MASK'.
2811
2812   IF THE COMPARE IS NOT EQUAL THEN
2813   THE FORMAL PARAMETER 'ERR_FLG' IS
2814   ASSIGNED A ONE TO INDICATE THE
2815   ERROR.
2816
2817   FORMAL PARAMETERS:
2818   TST PAT
2819   DATA PATTERN TO BE MASKED AND
2820   COMPARED AGAINST THE CONTENTS
2821   OF THE REGISTER UNDER TEST.
2822
2823   INDEX
2824   USED BY THE MACRO WRT_MASK TO
2825   SELECT THE CURRENT REGISTER'S ADDRESS.
2826   FORCED HI, FORCED LO AND DGN'T CARE
2827   MASK INFORMATION.
2828
2829   ERR_FLG
2830   CONTAINS THE ADDRESS (PASSED BY REF)
2831   OF THE CALLER'S ERROR_FLG TO ENABLE THE
2832   CALLER TO EXAMINE THE ERROR STATUS
2833   OF THE ROUTINE CALL.
2834
2835   IMPLICIT INPUTS:
2836   WT_DATA
2837   GETS LOADED WITH THE GENERATED
2838   WRT_MASK DATA PATTERN THUS ALLOWING
2839   CALLER TO PRINT FAILING GOOD DATA.
2840
2841   RD_DATA
2842   GETS LOADED WITH DATA READ FROM THE
2843   REGISTER THUS ALLOWING CALLER
2844   TO PRINT FAILING BAD DATA.
2845
2846   IMPLICIT OUTPUTS:
2847   GLOBAL LOCATION WR_DATA
2848   AND RD_DATA LOADED WITH GOOD
2849   AND BAD REGISTER DATA
2850
2851   --
2852
```

22-Oct-1980 10:47:44
22-Oct-1980 10:45:32

TOPS-20 Bliss-16 V2(206)
PA:<NEALE>BL2ML4.BLI.2 (23)

22-Oct-1980 10:47:44 TOPS-20 Bliss-16 V2(206)
22-Oct-1980 10:45:32 PA:<NEALE>BL2ML4.BLI.2 (23)

```

3500 ;ML4
3501 ;
3502 ;
3503 :      2853      .ERR_FLG = ZERO;          !CLEAR THE ERROR FLAG
3504 :      2854      PROM_DIS = ONE;          !SET THE PROM DISABLE BIT
3505 :      2855      WT_DATA = WRT_MASK;      !SAVE THE DATA WRITTEN TO THE REGISTER
3506 :      2856      RD_DATA = .MLPA or .IGNORE; !READ AND SAVE THE REGISTER
3507 :      2857
3508 :      2858      if .WT_DATA neq .RD_DATA then .ERR_FLG = ONE;    !READ THE REG FOR WRT_MASK
3509 :      2859
3510 :      2860      PROM_DIS = ZERO;          !SET ERROR IF NEQ
3511 :      2861      end;                    !CLEAR THE PROM DISABLE BIT
3515
3519 015150 004167 166632      RD.PA: JSR      R1,$SAVE2          ;
3520 015154 005076 000010      CLR      @10(SP)          ; ERR.FLG
3521 015160 152777 000040 174632  BISB     #40,@ML.REG+120 ;
3522 015166 016600 000012      MOV      12(SP),R0      ; INDEX,*
3523 015172 006300      ASL     R0
3524 015174 006300      ASL     R0
3525 015176 006300      ASL     R0
3526 015200 010001      MOV     R0,R1
3527 015202 016100 011702      MOV     ML.REG+2(R1),R0
3528 015206 056600 000014      BIS     14(SP),R0      ; TST.PAT,*
3529 015212 046100 011704      BIC     ML.REG+4(R1),R0
3530 015216 016102 011706      MOV     ML.REG+6(R1),R2
3531 015222 050002      BIS     R0,R2
3532 015224 010267 174440      MOV     R2,WT.DATA
3533 015230 017702 174544      MOV     @ML.REG+100,R2 ;
3534 015234 056102 011706      BIS     ML.REG+6(R1),R2
3535 015240 010267 174426      MOV     R2,RD.DATA
3536 015244 026767 174420 174420  CMP     WT.DATA,RD.DATA ;
3537 015252 001403      BEQ
3538 015254 012776 000001 000010  MOV     #1,@10(SP)      ; *,ERR.FLG
3539 015262 142777 000040 174530 1$: BICB     #40,@ML.REG+120 ;
3540 015270 000207      RTS     PC              ;
3541
3542
3543      ; Routine Size: 41 words
3548      ; Maximum stack depth per invocation: 3 words
3549

```

22-Oct-1980 10:47:44
22-Oct-1980 10:45:32

TOPS-20 Bliss-16 V2(206)
PA:<NEALE>BL2ML4.BLI.2 (24)

3551 ;ML4
3552 ;
3553 ;
3554 ;
3555 ;
3556 ;
3557 ;
3558 ;
3559 ;
3560 ;
3561 ;
3562 ;
3563 ;
3564 ;
3565 ;
3566 ;
3567 ;
3568 ;
3569 ;
3570 ;
3571 ;
3575 ;

2862 routine WRT_E1 (TST_PAT, index) : novalue =
2863 begin
2864
2865 !++
2866 FUNCTIONAL DESCRIPTION:
2867 LOADS THE ECC CRC WORD REG 1 WITH A DATA PATTERN GENERATED BY THE MACRO WRT_MASK
2868 FORMAL PARAMETERS:
2869 TST_PAT
2870 CURRENT DATA PATTERN TO BE LOADED IN THE REGISTER.
2871 INDEX
2872 USED BY THE MACRO WRT_MASK TO SELECT THE CURRENT REGISTERS ADDRESS,
2873 FORCED HI, FORCED LO AND DON'T CARE MASK INFORMATION.
2874 !--
2875
2876 ECC_DM = ONE;
2877 MLET = WRT_MASK;
2878 ECC_DM = ZERO;
2879 end;

!SET ECC DIAG MODE
!LOAD MLE1 WITH GENERATED WRT_MASK PATTERN
!CLEAR ECC DIAG MODE

3579	015272	004167	166510	WRT.E1:	JSR	R1,\$SAVE2	:	2862
3580	015276	152777	000001	174514	BISB	#1,@ML.REG+120	:	2876
3581	015304	016600	000010		MOV	10(SP),R0	: INDEX,*	2877
3582	015310	006300			ASL	R0		
3583	015312	006300			ASL	R0		
3584	015314	006300			ASL	R0		
3585	015316	010001			MOV	R0,R1		
3586	015320	016100	011702		MOV	ML.REG+2(R1),R0		
3587	015324	056600	000012		BIS	12(SP),R0	: TST.PAT,*	
3588	015330	046100	011704		BIC	ML.REG+4(R1),R0		
3589	015334	016102	011706		MOV	ML.REG+6(R1),R2		
3590	015340	050002			BIS	R0,R2		
3591	015342	010277	174502		MOV	R2,@ML.REG+150		
3592	015346	142777	000001	174444	BICB	#1,@ML.REG+120	:	2878
3593	015354	000207			RTS	PC	:	2862

3594
3595 ; Routine Size: 26 words
3596 ; Maximum stack depth per invocation: 3 words
3601
3602

3604 :ML4
3605 :
3606 :
3607 :
3608 :
3609 :
3610 :
3611 :
3612 :
3613 :
3614 :
3615 :
3616 :
3617 :
3618 :
3619 :
3620 :
3621 :
3622 :
3623 :
3624 :
3625 :
3626 :
3627 :
3628 :
3629 :
3630 :
3631 :
3632 :
3633 :
3634 :
3635 :
3636 :
3637 :
3638 :
3639 :
3640 :
3641 :
3642 :
3643 :
3644 :
3645 :
3646 :
3647 :
3648 :
3649 :
3650 :
3651 :
3652 :
3653 :
3654 :
3655 :
3656 :
3657 :
3658 :

22-Oct-1980 10:47:44
22-Oct-1980 10:45:32

TOPS-20 Bliss-16 V2(206)
PA:<NEALE>BL2ML4.BLI.2 (25)

```
2880 routine RD_E1 (TST_PAT, index, ERR_FLG) : novalue =
2881   begin
2882
2883   !++
2884   FUNCTIONAL DESCRIPTION:
2885
2886       COMPARES THE CONTENTS OF THE
2887       ECC CRC WORD REGISTER 1
2888       WITH THE MASKED DATA PATTERN
2889       GENERATED BY THE MACRO 'WRT_MASK'
2890
2891       IF THE COMPARE IS NOT EQUAL THEN
2892       THE FORMAL PARAMETER 'ERR_FLG' IS
2893       ASSIGNED A ONE TO INDICATE THE ERROR.
2894
2895   FORMAL PARAMETERS:
2896       TST_PAT
2897       DATA PATTERN TO BE MASKED AND
2898       COMPARED AGAINST THE CONTENTS
2899       OF THE REGISTER UNDER TEST.
2900
2901       INDEX
2902       USED BY THE MACRO WRT_MASK TO
2903       SELECT THE CURRENT REGISTER ADDRESS,
2904       FORCED HI, FORCED LO AND DON'T CARE
2905       MASK INFORMATION.
2906
2907       ERR_FLG
2908       CONTAINS THE ADDRESS (PASSED BY REF)
2909       OF THE CALLERS ERROR FLG TO ENABLE THE
2910       CALLER TO EXAMINE THE ERROR STATUS
2911       OF THE ROUTINE CALL.
2912
2913   IMPLICIT INPUTS:
2914       WT_DATA
2915       GETS LOADED WITH THE GENERATED
2916       WRT_MASK DATA PATTERN THUS ALLOWING
2917       CALLER TO PRINT FAILING GOOD DATA.
2918
2919       RD_DATA
2920       GETS LOADED WITH DATA READ FROM THE
2921       REGISTER THUS ALLOWING CALLER
2922       TO PRINT FAILING BAD DATA.
2923
2924   IMPLICIT OUTPUTS:
2925       GLOBAL LOCATION WR_DATA
2926       AND RD_DATA LOADED WITH GOOD
2927       AND BAD REGISTER DATA
2928
2929   !--
2930
2931   .ERR_FLG = ZERO;
```

!CLEAR THE ERROR FLAG

22-Oct-1980 10:47:44 TOPS-20 Bliss-16 V2(206)
22-Oct-1980 10:45:32 PA:<NEALE>BL2ML4.BLI.2 (25)

```

3660 :ML4
3661 :
3662 :
3663 :      2932      ECC_DM = ONE;
3664 :      2933      WT_DATA = WRT_MASK;
3665 :      2934      RD_DATA = .MLE1 or .IGNORE;
3666 :      2935
3667 :      2936      if .WT_DATA neq .RD_DATA then .ERR_FLG = ONE;
3668 :      2937
3669 :      2938      ECC_DM = ZERO;
3670 :      2939
3671 :      2940      end;
3675 :
3679 015356 004167 166424      RD.E1: JSR      R1,$SAVE2
3680 015362 005076 000010      CLR      @10(SP)
3681 015366 152777 000001 174424      BISB     #1,@ML.REG+120
3682 015374 016600 000012      MOV      12(SP),R0
3683 015400 006300      ASL      R0
3684 015402 006300      ASL      R0
3685 015404 006300      ASL      R0
3686 015406 010001      MOV      R0,R1
3687 015410 016100 011702      MOV      ML.REG+2(R1),R0
3688 015414 056600 000014      BIS      14(SP),R0
3689 015420 046100 011704      BIC      ML.REG+4(R1),R0
3690 015424 016102 011706      MOV      ML.REG+6(R1),R2
3691 015430 050002      BIS      R0,R2
3692 015432 010267 174232      MOV      R2,WT_DATA
3693 015436 017702 174406      MOV      @ML.REG+150,R2
3694 015442 056102 011706      BIS      ML.REG+6(R1),R2
3695 015446 010267 174220      MOV      R2,RD_DATA
3696 015452 026767 174212 174212      CMP      WT_DATA,RD_DATA
3697 015460 001403      BEQ      1$
3698 015462 012776 000001 000010      MOV      #1,@10(SP)
3699 015470 142777 000001 174322 1$: BICB     #1,@ML.REG+120
3700 015476 000207      RTS      PC
3701 :
3702 :
3703 :
3708 :
3709 :

```

```

!SET ECC DIAG MODE
!SAVE THE DATA WRITTEN TO THE REGISTER
!READ AND SAVE THE REGISTER
!READ THE REG FOR WRT_MASK
!SET ERROR FLAG IF NEQ
!CLEAR ECC DIAG MODE

```

```

2880
2931
2932
2933
2934
2936
2938
2880

```

```

; Routine Size: 41 words
; Maximum stack depth per invocation: 3 words

```

22-Oct-1980 10:47:44
22-Oct-1980 10:45:32

TOPS-20 Bliss-16 V2(206)
PA:<NEALE>BL2ML4.BLI.2 (26)

3711 :ML4

3712 :

3713 :

3714 : 2941 routine WRT_E2 (TST_PAT, index) : novalue =

3715 : 2942 begin

3716 : 2943

3717 : 2944 !++

3718 : 2945 FUNCTIONAL DESCRIPTION:

3719 : 2946

3720 : 2947 LOADS THE ECC CRC WORD

3721 : 2948 REGISTER 2 WITH A DATA PATTERN

3722 : 2949 GENERATED BY THE MACRO

3723 : 2950 WRT_MASK

3724 : 2951

3725 : 2952 FORMAL PARAMETERS:

3726 : 2953

3727 : 2954 TST_PAT

3728 : 2955 CURRENT DATA PATTERN TO BE LOADED IN THE REGISTER.

3729 : 2956

3730 : 2957 INDEX

3731 : 2958 USED BY THE MACRO WRT_MASK

3732 : 2959 TO SELECT THE CURRENT REGISTERS

3733 : 2960 ADDRESS, FORCED HI, FORCED

3734 : 2961 LO AND DON'T CARE MASK

3735 : 2962 INFORMATION.

3736 : 2963

3737 : 2964 !--

3738 : 2965

3739 : 2966 ECC_DM = ONE;

3740 : 2967 MLE2_MASK = %0'177700';

3741 : 2968 MLE2 = WRT_MASK;

3742 : 2969 MLE2_MASK = %0'000000';

3743 : 2970 ECC_DM = ZERO;

3744 : 2971 end;

!SET ECC DIAG MODE
!MASK OUT DATA DIAG BITS
!LOAD MLE2 WITH GENERATED WRT_MASK PATTERN
!RESTORE MLE2 MASK
!CLEAR ECC DIAG MODE

3748

3752 015500 004167 166302 WRT.E2: JSR R1,\$SAVE2 ; 2941

3753 015504 152777 000001 174306 BISB #1,@ML.REG+120 ; 2966

3754 015512 012767 177700 174346 MOV #-100,ML.REG+166 ; 2967

3755 015520 016600 000010 MOV 10(SP),R0 ; INDEX,* 2968

3756 015524 006300 ASL R0

3757 015526 006300 ASL R0

3758 015530 006300 ASL R0

3759 015532 010001 MOV R0,R1

3760 015534 016100 011702 MOV ML.REG+2(R1),R0

3761 015540 056600 000012 BIS 12(SP),R0 ; TST.PAT,*

3762 015544 046100 011704 BIC ML.REG+4(R1),R0

3763 015550 016102 011706 MOV ML.REG+6(R1),R2

3764 015554 050002 BIS R0,R2

3765 :ML4

3766 :

3767

3768 015556 010277 174276 MOV R2,@ML.REG+160

3769 015562 005067 174300 CLR ML.REG+166 ;

3770 015566 142777 000001 174224 BICB #1,@ML.REG+120 ;

3771 015574 000207 RTS PC ;

3776

3777

22-Oct-1980 10:47:44 TOPS
22-Oct-1980 10:45:32 PA:<

2969
2970
2941

3779 :ML4
3780 :
3781 :
3782 :
3783 :
3784 :
3785 :
3786 :
3787 :
3788 :
3789 :
3790 :
3791 :
3792 :
3793 :
3794 :
3795 :
3796 :
3797 :
3798 :
3799 :
3800 :
3801 :
3802 :
3803 :
3804 :
3805 :
3806 :
3807 :
3808 :
3809 :
3810 :
3811 :
3812 :
3813 :
3814 :
3815 :
3816 :
3817 :
3818 :
3819 :
3820 :
3821 :
3822 :
3823 :
3824 :
3825 :
3826 :
3827 :
3828 :
3829 :
3830 :
3831 :
3832 :
3833 :

22-Oct-1980 10:47:44
22-Oct-1980 10:45:32

TOPS-20 Bliss-16 V2(206)
PA:<NEALE>BL2ML4.BLI.2 (27)

2972 routine RD_E2 (TST_PAT, index, ERR_FLG) : novalue =
2973 begin
2974
2975 !++
2976 FUNCTIONAL DESCRIPTION:
2977
2978 COMPARES THE CONTENTS OF THE
2979 ECC CRC WORD REGISTER 2
2980 WITH THE MASKED DATA PATTERN
2981 GENERATED BY THE MACRO 'WRT_MASK'
2982
2983 IF THE COMPARE IS NOT EQUAL THEN
2984 THE FORMAL PARAMETER 'ERR_FLG' IS
2985 ASSIGNED A ONE TO INDICATE THE
2986 ERROR.
2987
2988 FORMAL PARAMETERS:
2989
2990 TST_PAT
2991 DATA PATTERN TO BE MASKED AND
2992 COMPARED AGAINST THE CONTENTS
2993 OF THE REGISTER UNDER TEST.
2994
2995 INDEX
2996 USED BY THE MACRO WRT_MASK TO
2997 SELECT THE CURRENT REGISTER ADDRESS,
2998 FORCED HI, FORCED LO AND DON'T CARE
2999 MASK INFORMATION.
3000
3001 ERR_FLG
3002 CONTAINS THE ADDRESS (PASSED BY REF)
3003 OF THE CALLERS ERROR FLG TO ENABLE THE
3004 CALLER TO EXAMINE THE ERROR STATUS
3005 OF THE ROUTINE CALL.
3006
3007 IMPLICIT INPUTS:
3008 WT_DATA
3009 GETS LOADED WITH THE GENERATED
3010 WRT_MASK DATA PATTERN THUS ALLOWING
3011 CALLER TO PRINT FAILING GOOD DATA.
3012
3013 RD_DATA
3014 GETS LOADED WITH DATA READ FROM THE
3015 REGISTER THUS ALLOWING CALLER
3016 TO PRINT FAILING BAD DATA.
3017
3018 IMPLICIT OUTPUTS:
3019 GLOBAL LOCATION WR_DATA
3020 AND RD_DATA LOADED WITH GOOD
3021 AND BAD REGISTER DATA
3022
3023 !--

22-Oct-1980 10:47:44 TOPS-20 Bliss-16 V2(206)
22-Oct-1980 10:45:32 PA:<NEALE>BL2ML4.BLI.2 (27)

```

3835 ;ML4
3836 ;
3837 ;
3838 : 3024
3839 : 3025 .ERR_FLG = ZERO; !CLEAR ERROR FLAG
3840 : 3026 ECC_DM = ONE; !SET ECC DIAG MODE
3841 : 3027 MLE2_MASK = %o'177700'; !MASK OUT DATA DIAG BITS
3842 : 3028 WT_DATA = WRT_MASK; !SAVE THE DATA WRITTEN TO THE REGISTER
3843 : 3029 RD_DATA = .MLE2 or .IGNORE; !READ AND SAVE THE REGISTER
3844 : 3030
3845 : 3031 if .WT_DATA neq .RD_DATA then .ERR_FLG = ONE; !READ THE REG FOR WRT_MASK
3846 : 3032
3847 : 3033 !SET ERROR FLG IF NEQ
3848 : 3034 MLE2_MASK = %o'000000'; !RESTORE MLE2 MASK
3849 : 3035 ECC_DM = ZERO; !CLEAR ECC DIAG MODE
3850 : 3036 end;
3854
3858 015576 004167 166204 RD.E2: JSR R1,$SAVE2 ; 2972
3859 015602 005076 000010 CLR @10(SP) ; ERR:FLG 3025
3860 015606 152777 000001 174204 BISB #1,@ML.REG+120 ; 3026
3861 015614 012767 177700 174244 MOV #-100,ML.REG+166 ; 3027
3862 015622 016600 000012 MOV 12(SP),R0 ; INDEX,* 3028
3863 015626 006300 ASL R0
3864 015630 006300 ASL R0
3865 015632 006300 ASL R0
3866 015634 010001 MOV R0,R1
3867 015636 016100 011702 MOV ML.REG+2(R1),R0
3868 015642 056600 000014 BIS 14(SP),R0 ; TST.PAT,*
3869 015646 046100 011704 BIC ML.REG+4(R1),R0
3870 015652 016102 011706 MOV ML.REG+6(R1),R2
3871 015656 050002 BIS R0,R2
3872 015660 010267 174004 MOV R2,WT_DATA
3873 015664 017702 174170 MOV @ML.REG+160,R2 ; 3029
3874 015670 056102 011706 BIS ML.REG+6(R1),R2
3875 015674 010267 173772 MOV R2,RD_DATA
3876 015700 026767 173764 173764 CMP WT_DATA,RD_DATA ; 3031
3877 015706 001403 BEQ 1$
3878 015710 012776 000001 000010 MOV #1,@10(SP) ; *,ERR.FLG
3879 015716 005067 174144 1$: CLR ML.REG+166 ; 3034
3880 015722 142777 000001 174070 BICB #1,@ML.REG+120 ; 3035
3881 015730 000207 RTS PC ; 2972
3882
3883 ; Routine Size: 46 words
3884 ; Maximum stack depth per invocation: 3 words

```

22-Oct-1980 10:47:44 TOPS-20 Bliss-16 V2(206)
 22-Oct-1980 10:45:32 PA:<NEALE>BL2ML4.BLI.2 (28)

```

3890 ;ML4
3891 ;
3892
3893 : 3037 routine WRT_PD (TST_PAT, index) : novalue =
3894 : 3038 begin
3895 : 3039
3896 : 3040 !++
3897 : 3041 FUNCTIONAL DESCRIPTION:
3898 : 3042
3899 : 3043 LOADS THE PROM DATA
3900 : 3044 REGISTER WITH A DATA PATTERN
3901 : 3045 GENERATED BY THE MACRO
3902 : 3046 WRT_MASK
3903 : 3047
3904 : 3048 FORMAL PARAMETERS:
3905 : 3049
3906 : 3050 TST_PAT
3907 : 3051 CURRENT DATA PATTERN TO BE
3908 : 3052 LOADED IN THE REGISTER.
3909 : 3053
3910 : 3054 INDEX
3911 : 3055 USED BY THE MACRO WRT_MASK
3912 : 3056 TO SELECT THE CURRENT REGISTERS
3913 : 3057 ADDRESS, FORCED HI, FORCED
3914 : 3058 LO AND DON'T CARE MASK
3915 : 3059 INFORMATION
3916 : 3060
3917 : 3061 !--
3918 : 3062
3919 : 3063 PROM_RW = ONE;
3920 : 3064 DAT_DM = ONE;
3921 : 3065 MLPD = WRT_MASK;
3922 : 3066 DAT_CLK = ONE;
3923 : 3067 PROM_RW = ZERO;
3924 : 3068 DAT_DM = ZERO;
3925 : 3069 end;
    
```

```

!SET PROM READ WRITE
!SET DATA DIAG MODE
!LOAD MLPD WITH GENERATED WRT_MASK PATTERN
!DO A DATA CLK
!CLEAR PROM READ WRITE
!CLEAR DATA DIAG MODE
    
```

```

3929
3933 015732 004167 166050 WRT.PD: JSR R1,$SAVE2 ; 3037
3934 015736 152777 000100 174054 BISB #100,@ML.REG+120 ; 3063
3935 015744 152777 000010 174046 BISB #10,@ML.REG+120 ; 3064
3936 015752 016600 000010 MOV 10(SP),R0 ; INDEX,* 3065
3937 015756 006300 ASL R0
3938 015760 006300 ASL R0
3939 015762 006300 ASL R0
3940 015764 010001 MOV R0,R1
3941 015766 016100 011702 MOV ML.REG+2(R1),R0
3942 015772 056600 000012 BIS 12(SP),R0 ; TST.PAT,*
3943 015776 046100 011704 BIC ML.REG+4(R1),R0
    
```

```
3945 ;ML4
3946 ;
3947
3948 016002 016102 011706 MOV ML.REG+6(R1),R2
3949 016006 050002 BIS R0,R2
3950 016010 010277 174114 MOV R2,@ML.REG+230
3951 016014 152777 000020 173776 BISB #20,@ML.REG+120 ;
3952 016022 142777 000100 173770 BICB #100,@ML.REG+120 ;
3953 016030 142777 000010 173762 BICB #10,@ML.REG+120 ;
3954 016036 000207 RTS PC ;
3955
3956 ; Routine Size: 35 words
3957 ; Maximum stack depth per invocation: 3 words
3962
3963
```

22-Oct-1980 10:47:44 TOPS
22-Oct-1980 10:45:32 PA:<

3066
3067
3068
3037

3965 :ML4

22-Oct-1980 10:47:44

TOPS-20 Bliss-16 V2(206)

3966 :

22-Oct-1980 10:45:32

PA:<NEALE>BL2ML4.BLI.2 (29)

3967 :

3968 :

3070 routine RD_PD (TST_PAT, index, ERR_FLG) : novalue =
3071 begin

3969 :

3970 :

3072

3971 :

3073

!++

3972 :

3074

FUNCTIONAL DESCRIPTION:

3973 :

3075

3974 :

3076

COMPARES THE CONTENTS OF THE
PROM DATA REGISTER
WITH THE MASKED DATA PATTERN
GENERATED BY THE MACRO 'WRT_MASK'

3975 :

3077

3976 :

3078

3977 :

3079

3978 :

3080

3979 :

3081

IF THE COMPARE IS NOT EQUAL THEN
THE FORMAL PARAMETER 'ERR_FLG' IS
ASSIGNED A ONE TO INDICATE THE
ERROR

3980 :

3082

3981 :

3083

3982 :

3084

3983 :

3085

FORMAL PARAMETERS:

3984 :

3086

3985 :

3087

TST PAT
DATA PATTERN TO BE MASKED AND
COMPARED AGAINST THE CONTENTS
OF THE REGISTER UNDER TEST

3986 :

3088

3987 :

3089

3988 :

3090

3989 :

3091

3990 :

3092

INDEX
USED BY THE MACRO WRT_MASK TO
SELECT THE CURRENT REGISTER ADDRESS,
FORCED HI, FORCED LO AND DON'T CARE
MASK INFORMATION.

3991 :

3093

3992 :

3094

3993 :

3095

3994 :

3096

3995 :

3097

3996 :

3098

ERR_FLG
CONTAINS THE ADDRESS (PASSED BY REF)
OF THE CALLERS ERROR_FLG TO ENABLE THE
CALLER TO EXAMINE THE ERROR STATUS
OF THE ROUTINE CALL.

3997 :

3099

3998 :

3100

3999 :

3101

4000 :

3102

4001 :

3103

4002 :

3104

IMPLICIT INPUTS:

4003 :

3105

WT_DATA
GETS LOADED WITH THE GENERATED
WRT_MASK DATA PATTERN THUS ALLOWING
CALLER TO PRINT FAILING GOOD DATA.

4004 :

3106

4005 :

3107

4006 :

3108

4007 :

3109

4008 :

3110

RD_DATA
GETS LOADED WITH DATA READ FROM THE
REGISTER THUS ALLOWING CALLER
TO PRINT FAILING BAD DATA.

4009 :

3111

4010 :

3112

4011 :

3113

4012 :

3114

4013 :

3115

IMPLICIT OUTPUTS:

4014 :

3116

GLOBAL LOCATION WR_DATA
AND RD_DATA LOADED WITH GOOD
AND BAD REGISTER DATA

4015 :

3117

4016 :

3118

4017 :

3119

4018 :

3120

4019 :

3121

--

22-Oct-1980 10:47:44 TOPS-20 Bliss-16 V2(206)
 22-Oct-1980 10:45:32 PA:<NEALE>BL2ML4.BLI.2 (29)

```

4021 :ML4
4022 :
4023 :
4024 :          3122
4025 :          3123      .ERR_FLG = ZERO;          !CLEAR ERROR FLG
4026 :          3124      PROM_RW = ONE;          !SET PROM READ WRITE
4027 :          3125      DAT_DM = ONE;          !SET DATA DIAG MODE
4028 :          3126      WT_DATA = WRT_MASK;      !SAVE THE DATA WRITTEN TO THE REGISTER
4029 :          3127      RD_DATA = .MLPD or .IGNORE; !READ AND SAVE THE REGISTER
4030 :          3128
4031 :          3129      if .WT_DATA neq .RD_DATA then .ERR_FLG = ONE;      !READ THE REG FOR WRT_MASK
4032 :          3130
4033 :          3131
4034 :          3132      PROM_RW = ZERO;          !SET ERROR FLAG IF NEQ
4035 :          3133      DAT_DM = ZERO;          !CLEAR PROM READ WRITE
4036 :          3134      end;          !CLEAR DATA DIAG MODE
4040 :
4044 016040 004167 165742      RD.PD: JSR      R1,$SAVE2          ;
4045 016044 005076 000010      CLR      @10(SP)          ; ERR.FLG
4046 016050 152777 000100 173742  BISB     #100,@ML.REG+120 ;
4047 016056 152777 000010 173734  BISB     #10,@ML.REG+120 ;
4048 016064 016600 000012      MOV      12(SP),R0      ; INDEX,*
4049 016070 006300      ASL      R0
4050 016072 006300      ASL      R0
4051 016074 006300      ASL      R0
4052 016076 010001      MOV      R0,R1
4053 016100 016100 011702      MOV      ML.REG+2(R1),R0
4054 016104 056600 000014      BIS      14(SP),R0      ; TST.PAT,*
4055 016110 046100 011704      BIC      ML.REG+4(R1),R0
4056 016114 016102 011706      MOV      ML.REG+6(R1),R2
4057 016120 050002      BIS      R0,R2
4058 016122 010267 173542      MOV      R2,WT_DATA
4059 016126 017702 173776      MOV      @ML.REG+230,R2 ;
4060 016132 056102 011706      BIS      ML.REG+6(R1),R2
4061 016136 010267 173530      MOV      R2,RD_DATA
4062 016142 026767 173522 173522  CMP      WT_DATA,RD_DATA ;
4063 016150 001403      BEQ
4064 016152 012776 000001 000010 1$:  MOV      #1,@10(SP)      ; *,ERR.FLG
4065 016160 142777 000100 173632  BICB     #100,@ML.REG+120 ;
4066 016166 142777 000010 173624  BICB     #10,@ML.REG+120 ;
4067 016174 000207      RTS      PC          ;
4068 :
4069 :
4070 :
; Routine Size: 47 words
; Maximum stack depth per invocation: 3 words
  
```

4076 ;ML4
4077 ;
4078
4079 :
4080 :
4081 :
4082 :
4083 :
4084 :
4085 :
4086 :
4087 :
4088 :
4089 :
4090 :
4091 :
4092 :
4093 :
4094 :
4095 :
4096 :
4097 :
4098 :
4099 :
4100 :
4101 :
4102 :
4103 :
4104 :
4108
4112 016176 000207
4113
4114
4115
4120
4121

22-Oct-1980 10:47:44
22-Oct-1980 10:45:32

TOPS-20 Bliss-16 V2(206)
PA:<NEALE>BL2ML4.BLI.2 (30)

```
3135 routine WRT_EL (TST_PAT, index) : novalue =  
3136 begin  
3137  
3138 !++  
3139 ! FUNCTIONAL DESCRIPTION:  
3140 ! DUMMY ROUTINE CALL TO ASSIST IN THE READ  
3141 ! WRITE REGISTER ALGORITHM  
3142  
3143 ! FORMAL PARAMETERS:  
3144 ! TST_PAT  
3145 ! DATA PATTERN TO BE MASKED AND  
3146 ! COMPARED AGAINST THE CONTENTS  
3147 ! OF THE REGISTER UNDER TEST  
3148  
3149 ! INDEX  
3150 ! USED BY THE MACRO WRT_MASK TO  
3151 ! SELECT THE CURRENT REGISTER ADDRESS,  
3152 ! FORCED HI, FORCED LO AND DON'T CARE  
3153 ! MASK INFORMATION.  
3154 !--  
3155 !  
3156 !  
3157 !  
3158 ! ERROR LOCATION REG IS READ ONLY  
3159 return;  
3160 end;
```

WRT.EL: RTS PC ;
; Routine Size: 1 word
; Maximum stack depth per invocation: 0 words

3135

4123 :ML4
4124 :
4125 :
4126 :
4127 :
4128 :
4129 :
4130 :
4131 :
4132 :
4133 :
4134 :
4135 :
4136 :
4137 :
4138 :
4139 :
4140 :
4141 :
4142 :
4143 :
4144 :
4145 :
4146 :
4147 :
4148 :
4149 :
4150 :
4151 :
4152 :
4153 :
4154 :
4155 :
4156 :
4157 :
4158 :
4159 :
4160 :
4161 :
4162 :
4163 :
4164 :
4165 :
4166 :
4167 :
4168 :
4169 :
4170 :
4171 :
4172 :
4173 :
4174 :
4175 :
4176 :
4177 :

22-Oct-1980 10:47:44
22-Oct-1980 10:45:32

TOPS-20 Bliss-16 V2(206)
PA:<NEALE>BL2ML4.BLI.2 (31)

```
3161 routine RD_EL (TST_PAT, index, ERR_FLG) : novalue =
3162 begin
3163
3164 !++
3165 FUNCTIONAL DESCRIPTION:
3166
3167 COMPARES THE CONTENTS OF THE
3168 ERROR LOCATION REGISTER
3169 WITH THE CONTENTS OF TST_PAT.
3170
3171 IF THE COMPARE IS NOT EQUAL THEN
3172 THE FORMAL PARAMETER 'ERR FLG' IS
3173 ASSIGNED A ONE TO INDICATE THE
3174 ERROR
3175
3176 FORMAL PARAMETERS:
3177 TST_PAT
3178 DATA PATTERN TO BE MASKED AND
3179 COMPARED AGAINST THE CONTENTS
3180 OF THE REGISTER UNDER TEST
3181
3182 INDEX
3183 USED BY THE MACRO WRT MASK TO
3184 SELECT THE CURRENT REGISTER ADDRESS,
3185 FORCED HI, FORCED LO AND DON'T CARE
3186 MASK INFORMATION.
3187
3188 ERR_FLG
3189 CONTAINS THE ADDRESS (PASSED BY REF)
3190 OF THE CALLERS ERROR FLG TO ENABLE THE
3191 CALLER TO EXAMINE THE ERROR STATUS
3192 OF THE ROUTINE CALL.
3193
3194 IMPLICIT INPUTS:
3195 WT_DATA
3196 GETS LOADED WITH THE GENERATED
3197 WRT_MASK DATA PATTERN THUS ALLOWING
3198 CALLER TO PRINT FAILING GOOD DATA.
3199
3200 RD_DATA
3201 GETS LOADED WITH DATA READ FROM THE
3202 REGISTER THUS ALLOWING CALLER
3203 TO PRINT FAILING BAD DATA.
3204
3205 IMPLICIT OUTPUTS:
3206 GLOBAL LOCATION WR_DATA
3207 AND RD_DATA LOADED WITH GOOD
3208 AND BAD REGISTER DATA
3209
3210 --
3211
3212 .ERR_FLG = ZERO;
```


4179 ;ML4
4180 ;
4181 ;
4182 : 3213
4183 : 3214
4184 : 3215
4185 : 3216
4186 : 3217
4187 : 3218
4191
4195 016200 005076 000002
4196 016204 016667 000006 173456
4197 016212 017767 173702 173452
4198 016220 026767 173446 173442
4199 016226 001403
4200 016230 012776 000001 000002
4201 016236 000207
4202
4203
4204
4209
4210

WT_DATA = .TST_PAT;
RD_DATA = .MLEC;
if .RD_DATA neq .WT_DATA then .ERR_FLG = ONE;
end;

22-Oct-1980 10:47:44 TOPS-20 Bliss-16 V2(206)
22-Oct-1980 10:45:32 PA:<NEALE>BL2ML4.BLI.2 (31)

!SAVE THE DATA WRITTEN TO THE REGISTER
!READ AND SAVE THE REGISTER
!READ REGISTER FOR WT_DATA

RD.EL:	CLR	@2(SP)			; ERR.FLG	3212
	MOV	6(SP),WT_DATA			; TST.PAT,*	3213
	MOV	@ML.REG+220,RD.DATA				3214
	CMP	RD.DATA,WT_DATA				3216
	BEQ	1\$				
	MOV	#1,@2(SP)			; *,ERR.FLG	
1\$:	RTS	PC				3161

; Routine Size: 16 words
; Maximum stack depth per invocation: 0 words

4212 :ML4
4213 :
4214 :
4215 :
4216 :
4217 :
4218 :
4219 :
4220 :
4221 :
4222 :
4223 :
4224 :
4225 :
4226 :
4227 :
4228 :
4229 :
4230 :
4231 :
4232 :
4233 :
4234 :
4235 :
4236 :
4237 :
4238 :
4239 :
4243 :
4247 :
4248 :
4249 :
4250 :
4255 :
4256 :

22-Oct-1980 10:47:44
22-Oct-1980 10:45:32

TOPS-20 Bliss-16 V2(206)
PA:<NEALE>BL2ML4.BLI.2 (32)

```
3219 routine WRT_EE (TST_PAT, index) : novalue =
3220     begin
3221
3222     !++
3223     FUNCTIONAL DESCRIPTION:
3224     DUMMY ROUTINE CALL TO ASSIST IN THE READ
3225     WRITE REGISTER ALGORITHM
3226
3227     FORMAL PARAMETERS:
3228     TST_PAT
3229     DATA PATTERN TO BE MASKED AND
3230     COMPARED AGAINST THE CONTENTS
3231     OF THE REGISTER UNDER TEST
3232
3233     INDEX
3234     USED BY THE MACRO WRT_MASK TO
3235     SELECT THE CURRENT REGISTER ADDRESS,
3236     FORCED HI, FORCED LO AND DON'T CARE
3237     MASK INFORMATION.
3238
3239     !--
3240
3241     ! ECC ERROR REGISTER IS READ ONLY
3242     return;
3243     end;
```

016240 000207

WRT.EE: RTS PC ;

3219

; Routine Size: 1 word
; Maximum stack depth per invocation: 0 words

22-Oct-1980 10:47:44
 22-Oct-1980 10:45:32

TOPS-20 Bliss-16 V2(206)
 PA:<NEALE>BL2ML4.BLI.2 (33)

```

4258 :ML4
4259 :
4260 :
4261 : 3244 routine RD_EE (TST_PAT, index, ERR_FLG) : novalue =
4262 : 3245     begin
4263 : 3246
4264 : 3247 !++
4265 : 3248 ! FUNCTIONAL DESCRIPTION:
4266 : 3249
4267 : 3250     COMPARES THE CONTENTS OF THE
4268 : 3251     ECC ERROR REGISTER WITH TST_PAT.
4269 : 3252
4270 : 3253     IF THE COMPARE IS NOT EQUAL THEN
4271 : 3254     THE FORMAL PARAMETER 'ERR_FLG' IS
4272 : 3255     ASSIGNED A ONE TO INDICATE THE
4273 : 3256     ERROR.
4274 : 3257
4275 : 3258 ! FORMAL PARAMETERS:
4276 : 3259     TST_PAT
4277 : 3260     DATA PATTERN TO BE
4278 : 3261     COMPARED AGAINST THE CONTENTS
4279 : 3262     OF THE REGISTER UNDER TEST.
4280 : 3263
4281 : 3264     INDEX
4282 : 3265     USED BY THE MACRO MLEE TO
4283 : 3266     SELECT THE CURRENT REGISTER'S ADDRESS.
4284 : 3267
4285 : 3268     ERR_FLG
4286 : 3269     CONTAINS THE ADDRESS (PASSED
4287 : 3270     BY REF) OF THE CALLER'S ERROR_FLG TO ENABLE
4288 : 3271     THE CALLER TO EXAMINE THE ERROR STATUS
4289 : 3272     OF THE ROUTINE CALL.
4290 : 3273
4291 : 3274 ! IMPLICIT INPUTS:
4292 : 3275     WT_DATA
4293 : 3276     GETS LOADED WITH THE TST_PAT
4294 : 3277     THUS ALLOWING CALLER TO PRINT
4295 : 3278     THE FAILING DATA.
4296 : 3279
4297 : 3280     RD_DATA
4298 : 3281     GETS LOADED WITH DATA READ FROM THE
4299 : 3282     REGISTER THUS ALLOWING CALLER
4300 : 3283     TO PRINT FAILING BAD DATA.
4301 : 3284
4302 : 3285 ! IMPLICIT OUTPUTS:
4303 : 3286     GLOBAL LOCATION WR_DATA
4304 : 3287     AND RD_DATA LOADED WITH GOOD
4305 : 3288     AND BAD REGISTER DATA
4306 : 3289
4307 : 3290 !--
4308 : 3291
4309 : 3292     .ERR_FLG = ZERO;
4310 : 3293     WT_DATA = .TST_PAT;
4311 : 3294     RD_DATA = .MLEE;
4312 : 3295
  
```

```

!CLEAR THE ERROR FLAG
!SAVE DATA WRITTEN TO THE REGISTER
!READ AND SAVE THE REGISTER
  
```

22-Oct-1980 10:47:44 TOPS-20 Bliss-16 V2(206)
22-Oct-1980 10:45:32 PA:<NEALE>BL2ML4.BLI.2 (33)

```
4314 ;ML4
4315 ;
4316
4317 :          3296      *f .RD_DATA neq .WT_DATA then .ERR_FLG = ONE;      !READ MLEE FOR TST_PAT
4318 :          3297
4319 :          3298      !SET THE ERROR FLAG IF NEQ
4320 :          3299      end;
4324
4328 016242 005076 000002          RD.EE: CLR      @2(SP)          ; ERR.FLG          3292
4329 016246 016667 000006 173414      MOV      6(SP),WT.DATA      ; TST.PAT,*      3293
4330 016254 017767 173630 173410      MOV      @ML.REG+210,RD.DATA ;                   3294
4331 016262 026767 173404 173400      CMP      RD.DATA,WT.DATA    ;                   3296
4332 016270 001403          BEQ      1$
4333 016272 012776 000001 000002      MOV      #1,@2(SP)          ; *,ERR.FLG
4334 016300 000207          1$: RTS      PC              ;                   3244
4335
4336          ; Routine Size: 16 words
4337          ; Maximum stack depth per invocation: 0 words
4342
4343
```

22-Oct-1980 10:47:44 TOPS-20 Bliss-16 V2(206)
22-Oct-1980 10:45:32 PA:<NEALE>BL2ML4.BLI.2 (34)

4345 ;ML4
4346 ;
4347 ;
4348 ;
4349 ;
4350 ;
4351 ;
4352 ;
4353 ;
4354 ;
4355 ;
4356 ;
4357 ;
4358 ;
4359 ;
4360 ;
4361 ;
4362 ;
4363 ;
4364 ;
4365 ;
4369 ;
4373
4374
4375
4376
4377
4378
4379
4380
4381
4382
4383
4384
4385
4386
4387
4388
4389
4390
4395
4396

```

3300 routine WRT_D1 (TST_PAT, index) : novalue =
3301   begin
3302
3303   !++
3304   !FUNCTIONAL DESCRIPTION:
3305   !LOADS THE DATA DIAG REG 1 WITH A DATA PATTERN GENERATED BY THE MACRO WRT_MASK
3306   !FORMAL PARAMETERS:
3307   !TST_PAT
3308   !CURRENT DATA PATTERN TO BE LOADED IN THE REGISTER.
3309   !INDEX
3310   !USED BY THE MACRO WRT_MASK TO SELECT THE CURRENT REGISTERS ADDRESS,
3311   !FORCED HI, FORCED LO AND DON'T CARE MASK INFORMATION.
3312   !--
3313
3314   DAT_DM = ONE;
3315   MLD1 = WRT_MASK;
3316   DAT_DM = ZERO;
3317   end;
  
```

```

!SET DATA DIAG MODE
!LOAD MLD1 WITH GENERATED WRT_MASK PATTERN
!CLEAR DATA DIAG MODE
  
```

```

WRT.D1: JSR      R1,$SAVE2          ;          3300
        BISB    #10,@ML.REG+120   ;          3314
        MOV     10(SP),R0         ; INDEX,*  3315
        ASL    R0
        ASL    R0
        ASL    R0
        MOV    R0,R1
        MOV    ML.REG+2(R1),R0
        BIS    12(SP),R0         ; TST.PAT,*
        BIC    ML.REG+4(R1),R0
        MOV    ML.REG+6(R1),R2
        BIS    R0,R2
        MOV    R2,@ML.REG+170
        BICB   #10,@ML.REG+120   ;          3316
        RTS    PC                ;          3300
  
```

```

; Routine Size: 26 words
; Maximum stack depth per invocation: 3 words
  
```

22-Oct-1980 10:47:44
22-Oct-1980 10:45:32

TOPS-20 Bliss-16 V2(206)
PA:<NEALE>BL2ML4.BLI.2 (35)

4398 :ML4
4399 :
4400 :
4401 :
4402 :
4403 :
4404 :
4405 :
4406 :
4407 :
4408 :
4409 :
4410 :
4411 :
4412 :
4413 :
4414 :
4415 :
4416 :
4417 :
4418 :
4419 :
4420 :
4421 :
4422 :
4423 :
4424 :
4425 :
4426 :
4427 :
4428 :
4429 :
4430 :
4431 :
4432 :
4433 :
4434 :
4435 :
4436 :
4437 :
4438 :
4439 :
4440 :
4441 :
4442 :
4443 :
4444 :
4445 :
4446 :
4447 :
4448 :
4449 :
4450 :
4451 :
4452 :

```
3318 routine RD_D1 (TST_PAT, index, ERR_FLG) : novalue =
3319   begin
3320
3321   !++
3322   FUNCTIONAL DESCRIPTION:
3323
3324   COMPARES THE CONTENTS OF THE
3325   DATA DIAG REGISTER 1
3326   WITH THE MASKED DATA PATTERN
3327   GENERATED BY THE MACRO 'WRT_MASK'.
3328
3329   IF THE COMPARE IS NOT EQUAL THEN
3330   THE FORMAL PARAMETER 'ERR_FLG' IS
3331   ASSIGNED A ONE TO INDICATE THE
3332   ERROR.
3333
3334   FORMAL PARAMETERS:
3335
3336   TST PAT
3337   DATA PATTERN TO BE MASKED AND
3338   COMPARED AGAINST THE CONTENTS
3339   OF THE REGISTER UNDER TEST.
3340
3341   INDEX
3342   USED BY THE MACRO WRT_MASK TO
3343   SELECT THE CURRENT REGISTER ADDRESS,
3344   FORCED HI, FORCED LO AND DON'T CARE
3345   MASK INFORMATION.
3346
3347   ERR_FLG
3348   CONTAINS THE ADDRESS (PASSED BY REF)
3349   OF THE CALLERS ERROR_FLG TO ENABLE THE
3350   CALLER TO EXAMINE THE ERROR STATUS
3351   OF THE ROUTINE CALL.
3352
3353   IMPLICIT INPUTS:
3354   WT DATA
3355   GETS LOADED WITH THE GENERATED
3356   WRT_MASK DATA PATTERN THUS ALLOWING
3357   CALLER TO PRINT FAILING GOOD DATA.
3358
3359   RD DATA
3360   GETS LOADED WITH DATA READ FROM THE
3361   REGISTER THUS ALLOWING CALLER
3362   TO PRINT FAILING BAD DATA.
3363
3364   IMPLICIT OUTPUTS:
3365   GLOBAL LOCATION WR_DATA
3366   AND RD DATA LOADED WITH GOOD
3367   AND BAD REGISTER DATA
3368
3369   !--
```

22-Oct-1980 10:47:44 TOPS-20 Bliss-16 V2(206)
22-Oct-1980 10:45:32 PA:<NEALE>BL2ML4.BLI.2 (35)

```

4454 ;ML4
4455 :
4456 :
4457 : 3370
4458 : 3371 .ERR_FLG = ZERO; !CLEAR THE ERROR FLG
4459 : 3372 DAT_DM = ONE; !SET DATA DIAG MODE
4460 : 3373 ML_FUNC = write; !LOAD MLCS1 WITH WRITE FUNCTION
4461 : 3374 DAT_CLK = ONE; !DO A DATA CLK
4462 : 3375
4463 : 3376 if .REG_INIT_FLG IS_SET !SEE IF CALLER IS REG INIT TEST
4464 : 3377 !SET ERROR FLAG IF NEQ
4465 : 3378 then
4466 : 3379 begin
4467 : 3380 CLR_MBUS; !CLEAR MBUS TO GENERATE INIT DATA
4468 : 3381 DAT_DM = ONE;
4469 : 3382 REG_INIT_FLG = ZERO;
4470 : 3383 end;
4471 : 3384
4472 : 3385 WT_DATA = WRT_MASK; !SAVE THE DATA WRITTEN TO THE REGISTER
4473 : 3386 RD_DATA = .MLD1; !READ AND SAVE THE REGISTER
4474 : 3387
4475 : 3388 if .RD_DATA neq .WT_DATA then .ERR_FLG = ONE; !READ MLD1 FOR WRT_MASK
4476 : 3389
4477 : 3390 !SET ERR_FLG IF NEQ
4478 : 3391 CLR_MBUS; !CLEAR THE MBUS
4479 : 3392 end;
4483 :
4487 016366 004167 165414 RD.D1: JSR R1,$SAVE2 ; 3318
4488 016372 005076 000010 CLR @10(SP) ; ERR.FLG 3371
4489 016376 152777 000010 173414 BISB #10,@ML.REG+120 ; 3372
4490 016404 142777 000077 173266 BICB #77,@ML.REG ; 3373
4491 016412 152777 000061 173260 BISB #61,@ML.REG
4492 016420 152777 000020 173372 BISB #20,@ML.REG+120 ; 3374
4493 016426 026727 173244 000001 CMP REG.INIT.FLG,#1 ; 3376
4494 016434 001021 BNE 1$
4495 016436 152777 000040 173274 BISB #40,@ML.REG+40 ; 3379
4496 016444 016701 173520 MOV ML.DUT,R1
4497 016450 042701 177770 BIC #177770,R1
4498 016454 142777 000007 173256 BICB #7,@ML.REG+40
4499 016462 150177 173252 BISB R1,@ML.REG+40
4500 016466 152777 000010 173324 BISB #10,@ML.REG+120 ; 3381
4501 016474 005067 173176 CLR REG.INIT.FLG ; 3382
4502 016500 016600 000012 1$: MOV 12(SP),R0 ; INDEX,* 3385
4503 016504 006300 ASL R0
4504 016506 006300 ASL R0
4505 016510 006300 ASL R0
4506 016512 010001 MOV R0,R1
4507 016514 016100 011702 MOV ML.REG+2(R1),R0

```

```
4509          ;ML4
4510          ;
4511
4512 016520 056600 000014      BIS      14(SP),R0      ; TST.PAT,*
4513 016524 046100 011704      BIC      ML.REG+4(R1),R0
4514 016530 016102 011706      MOV      ML.REG+6(R1),R2
4515 016534 050002      BIS      R0,R2
4516 016536 010267 173126      MOV      R2,WT.DATA
4517 016542 017767 173322 173122      MOV      @ML.REG+170,RD.DATA      ;
4518 016550 026767 173116 173112      CMP      RD.DATA,WT.DATA      ;
4519 016556 001403      BEQ      2$
4520 016560 012776 000001 000010      MOV      #1,@10(SP)      ; *,ERR.FLG
4521 016566 152777 000040 173144 2$:      BISB     #40,@ML.REG+40
4522 016574 016702 173370      MOV      ML.DUT,R2
4523 016600 042702 177770      BIC      #177770,R2
4524 016604 142777 000007 173126      BICB     #7,@ML.REG+40
4525 016612 150277 173122      BISB     R2,@ML.REG+40
4526 016616 000207      RTS      PC      ;
4527
4528          ; Routine Size: 77 words
4529          ; Maximum stack depth per invocation: 3 words
4534
4535
```

22-Oct-1980 10:47:44 TOPS
22-Oct-1980 10:45:32 PA:<

3386
3388

3318

22-Oct-1980 10:47:44 TOPS-20' Bliss-16 V2(206)
22-Oct-1980 10:45:32 PA:<NEALE>BL2ML4.BLI.2 (36)

4537 :ML4

4538 :
4539 :
4540 : 3393
4541 : 3394
4542 : 3395
4543 : 3396
4544 : 3397
4545 : 3398
4546 : 3399
4547 : 3400
4548 : 3401
4549 : 3402
4550 : 3403
4551 : 3404
4552 : 3405
4553 : 3406
4554 : 3407
4555 : 3408
4556 : 3409
4557 : 3410

routine WRT_D2 (TST_PAT, index) : novalue =
begin

!++
FUNCTIONAL DESCRIPTION:
LOADS THE DATA DIAG REG 2 WITH A DATA PATTERN GENERATED BY THE MACRO WRT_MASK
FORMAL PARAMETERS:
TST PAT
CURRENT DATA PATTERN TO BE LOADED IN THE REGISTER.
INDEX
USED BY THE MACRO WRT_MASK TO SELECT THE CURRENT REGISTERS ADDRESS,
FORCED HI, FORCED LO AND DON'T CARE MASK INFORMATION.
!--

DAT_DM = ONE; !SET DATA DIAG MODE
MLD2 = WRT_MASK; !LOAD MLD2 WITH GENERATED WRT_MASK PATTERN
DAT_DM = ZERO; !CLEAR DATA DIAG MODE
end;

4561
4565 016620 004167 165162 WRT.D2: JSR R1,\$SAVE2 ; 3393
4566 016624 152777 000010 173166 BISB #10,@ML.REG+120 ; 3407
4567 016632 016600 000010 MOV 10(SP),R0 ; INDEX,* 3408
4568 016636 006300 ASL R0
4569 016640 006300 ASL R0
4570 016642 006300 ASL R0
4571 016644 010001 MOV R0,R1
4572 016646 016100 011702 MOV ML.REG+2(R1),R0
4573 016652 056600 000012 BIS 12(SP),R0 ; TST.PAT,*
4574 016656 046100 011704 BIC ML.REG+4(R1),R0
4575 016662 016102 011706 MOV ML.REG+6(R1),R2
4576 016666 050002 BIS R0,R2
4577 016670 010277 173204 MOV R2,@ML.REG+200
4578 016674 142777 000010 173116 BICB #10,@ML.REG+120 ; 3409
4579 016702 000207 RTS PC ; 3393

; Routine Size: 26 words
; Maximum stack depth per invocation: 3 words

4580
4581
4582
4587
4588

22-Oct-1980 10:47:44
22-Oct-1980 10:45:32

TOPS-20 Bliss-16 V2(206)
PA:<NEALE>BL2ML4.BLI.2 (37)

4590 :ML4
4591 :
4592 :
4593 :
4594 :
4595 :
4596 :
4597 :
4598 :
4599 :
4600 :
4601 :
4602 :
4603 :
4604 :
4605 :
4606 :
4607 :
4608 :
4609 :
4610 :
4611 :
4612 :
4613 :
4614 :
4615 :
4616 :
4617 :
4618 :
4619 :
4620 :
4621 :
4622 :
4623 :
4624 :
4625 :
4626 :
4627 :
4628 :
4629 :
4630 :
4631 :
4632 :
4633 :
4634 :
4635 :
4636 :
4637 :
4638 :
4639 :
4640 :
4641 :
4642 :
4643 :
4644 :

```
3411 routine RD_D2 (TST_PAT, index, ERR_FLG) : novalue =  
3412 begin  
3413  
3414 !++  
3415 FUNCTIONAL DESCRIPTION:  
3416  
3417 COMPARE THE CONTENTS OF THE  
3418 DATA DIAG REGISTER 2  
3419 WITH THE MASKED DATA PATTERN  
3420 GENERATED BY THE MACRO 'WRT_MASK'.  
3421 IF THE COMPARE IS NOT EQUAL THEN THE  
3422 FORMAL PARAMETER 'ERR_FLG' IS  
3423 ASSIGNED A ONE TO INDICATE THE  
3424 ERROR.  
3425  
3426 FORMAL PARAMETERS:  
3427  
3428 TST_PAT  
3429 DATA PATTERN TO BE MASKED AND  
3430 COMPARED AGAINST THE CONTENTS  
3431 OF THE REGISTER UNDER TEST.  
3432  
3433 INDEX  
3434 USED BY THE MACRO WRT_MASK TO  
3435 SELECT THE CURRENT REGISTER ADDRESS,  
3436 FORCED HI, FORCED LO AND DON'T CARE  
3437 MASK INFORMATION.  
3438  
3439 ERR_FLG  
3440 CONTAINS THE ADDRESS (PASSED BY REF)  
3441 OF THE CALLERS ERROR FLG TO ENABLE THE  
3442 CALLER TO EXAMINE THE ERROR STATUS  
3443 OF THE ROUTINE CALL.  
3444  
3445 IMPLICIT INPUTS:  
3446 WT_DATA  
3447 GETS LOADED WITH THE GENERATED  
3448 WRT_MASK DATA PATTERN THUS ALLOWING  
3449 CALLER TO PRINT FAILING GOOD DATA.  
3450  
3451 RD_DATA  
3452 GETS LOADED WITH DATA READ FROM THE  
3453 REGISTER THUS ALLOWING CALLER  
3454 TO PRINT FAILING BAD DATA.  
3455  
3456 IMPLICIT OUTPUTS:  
3457 GLOBAL LOCATION WR_DATA  
3458 AND RD_DATA LOADED WITH GOOD  
3459 AND BAD REGISTER DATA  
3460  
3461 --  
3462
```

22-Oct-1980 10:47:44 TOPS-20 Bliss-16 V2(206)
 22-Oct-1980 10:45:32 PA:<NEALE>BL2ML4.BLI.2 (37)

```

4646 :ML4
4647 :
4648 :
4649 :      3463      .ERR_FLG = ZERO;
4650 :      3464      DAT_DM = ONE;
4651 :      3465      ML_FUNC = write;
4652 :      3466      DAT_CLK = ONE;
4653 :      3467
4654 :      3468      if .REG_INIT_FLG IS_SET
4655 :      3469      then
4656 :      3470          begin
4657 :      3471              CLR_MBUS;
4658 :      3472              DAT_DM = ONE;
4659 :      3473              REG_INIT_FLG = ZERO;
4660 :      3474          end;
4661 :      3475
4662 :      3476      WT_DATA = WRT_MASK;
4663 :      3477      RD_DATA = .MLD2;
4664 :      3478
4665 :      3479      if .RD_DATA neq .WT_DATA then .ERR_FLG = ONE;
4666 :      3480
4667 :      3481
4668 :      3482      CLR_MBUS;
4669 :      3483      end;
4670 :
4671 :
4672 :
4673 :
4674 :
4675 :
4676 :
4677 016704 004167 165076      RD.D2: JSR      R1,$SAVE2
4678 016710 005076 000010      CLR      @10(SP)
4679 016714 152777 000010 173076      BISB    #10,@ML.REG+120
4680 016722 142777 000077 172750      BICB    #77,@ML.REG
4681 016730 152777 000061 172742      BISB    #61,@ML.REG
4682 016736 152777 000020 173054      BISB    #20,@ML.REG+120
4683 016744 026727 172726 000001      CMP     REG.INIT.FLG,#1
4684 016752 001021      BNE     1$
4685 016754 152777 000040 172756      BISB    #40,@ML.REG+40
4686 016762 016701 173202      MOV     ML.DUT,R1
4687 016766 042701 177770      BIC     #177770,R1
4688 016772 142777 000007 172740      BICB    #7,@ML.REG+40
4689 017000 150177 172734      BISB    R1,@ML.REG+40
4690 017004 152777 000010 173006      BISB    #10,@ML.REG+120
4691 017012 005067 172660      CLR     REG.INIT.FLG
4692 017016 016600 000012      1$:    MOV     12(SP),R0
4693 017022 006300      ASL     R0
4694 017024 006300      ASL     R0
4695 017026 006300      ASL     R0
4696 017030 010001      MOV     R0,R1
4697 017032 016100 011702      MOV     ML.REG+2(R1),R0
4698 017036 056600 000014      BIS     14(SP),R0
4699 017042 046100 011704      BIC     ML.REG+4(R1),R0

```

```

!CLEAR ERROR FLAG
!SET DATA DIAG MODE
!LOAD WRITE FUNC TO CS1
!DO A DATA CLOCK

!SEE IF CALLER IS REG INIT TEST
!CLEAR MBUS TO GENERATE INIT DATA

```

```

!SAVE THE DATA WRITTEN TO THE REGISTER
!READ AND SAVE THE REGISTER

```

```

!READ MLD2 FR WRT_MASK PATTERN

```

```

!SET ERROR FLAG IF NEQ
!CLR MASS BUSS

```

```

3411
3463
3464
3465
3466
3468
3470
3472
3473
3476

```

```

; TST.PAT,*

```

22-Oct-1980 10:47:44 TOPS
22-Oct-1980 10:45:32 PA:<

```
4701          ;ML4
4702          ;
4703
4704 017046 016102 011706          MOV    ML.REG+6(R1),R2
4705 017052 050002                BIS    R0,R2
4706 017054 010267 172610          MOV    R2,WT.DATA
4707 017060 017767 173014 172604  MOV    @ML.REG+200,RD.DATA          ;
4708 017066 026767 172600 172574  CMP    RD.DATA,WT.DATA          ;
4709 017074 001403                BEQ    2$
4710 017076 012776 000001 000010  MOV    #1,@10(SP)                ; *,ERR.FLG
4711 017104 152777 000040 172626 2$: BISB  #40,@ML.REG+40
4712 017112 016702 173052          MOV    ML.DUT,R2
4713 017116 042702 177770          BIC    #177770,R2
4714 017122 142777 000007 172610  BICB  #7,@ML.REG+40
4715 017130 150277 172604          BISB  R2,@ML.REG+40
4716 017134 000207                RTS   PC          ;
4717
4718          ; Routine Size: 77 words
4719          ; Maximum stack depth per invocation: 3 words
4724
4725
```

3477

3479

3411

22-Oct-1980 10:47:44
22-Oct-1980 10:45:32

TOPS-20 Bliss-16 V2(206)
PA:<NEALE>BL2ML4.BLI.2 (38)

4727 :ML4
4728 :
4729 :
4730 :
4731 :
4732 :
4733 :
4734 :
4735 :
4736 :
4737 :
4738 :
4739 :
4740 :
4741 :
4742 :
4743 :
4744 :
4745 :
4746 :
4747 :
4748 :
4749 :
4750 :
4751 :
4752 :
4753 :
4754 :
4755 :
4756 :

```

3484 routine WRT_D3 (TST_PAT, index) : novalue =
3485     begin
3487     ++
3488     FUNCTIONAL DESCRIPTION:
3489
3490         LOADS THE DATA DIAG
3491         REGISTER 3 WITH A DATA PATTERN
3492         GENERATED BY THE MACRO
3493         WRT MASK.
3495     FORMAL PARAMETERS:
3497         TST PAT
3498         CURRENT DATA PATTERN TO BE
3499         LOADED IN THE REGISTER.
3501         INDEX
3502         USE BY THE MACRO WRT_MASK
3503         TO SELECT THE CURRENT REGISTERS
3504         ADDRESS, FORCED HI, FORCED
3505         LO AND DON'T CARE MASK
3506         INFORMATION.
3507
3508     --
3510     DAT_DM = ONE;
3511     MLE2_MASK = %o'000377';
3512     MLE2 = WRT_MASK;
3513     MLE2_MASK = %o'100300';
3514     DAT_DM = ZERO;
3515     end;

```

```

!SET DATA DIAG MODE
!MASK OUT ECC CRC WORD BITS
!LOAD MLE2 WITH GENERATED WRT_MASK PATTERN
!RESTORE MASK
!CLEAR DATA DIAG MODE.

```

4764	017136	004167	164644		WRT.D3:	JSR	R1,\$SAVE2	:		3484
4765	017142	152777	000010	172650		BISB	#10,@ML.REG+120	:		3510
4766	017150	012767	000377	172710		MOV	#377,ML.REG+166	:		3511
4767	017156	016600	000010			MOV	10(SP),R0	:	INDEX,*	3512
4768	017162	006300				ASL	R0			
4769	017164	006300				ASL	R0			
4770	017166	006300				ASL	R0			
4771	017170	010001				MOV	R0,R1			
4772	017172	016100	011702			MOV	ML.REG+2(R1),R0			
4773	017176	056600	000012			BIS	12(SP),R0	:	TST.PAT,*	
4774	017202	046100	011704			BIC	ML.REG+4(R1),R0			
4775	017206	016102	011706			MOV	ML.REG+6(R1),R2			
4776										
4777										
4778										
4779	017212	050002				BIS	R0,R2			
4780	017214	010277	172640			MOV	R2,@ML.REG+160			
4781	017220	012767	100300	172640		MOV	#-77500,ML.REG+166	:		3513
4782	017226	142777	000010	172564		BICB	#10,@ML.REG+120	:		3514
4783	017234	000207				RTS	PC	:		3484
4788										
4789										

22-Oct-1980 10:47:44 TOPS
22-Oct-1980 10:45:32 PA:<

4791 :ML4
4792 :
4793 :
4794 :
4795 :
4796 :
4797 :
4798 :
4799 :
4800 :
4801 :
4802 :
4803 :
4804 :
4805 :
4806 :
4807 :
4808 :
4809 :
4810 :
4811 :
4812 :
4813 :
4814 :
4815 :
4816 :
4817 :
4818 :
4819 :
4820 :
4821 :
4822 :
4823 :
4824 :
4825 :
4826 :
4827 :
4828 :
4829 :
4830 :
4831 :
4832 :
4833 :
4834 :
4835 :
4836 :
4837 :
4838 :
4839 :
4840 :
4841 :
4842 :
4843 :
4844 :
4845 :

22-Oct-1980 10:47:44
22-Oct-1980 10:45:32

TOPS-20 Bliss-16 V2(206)
PA:<NEALE>BL2ML4.BLI.2 (39)

```
3516 routine RD_D3 (TST_PAT, index, ERR_FLG) : novalue =
3517   begin
3518
3519   !++
3520   FUNCTIONAL DESCRIPTION:
3521
3522       COMPARES THE CONTENTS OF THE
3523       DATA DIAG REGISTER 3
3524       WITH THE MASKED DATA PATTERN
3525       GENERATED BY THE MACRO 'WRT_MASK'.
3526
3527       IF THE COMPARE IS NOT EQUAL THE
3528       FORMAL PARAMETER 'ERR_FLG' IS
3529       ASSIGNED A ONE TO INDICATE THE
3530       ERROR.
3531
3532   FORMAL PARAMETERS:
3533
3534       TST_PAT
3535       DATA PATTERN TO BE MASKED AND
3536       COMPARED AGAINST THE CONTENTS
3537       OF THE REGISTER UNDER TEST.
3538
3539       INDEX
3540       USED BY THE MACRO WRT_MASK TO
3541       SELECT THE CURRENT REGISTERS ADDRESS,
3542       FORCED HI, FORCED LO AND DON'T CARE
3543       MASK INFORMATION
3544
3545       ERR_FLG
3546       CONTAINS THE ADDRESS (PASSED BY REF)
3547       OF THE CALLERS ERROR FLG TO ENABLE THE
3548       CALLER TO EXAMINE THE ERROR STATUS
3549       OF THE ROUTINE CALL.
3550
3551   IMPLICIT INPUTS:
3552       WT_DATA
3553       GETS LOADED WITH THE GENERATED
3554       WRT_MASK DATA PATTERN THUS ALLOWING
3555       CALLER TO PRINT FAILING GOOD DATA.
3556
3557       RD_DATA
3558       GETS LOADED WITH DATA READ FROM THE
3559       REGISTER THUS ALLOWING CALLER
3560       TO PRINT FAILING BAD DATA.
3561
3562   IMPLICIT OUTPUTS:
3563       GLOBAL LOCATION WR_DATA
3564       AND RD_DATA LOADED WITH GOOD
3565       AND BAD REGISTER DATA
3566
3567   --
```

4847 :ML4
4848 :
4849 :
4850 :
4851 :
4852 :
4853 :
4854 :
4855 :
4856 :
4857 :
4858 :
4859 :
4860 :
4861 :
4862 :
4863 :
4864 :
4865 :
4866 :
4867 :
4868 :
4869 :
4870 :
4871 :
4872 :
4876 :

3568
3569
3570
3571
3572
3573
3574
3575
3576
3577
3578
3579
3580
3581
3582
3583
3584
3585
3586
3587
3588
3589
3590

```
.ERR_FLG = ZERO;
MLE2_MASK = %0'000377';
DAT_DM = ONE;
ML_FUNC = write;
DAT_CLK = ONE;

if .REG_INIT_FLG IS_SET
then
begin
  CLR_MBUS;
  DAT_DM = ONE;
  REG_INIT_FLG = ZERO;
end;

WT_DATA = WRT_MASK;
RD_DATA = .MLE2 or .IGNORE;

if .WT_DATA neq .RD_DATA then .ERR_FLG = ONE;

MLE2_MASK = %0'100300';
CLR_MBUS;
end;
```

22-Oct-1980 10:47:44 TOPS-20 Bliss-16 V2(206)
22-Oct-1980 10:45:32 PA:<NEALE>BL2ML4.BLI.2 (39)

```
!CLEAR ERROR FLAG
!SET DATA DIAG MODE

!LOAD WRITE FUNCTION TO MLCS1
!DO A DATA CLOCK

!SEE IF CALLER IS REG INIT TEST
!CLEAR MBUS TO GENERATE INIT DATA

!SAVE THE DATA WRITTEN TO THE REGISTER
!READ AND SAVE THE REGISTER

!READ THE REG FOR WRT_MASK

!CLEAR THE MASS BUS
```

4880	017236	004167	164544		RD.D3:	JSR	R1,\$SAVE2	:	3516
4881	017242	005076	000010			CLR	@10(SP)	:	3569
4882	017246	012767	000377	172612		MOV	#377,ML.REG+166	:	3570
4883	017254	152777	000010	172536		BISB	#10,@ML.REG+120	:	3571
4884	017262	142777	000077	172410		BICB	#77,@ML.REG	:	3572
4885	017270	152777	000061	172402		BISB	#61,@ML.REG	:	
4886	017276	152777	000020	172514		BISB	#20,@ML.REG+120	:	3573
4887	017304	026727	172366	000001		CMP	REG.INIT.FLG,#1	:	3575
4888	017312	001021				BNE	1\$:	
4889	017314	152777	000040	172416		BISB	#40,@ML.REG+40	:	3577
4890	017322	016701	172642			MOV	ML.DUT,R1	:	
4891	017326	042701	177770			BIC	#177770,R1	:	
4892	017332	142777	000007	172400		BICB	#7,@ML.REG+40	:	
4893	017340	150177	172374			BISB	R1,@ML.REG+40	:	
4894	017344	152777	000010	172446		BISB	#10,@ML.REG+120	:	3579
4895	017352	005067	172320			CLR	REG.INIT.FLG	:	3580
4896	017356	016600	000012		1\$:	MOV	12(SP),R0	:	3583
4897	017362	006300				ASL	R0	:	
4898	017364	006300				ASL	R0	:	
4899	017366	006300				ASL	R0	:	
4900	017370	010001				MOV	R0,R1	:	

```
4902          ;ML4
4903          ;
4904
4905 017372 016100 011702      MOV      ML.REG+2(R1),R0
4906 017376 056600 000014      BIS      14(SP),R0          ; TST.PAT,*
4907 017402 046100 011704      BIC      ML.REG+4(R1),R0
4908 017406 016102 011706      MOV      ML.REG+6(R1),R2
4909 017412 050002              BIS      R0,R2
4910 017414 010267 172250      MOV      R2,WT.DATA
4911 017420 017702 172434      MOV      @ML.REG+160,R2    ;
4912 017424 056102 011706      BIS      ML.REG+6(R1),R2
4913 017430 010267 172236      MOV      R2,RD.DATA
4914 017434 026767 172230 172230  CMP      WT.DATA,RD.DATA    ;
4915 017442 001403              BEQ      2$
4916 017444 012776 000001 000010  MOV      #1,@10(SP)        ; *,ERR.FLG
4917 017452 012767 100300 172406 2$:  MOV      #-77500,ML.REG+166 ;
4918 017460 152777 000040 172252  BISB     #40,@ML.REG+40
4919 017466 016702 172476      MOV      ML.DUT,R2
4920 017472 042702 177770      BIC      #177770,R2
4921 017476 142777 000007 172234  BICB     #7,@ML.REG+40
4922 017504 150277 172230      BISB     R2,@ML.REG+40
4923 017510 000207              RTS      PC                ;
4924
4925          ; Routine Size: 86 words
4926          ; Maximum stack depth per invocation: 3 words
4931
4932
```


4934 ;ML4
4935 :
4936 :
4937 :
4938 :
4939 :
4940 :
4941 :
4942 :
4943 :
4944 :
4945 :
4946 :
4947 :
4948 :
4949 :
4950 :
4951 :
4952 :
4953 :
4954 :
4955 :
4956 :
4957 :
4958 :
4959 :
4960 :
4961 :
4962 :
4966 :
4970 017512 000207
4971 :
4972 :
4973 :
4978 :
4979 :

22-Oct-1980 10:47:44
22-Oct-1980 10:45:32

TJPS-20 Bliss-16 V2(206)
PA:<NEALE>BL2ML4.BLI.2 (40)

```
3591 routine WRT_DS (TST_PAT, index) : novalue =  
3592     begin  
3593  
3594     !++  
3595     FUNCTIONAL DESCRIPTION:  
3596     DUMMY ROUTINE CALL TO ASSIST IN THE READ  
3597     WRITE REGISTER ALGORITHM  
3598  
3599     FORMAL PARAMETERS:  
3600     TST_PAT  
3601     DATA PATTERN TO BE MASKED AND  
3602     COMPARED AGAINST THE CONTENTS  
3603     OF THE REGISTER UNDER TEST  
3604  
3605     INDEX  
3606     USED BY THE MACRO WRT_MASK TO  
3607     SELECT THE CURRENT REGISTERS ADDRESS,  
3608     FORCED HI, FORCED LO AND DON'T CARE  
3609     MASK INFORMATION.  
3610  
3611     !--  
3612  
3613     !  
3614     DRIVE STATUS REG IS READ ONLY  
3615     return;  
3616     end;
```

```
WRT_DS: RTS    PC    ;  
; Routine Size: 1 word  
; Maximum stack depth per invocation: 0 words
```

3591

4981 :ML4
4982 :
4983 :
4984 :
4985 :
4986 :
4987 :
4988 :
4989 :
4990 :
4991 :
4992 :
4993 :
4994 :
4995 :
4996 :
4997 :
4998 :
4999 :
5000 :
5001 :
5002 :
5003 :
5004 :
5005 :
5006 :
5007 :
5008 :
5009 :
5010 :
5011 :
5012 :
5013 :
5014 :
5015 :
5016 :
5017 :
5018 :
5019 :
5020 :
5021 :
5022 :
5023 :
5024 :
5025 :
5026 :
5027 :
5028 :
5029 :
5030 :
5031 :
5032 :
5033 :
5034 :
5035 :

```
3617 routine RD_DS (TST_PAT, index, ERR_FLG) : novalue =  
3618 begin  
3619  
3620 !++  
3621 FUNCTIONAL DESCRIPTION:  
3622  
3623 COMPARES THE CONTENTS OF THE  
3624 DRIVE STATUS REGISTER WITH THE  
3625 MASKED DATA PATTERN  
3626 GENERATED BY THE MACRO 'WRT_MASK'.  
3627  
3628 IF THE COMPARE IS NOT EQUAL THEN  
3629 THE FORMAL PARAMETER 'ERR_FLG' IS  
3630 ASSIGNED A ONE TO INDICATE THE  
3631 ERROR.  
3632  
3633 FORMAL PARAMETERS:  
3634  
3635 TST_PAT  
3636 DATA PATTERN TO BE MASKED AND  
3637 COMPARED AGAINST THE CONTENTS  
3638 OF THE REGISTER UNDER TEST.  
3639  
3640 INDEX  
3641 USED BY THE MACRO WRT_MASK TO  
3642 SELECT THE CURRENT REGISTER ADDRESS,  
3643 FORCED HI, FORCED LO AND DON'T CARE  
3644 MASK INFORMATION.  
3645  
3646 ERR_FLG  
3647 CONTAINS THE ADDRESS (PASSED BY REF)  
3648 OF THE CALLERS ERROR FLG TO ENABLE THE  
3649 CALLER TO EXAMINE THE ERROR STATUS  
3650 OF THE ROUTINE CALL.  
3651  
3652 IMPLICIT INPUTS:  
3653 WT_DATA  
3654 GETS LOADED WITH THE GENERATED  
3655 WRT_MASK DATA PATTERN THUS ALLOWING  
3656 CALLER TO PRINT FAILING GOOD DATA.  
3657  
3658 RD_DATA  
3659 GETS LOADED WITH DATA READ FROM THE  
3660 REGISTER THUS ALLOWING CALLER  
3661 TO PRINT FAILING BAD DATA.  
3662  
3663 IMPLICIT OUTPUTS:  
3664 GLOBAL LOCATION WR_DATA  
3665 AND RD_DATA LOADED WITH GOOD  
3666 AND BAD REGISTER DATA  
3667  
3668 !--
```

22-Oct-1980 10:47:44
22-Oct-1980 10:45:32

TOPS-20 Bliss-16 V2(206)
PA:<NEALE>BL2ML4.BLI.2 (41)

22-Oct-1980 10:47:44 TOPS-20 Bliss-16 V2(206)
22-Oct-1980 10:45:32 PA:<NEALE>BL2ML4.BLI.2 (41)

```

5037 :ML4
5038 :
5039 :
5040 : 3669
5041 : 3670
5042 : 3671
5043 : 3672
5044 : 3673
5045 : 3674
5046 : 3675
5047 : 3676
5048 : 3677
5052 :
5056 017514 004167 164266 RD.DS: JSR R1,$SAVE2 ; 3617
5057 017520 005076 000010 CLR @10(SP) ; ERR.FLG 3670
5058 017524 016600 000012 MOV 12(SP),R0 ; INDEX,* 3671
5059 017530 006300 ASL R0
5060 017532 006300 ASL R0
5061 017534 006300 ASL R0
5062 017536 010001 MOV R0,R1
5063 017540 016100 011702 MOV ML.REG+2(R1),R0
5064 017544 056600 000014 BIS 14(SP),R0 ; TST.PAT,*
5065 017550 046100 011704 BIC ML.REG+4(R1),R0
5066 017554 016102 011706 MOV ML.REG+6(R1),R2
5067 017560 050002 BIS R0,R2
5068 017562 010267 172102 MOV R2,WT.DATA
5069 017566 017702 172156 MOV @ML.REG+50,R2 ; 3672
5070 017572 056102 011706 BIS ML.REG+6(R1),R2
5071 017576 010267 172070 MOV R2,RD.DATA
5072 017602 026767 172062 172062 CMP WT.DATA,RD.DATA ; 3674
5073 017610 001403 BEQ 1$
5074 017612 012776 000001 000010 MOV #1,@10(SP) ; *,ERR.FLG
5075 017620 000207 1$: RTS PC ; 3617
5076 :
5077 : ; Routine Size: 35 words
5078 : ; Maximum stack depth per invocation: 3 words
5083 :
5084 :

```

5086 :ML4
 5087 :
 5088 :
 5089 :
 5090 :
 5091 :
 5092 :
 5093 :
 5094 :
 5095 :
 5096 :
 5097 :
 5098 :
 5099 :
 5100 :
 5101 :
 5102 :
 5103 :
 5104 :
 5105 :
 5106 :
 5107 :
 5108 :
 5109 :
 5110 :
 5111 :
 5112 :
 5113 :
 5114 :
 5115 :
 5116 :
 5117 :
 5118 :
 5119 :
 5120 :
 5121 :
 5122 :
 5123 :
 5124 :
 5125 :
 5126 :
 5127 :
 5128 :
 5129 :
 5130 :
 5131 :
 5132 :
 5133 :
 5134 :
 5135 :
 5136 :
 5137 :
 5138 :
 5139 :
 5140 :

22-Oct-1980 10:47:44
 22-Oct-1980 10:45:32

TOPS-20 Bliss-16 V2(206)
 PA:<NEALE>BL2ML4.BLI.2 (42)

3678 routine WRT_REG (TST_PAT, REG_SEL, index) : novalue =
 3679 begin

```

  3681 !++
  3682 !FUNCTIONAL DESCRIPTION:
  3683 !   A ROUTINE TO SELECTIVELY
  3684 !   CALLED ROUTINES WHICH
  3685 !   WRITE TO INDIVIDUAL ML11
  3686 !   REGISTERS
  3687
  3688 !FORMAL PARAMETERS:
  3689 !   REG_SEL
  3690 !   CASE SELECT EXPRESSION TO
  3691 !   SELECT THE WRITE REGISTER
  3692 !   ROUTINE TO CALLED
  3693
  3694 !   TST_PAT
  3695 !   DATA PATTERN WHICH THE SELECTED
  3696 !   REGISTER WILL BE TESTED AGAINST
  3697
  3698 !   INDEX
  3699 !   LOADED WITH THE ML_REG_INDEX
  3700 !   SELECT NUMBER OF THE REGISTER
  3701 !   BEING TESTED
  3702
  3703 !SIDE EFFECTS:
  3704 !   WHEN A WRITE REGISTER ROUTINE IS CALLED
  3705 !   THE VARIABLE 'INDEX' FROM THE CALLING
  3706 !   TEST IS LOADED WITH THE REGISTERS
  3707 !   ML_REG_INDEX NUMBER.
  3708
  3709 !   THIS ENABLES THE CALLING TEST TO FIND
  3710 !   THE FAILING REGISTER ADDRESS.
  3711
  3712 !--
  
```

```

  3713
  3714 case .REG_SEL from 0 to 13 of
  3715 set
  3716
  3717 [0] :
  3718 WRT_CS1 (.TST_PAT, .index = 0);
  3719 !SELECT THE WRITE REGISTER ROUTINE CALL
  3720 !CALL ROUTINE TO LOAD MLCS1
  3721
  3722 [1] :
  3723 WRT_ER (.TST_PAT, .index = 6);
  3724 !CALL ROUTINE TO LOAD MLER
  3725
  3726 [2] :
  3727 WRT_DA (.TST_PAT, .index = 3);
  3728 !CALL ROUTINE TO LOAD MLDA
  3729
  3730 [3] :
  3731 WRT_MR (.TST_PAT, .index = 10);
  3732 !CALL ROUTINE TO LOAD MLMR
  3733
  3734 [4] :
  
```

```

5142 ;ML4
5143 ;
5144
5145 : 3730 WRT_E1 (.TST_PAT, .index = 13); !CALL ROUTINE TO LOAD MLE1
5146 : 3731
5147 : 3732 [5] : WRT_E2 (.TST_PAT, .index = 14); !CALL ROUTINE TO LOAD MLE2
5148 : 3733
5149 : 3734
5150 : 3735 [6] : WRT_PA (.TST_PAT, .index = 8); !CALL ROUTINE TO LOAD MLPA
5151 : 3736
5152 : 3737
5153 : 3738 [7] : WRT_D1 (.TST_PAT, .index = 15); !CALL ROUTINE TO LOAD MLD1
5154 : 3739
5155 : 3740 [8] : WRT_D2 (.TST_PAT, .index = 16); !CALL ROUTINE TO LOAD MLD2
5156 : 3741
5157 : 3742 [9] : WRT_D3 (.TST_PAT, .index = 14); !CALL ROUTINE TO LOAD MLE2
5158 : 3743
5159 : 3744
5160 : 3745 [10] : WRT_PD (.TST_PAT, .index = 19); !CALL ROUTINE TO LOAD MLPD
5161 : 3746
5162 : 3747
5163 : 3748 [11] : WRT_FE (.TST_PAT, .index = 17); !CALL ROUTINE TO LOAD MLEE
5164 : 3749
5165 : 3750 [12] : WRT_EL (.TST_PAT, .index = 18); !CALL ROUTINE TO LOAD MLEL
5166 : 3751
5167 : 3752 [13] : WRT_DS (.TST_PAT, .index = 5); !CALL ROUTINE TO LOAD MLDS
5168 : 3753
5169 : 3754
5170 : 3755
5171 : 3756
5172 : 3757
5173 : 3758
5174 : 3759
5175 : 3760 end;
5179

```

```

5183 017622 004167 164160 WRT.REG:JSR R1,$SAVE2 ; 3678
5184 017626 016600 000010 MOV 10(SP),R0 ; INDEX,* 3718
5185 017632 016601 000014 MOV 14(SP),R1 ; TST.PAT,*
5186 017636 016602 000012 MOV 12(SP),R2 ; REG.SEL,* 3714
5187 017642 006302 ASL R2
5188 017644 066207 017650 ADD 1$(R2),PC
5189 017650 000034 1$: .WORD 2$-1$
5190 017652 000050 .WORD 3$-1$
5191 017654 000066 .WORD 4$-1$
5192 017656 000104 .WORD 5$-1$
5193 017660 000122 .WORD 6$-1$
5194 017662 000140 .WORD 7$-1$
5195 017664 000156 .WORD 8$-1$

```

Address	OpCode	Operand 1	Operand 2	Label	Instruction	Comment	Seq
5197							
5198							
5199							
5200	017666	000174			.WORD	9\$-1\$	
5201	017670	000212			.WORD	10\$-1\$	
5202	017672	000230			.WORD	11\$-1\$	
5203	017674	000246			.WORD	12\$-1\$	
5204	017676	000264			.WORD	13\$-1\$	
5205	017700	000302			.WORD	14\$-1\$	
5206	017702	000320			.WORD	15\$-1\$	
5207	017704	010146		2\$:	MOV	R1,-(SP)	3718
5208	017706	005010			CLR	(R0)	
5209	017710	005046			CLR	-(SP)	
5210	017712	004767	174256		JSR	PC,WRT.CS1	
5211	017716	000532			BR	16\$	3714
5212	017720	010146		3\$:	MOV	R1,-(SP)	3721
5213	017722	012710	000006		MOV	#6,(R0)	
5214	017726	011046			MOV	(R0),-(SP)	
5215	017730	004767	174416		JSR	PC,WRT.ER	
5216	017734	000523			BR	16\$	3714
5217	017736	010146		4\$:	MOV	R1,-(SP)	3724
5218	017740	012710	000003		MOV	#3,(R0)	
5219	017744	011046			MOV	(R0),-(SP)	
5220	017746	004767	174556		JSR	PC,WRT.DA	
5221	017752	000514			BR	16\$	3714
5222	017754	010146		5\$:	MOV	R1,-(SP)	3727
5223	017756	012710	000012		MOV	#12,(R0)	
5224	017762	011046			MOV	(R0),-(SP)	
5225	017764	004767	174716		JSR	PC,WRT.MR	
5226	017770	000505			BR	16\$	3714
5227	017772	010146		6\$:	MOV	R1,-(SP)	3730
5228	017774	012710	000015		MOV	#15,(R0)	
5229	020000	011046			MOV	(R0),-(SP)	
5230	020002	004767	175264		JSR	PC,WRT.E1	
5231	020006	000476			BR	16\$	3714
5232	020010	010146		7\$:	MOV	R1,-(SP)	3733
5233	020012	012710	000016		MOV	#16,(R0)	
5234	020016	011046			MOV	(R0),-(SP)	
5235	020020	004767	175454		JSR	PC,WRT.E2	
5236	020024	000467			BR	16\$	3714
5237	020026	010146		8\$:	MOV	R1,-(SP)	3736
5238	020030	012710	000010		MOV	#10,(R0)	
5239	020034	011046			MOV	(R0),-(SP)	
5240	020036	004767	175022		JSR	PC,WRT.PA	
5241	020042	000460			BR	16\$	3714
5242	020044	010146		9\$:	MOV	R1,-(SP)	3739
5243	020046	012710	000017		MOV	#17,(R0)	
5244	020052	011046			MOV	(R0),-(SP)	
5245	020054	004767	176222		JSR	PC,WRT.D1	
5246	020060	000451			BR	16\$	3714
5247	020062	010146		10\$:	MOV	R1,-(SP)	3742
5248	020064	012710	000020		MOV	#20,(R0)	
5249	020070	011046			MOV	(R0),-(SP)	
5250	020072	004767	176522		JSR	PC,WRT.D2	
5251	020076	000442			BR	16\$	3714

22-Oct-1980 10:47:44 TOPS
 22-Oct-1980 10:45:32 PA:<

5253				:ML4				
5254				:				
5255								
5256	020100	010146		11\$:	MOV	R1,-(SP)	:	
5257	020102	012710	000016		MOV	#16,(R0)	:	
5258	020106	011046			MOV	(R0),-(SP)	:	
5259	020110	004767	177022		JSR	PC,WRT.DS	:	
5260	020114	000433			BR	16\$:	3745
5261	020116	010146		12\$:	MOV	R1,-(SP)	:	
5262	020120	012710	000023		MOV	#23,(R0)	:	3714
5263	020124	011046			MOV	(R0),-(SP)	:	3748
5264	020126	004767	175600		JSR	PC,WRT.PD	:	
5265	020132	000424			BR	16\$:	3714
5266	020134	010146		13\$:	MOV	R1,-(SP)	:	3751
5267	020136	012710	000021		MOV	#21,(R0)	:	
5268	020142	011046			MOV	(R0),-(SP)	:	
5269	020144	004767	176070		JSR	PC,WRT.EE	:	
5270	020150	000415			BR	16\$:	3714
5271	020152	010146		14\$:	MOV	R1,-(SP)	:	3754
5272	020154	012710	000022		MOV	#22,(R0)	:	
5273	020160	011046			MOV	(R0),-(SP)	:	
5274	020162	004767	176010		JSR	PC,WRT.EL	:	
5275	020166	000406			BR	16\$:	3714
5276	020170	010146		15\$:	MOV	R1,-(SP)	:	3757
5277	020172	012710	000005		MOV	#5,(R0)	:	
5278	020176	011046			MOV	(R0),-(SP)	:	
5279	020200	004767	177306		JSR	PC,WRT.DS	:	
5280	020204	022626		16\$:	CMP	(SP)+,(SP)+	:	3679
5281	020206	000207			RTS	PC	:	3678

; Routine Size: 123 words
 ; Maximum stack depth per invocation: 5 words

5282
 5283
 5284
 5289
 5290

CZ
 HA

22-Oct-1980 10:47:44 TOPS-20 Bliss-16 V2(206)
 22-Oct-1980 10:45:32 PA:<NEALE>BL2ML4.BLI.2 (43)

5292 :ML4
 5293 :
 5294 :
 5295 :
 5296 :
 5297 :
 5298 :
 5299 :
 5300 :
 5301 :
 5302 :
 5303 :
 5304 :
 5305 :
 5306 :
 5307 :
 5308 :
 5309 :
 5310 :
 5311 :
 5312 :
 5313 :
 5314 :
 5315 :
 5316 :
 5317 :
 5318 :
 5319 :
 5320 :
 5321 :
 5322 :
 5323 :
 5324 :
 5325 :
 5326 :
 5327 :
 5328 :
 5329 :
 5330 :
 5331 :
 5332 :
 5333 :
 5334 :
 5335 :
 5336 :
 5337 :
 5338 :
 5339 :
 5340 :
 5341 :
 5342 :
 5343 :
 5344 :
 5345 :
 5346 :

```

3761 routine RD_REG (TST_PAT, REG_SEL, ERR_FLG) : novalue =
3762   begin
3763
3764   !++
3765   FUNCTIONAL DESCRIPTION:
3766   A ROUTINE TO SELECTIVELY
3767   CALLED ROUTINES WHICH
3768   READ TO INDIVIDUAL ML11
3769   REGISTERS.
3770
3771   FORMAL PARAMETERS:
3772   ERR_FLG
3773   CONTAINS THE ADDRESS (PASSED BY REF)
3774   OF THE CALLERS ERROR FLG TO ENABLE THE
3775   CALLER TO EXAMINE THE ERROR STATUS
3776   OF THE ROUTINE CALL.
3777
3778   REG_SEL
3779   CASE SELECT EXPRESSION TO
3780   SELECT THE WRITE REGISTER
3781   ROUTINE TO CALLED
3782
3783   TST_PAT
3784   DATA PATTERN WHICH THE SELECTED
3785   REGISTER WILL BE TESTED AGAINST
3786
3787   !--
3788
3789   case .REG_SEL from 0 to 13 of
3790     set
3791
3792     [0] :
3793         RD_CS1 (.TST_PAT, 0, .ERR_FLG);      !CALL ROOUTINE TO READ MLCS1
3794
3795     [1] :
3796         RD_ER (.TST_PAT, 6, .ERR_FLG);      !CALL ROUTINE TO READ MLER
3797
3798     [2] :
3799         RD_DA (.TST_PAT, 3, .ERR_FLG);      !CALL ROUTINE TO READ MLDA
3800
3801     [3] :
3802         RD_MR (.TST_PAT, 10, .ERR_FLG);     !CALL ROUTINE TO READ MLMR
3803
3804     [4] :
3805         RD_E1 (.TST_PAT, 13, .ERR_FLG);     !CALL ROUTINE TO READ MLE1
3806
3807     [5] :
3808         RD_E2 (.TST_PAT, 14, .ERR_FLG);     !CALL ROUTINE TO READ MLE2
3809
3810     [6] :
3811         RD_PA (.TST_PAT, 8, .ERR_FLG);      !CALL ROUTINE TO READ MLPA
3812
  
```


22-Oct-1980 10:47:44 TOPS-20 BLISS-16 V2(206)
22-Oct-1980 10:45:32 PA:<NEALE>BL2ML4.BLI.2 (43)

5348 :ML4
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 :
5377 :

3813
3814
3815
3816
3817
3818
3819
3820
3821
3822
3823
3824
3825
3826
3827
3828
3829
3830
3831
3832
3833
3834
3835

[7] : RD_D1 (.TST_PAT, 15, .ERR_FLG); !CALL ROUTINE TO READ MLD1
[8] : RD_D2 (.TST_PAT, 16, .ERR_FLG); !CALL ROUTINE TO READ MLD2
[9] : RD_D3 (.TST_PAT, 14, .ERR_FLG); !CALL ROUTINE TO READ MLD2
[10] : RD_PD (.TST_PAT, 19, .ERR_FLG); !CALL ROUTINE TO READ MLPD
[11] : RD_EE (.TST_PAT, 17, .ERR_FLG); !CALL ROUTINE TO READ MLEE
[12] : RD_EL (.TST_PAT, 18, .ERR_FLG); !CALL ROUTINE TO READ MLEL
[13] : RD_DS (.TST_PAT, 5, .ERR_FLG); !CALL ROUTINE TO READ MLDS
tes;

end;

5381 020210 004167 163572
5382 020214 016600 000010
5383 020220 016601 000014
5384 020224 016602 000012
5385 020230 006302
5386 020232 066207 020236
5387 020236 000034
5388 020240 000050
5389 020242 000066
5390 020244 000104
5391 020246 000122
5392 020250 000140
5393 020252 000156
5394 020254 000174
5395 020256 000212
5396 020260 000230
5397 020262 000246
5398 020264 000264
5399 020266 000302
5400 020270 000320
5401 020272 010146

RD.REG: JSR R1,\$SAVE2
MOV 10(SP),R0
MOV 14(SP),R1
MOV 12(SP),R2
ASL R2
ADD 1\$(R2),PC
1\$: .WORD 2\$(R1)
.WORD 3\$(R1)
.WORD 4\$(R1)
.WORD 5\$(R1)
.WORD 6\$(R1)
.WORD 7\$(R1)
.WORD 8\$(R1)
.WORD 9\$(R1)
.WORD 10\$(R1)
.WORD 11\$(R1)
.WORD 12\$(R1)
.WORD 13\$(R1)
.WORD 14\$(R1)
.WORD 15\$(R1)
2\$: MOV R1,-(SP)

:
: ERR.FLG,*
: TST.PAT,*
: REG.SEL,*

3761
3793
3789

3793

```

5403       ;ML4
5404       ;
5405
5406 020274 005046      CLR      -(SP)
5407 020276 010046      MOV      R0,-(SP)
5408 020300 004767 173740 JSR      PC,RD.CS1
5409 020304 000532      BR       16$
5410 020306 010146      3$: MOV      R1,-(SP)
5411 020310 012746 000006   MOV      #6,-(SP)
5412 020314 010046      MOV      R0,-(SP)
5413 020316 004767 174100 JSR      PC,RD.ER
5414 020322 000523      BR       16$
5415 020324 010146      4$: MOV      R1,-(SP)
5416 020326 012746 000003   MOV      #3,-(SP)
5417 020332 010046      MOV      R0,-(SP)
5418 020334 004767 174240 JSR      PC,RD.DA
5419 020340 000514      BR       16$
5420 020342 010146      5$: MOV      R1,-(SP)
5421 020344 012746 000012   MOV      #12,-(SP)
5422 020350 010046      MOV      R0,-(SP)
5423 020352 004767 174400 JSR      PC,RD.MR
5424 020356 000505      BR       16$
5425 020360 010146      6$: MOV      R1,-(SP)
5426 020362 012746 000015   MOV      #15,-(SP)
5427 020366 010046      MOV      R0,-(SP)
5428 020370 004767 174762 JSR      PC,RD.E1
5429 020374 000476      BR       16$
5430 020376 010146      7$: MOV      R1,-(SP)
5431 020400 012746 000016   MOV      #16,-(SP)
5432 020404 010046      MOV      R0,-(SP)
5433 020406 004767 175164 JSR      PC,RD.E2
5434 020412 000467      BR       16$
5435 020414 010146      8$: MOV      R1,-(SP)
5436 020416 012746 000010   MOV      #10,-(SP)
5437 020422 010046      MOV      R0,-(SP)
5438 020424 004767 174520 JSR      PC,RD.PA
5439 020430 000460      BR       16$
5440 020432 010146      9$: MOV      R1,-(SP)
5441 020434 012746 000017   MOV      #17,-(SP)
5442 020440 010046      MOV      R0,-(SP)
5443 020442 004767 175720 JSR      PC,RD.D1
5444 020446 000451      BR       16$
5445 020450 010146      10$: MOV     R1,-(SP)
5446 020452 012746 000020   MOV     #20,-(SP)
5447 020456 010046      MOV     R0,-(SP)
5448 020460 004767 176220 JSR     PC,RD.D2
5449 020464 000442      BR       16$
5450 020466 010146      11$: MOV     R1,-(SP)
5451 020470 012746 000016   MOV     #16,-(SP)
5452 020474 010046      MOV     R0,-(SP)
5453 020476 004767 176534 JSR     PC,RD.D3
5454 020502 000433      BR       16$
5455 020504 010146      12$: MOV     R1,-(SP)
5456 020506 012746 000023   MOV     #23,-(SP)
5457 020512 010046      MOV     R0,-(SP)

```

22-Oct-1980 10:47:44 TOPS
22-Oct-1980 10:45:32 PA:<

3789
3796
3789
3799
3789
3802
3789
3805
3789
3808
3789
3811
3789
3814
3789
3817
3789
3820
3789
3823

5459
5460
5461
5462 020514 004767 175320
5463 020520 000424
5464 020522 010146
5465 020524 012746 000021
5466 020530 010046
5467 020532 004767 175504
5468 020536 000415
5469 020540 010146
5470 020542 012746 000022
5471 020546 010046
5472 020550 004767 175424
5473 020554 000406
5474 020556 010146
5475 020560 012746 000005
5476 020564 010046
5477 020566 004767 176722
5478 020572 062706 000006
5479 020576 000207
5480
5481
5482
5487
5488
5489 ;

```
;ML4  
;  
JSR PC, RD.PD  
BR 16$  
13$: MOV R1, -(SP)  
MOV #21, -(SP)  
MOV R0, -(SP)  
JSR PC, RD.EE  
BR 16$  
14$: MOV R1, -(SP)  
MOV #22, -(SP)  
MOV R0, -(SP)  
JSR PC, RD.EL  
BR 16$  
15$: MOV R1, -(SP)  
MOV #5, -(SP)  
MOV R0, -(SP)  
JSR PC, RD.DS  
16$: ADD #6, SP  
RTS PC
```

```
22-Oct-1980 10:47:44 TOPS  
22-Oct-1980 10:45:32 PA:<  
  
3789  
3826  
  
3789  
3829  
  
3789  
3832  
  
3762  
3761
```

; Routine Size: 124 words
; Maximum stack depth per invocation: 6 words

3836 !<BLF/PAGE>

22-Oct-1980 10:47:44 TOPS-20 Bliss-16 V2(206)
 22-Oct-1980 10:45:32 PA:<NEALE>BL2ML4.BLI.2 (44)

```

5491 ;ML4
5492 :
5493 :
5494 : 3837 !
5495 : 3838 BGNINIT;
5496 : 3839 !
5497 : 3840 !
5498 : 3841 !
5499 : 3842 !
5500 : 3843 !
5501 : 3844 !
5502 : 3845 !
5503 : 3846 !
5504 : 3847 local
5505 : 3848   OFFSET;
5506 : 3849
5507 : 3850 external
5508 : 3851   L$UNIT;
5509 : 3852
5510 : 3853 if not READEF (EF_CONTINUE)      !SKIP INIT CODE IF CONTINUE
5511 : 3854 then
5512 : 3855   begin
5513 : 3856     if READEF (EF_START)
5514 : 3857     then
5515 : 3858       begin
5516 : 3859         ML_LUN = -1;
5517 : 3860
5518 : 3861         do
5519 : 3862           begin
5520 : 3863             ML_LUN = .ML_LUN + 1;      !INCREMENT LOGICAL UNIT NUMBER
5521 : 3864
5522 : 3865             if .ML_LUN geq .L$UNIT then DOCLN; !START OVER IF ALL UNITS HAVE BEEN TESTED
5523 : 3866
5524 : 3867           end
5525 : 3868         until (GPHARD (.ML_LUN, PTBL_PTR)) neq 0; !REPEAT THE GPHARD UNTIL A 0 IS RETURNED
5526 : 3869
5527 : 3870         RH_ADD = .((.PTBL_PTR) + 0); !GET BASE RH ADDRESS FOR THIS UNIT
5528 : 3871         RH_TYP = .((.PTBL_PTR) + 2); !GET RH TYPE FOR THIS UNIT
5529 : 3872         RH_VEC = .((.PTBL_PTR) + 4); !GET RH VECTOR FOR THIS UNIT
5530 : 3873         OFFSET = 0; !INIT OFF SET COUNT
5531 : 3874
5532 : 3875         incr COUNT from 0 to 21 do !LOAD THE REGISTER ADDRESS FOR THIS UNIT INTO ML_REG
5533 : 3876           begin
5534 : 3877             ML_REG [.COUNT, REGISTER_ADD] = .RH_ADD + .OFFSET;
5535 : 3878             OFFSET = .OFFSET + 2;
5536 : 3879           end;
5537 : 3880
5538 : 3881         end
5539 : 3882       else
5540 : 3883         begin
5541 : 3884           !IS THIS A NEW PASS
5542 : 3885
5543 : 3886           if READEF (EF_NEW) then ML_LUN = -1; !IF NEW PASS START GPHARDS AT LUN 0
5544 : 3887
5545 : 3888         do
  
```

22-Oct-1980 10:47:44 TOPS-20 Bliss-16 V2(206)
 22-Oct-1980 10:45:32 PA:<NEALE>BL2ML4.BLI.2 (44)

```

5547 :ML4
5548 :
5549 :
5550 :      3889      begin
5551 :      3890      ML_LUN = .ML_LUN + 1;          !IF NOT GET NEXT LUN PTABLE
5552 :      3891
5553 :      3892      if .ML_LUN geq .L$UNIT then DOCLN; !START OVER IF ALL UNITS ARE TESTED
5554 :      3893
5555 :      3894      end
5556 :      3895      until (GPHARD (.ML_LUN, PTBL_PTR); neq 0;
5557 :      3896
5558 :      3897      end;
5559 :      3898
5560 :      3899      PAR_DIS = .((.PTBL_PTR) + 12);    !GET PARITY DISABLE FLAG
5561 :      3900      ML_DUT = .((.PTBL_PTR) + 10);    !GET DRIVE NUMBER
5562 :      3901      OP_NUM_ARR = .((.PTBL_PTR) + 6) - 1; !GET OPERATOR NUMBER OF ARRAYS
5563 :      3902      GOOD_BLK = ZEROES;              !INIT GOOD BLOCK TO BLOCK ZERO
5564 :      3903      ARR_16 = ZEROES;                !INIT ARRAY 16 TO ZERO
5565 :      3904      LST_ARR = ZEROES;              !INIT LAST ARRAY TO ZERO
5566 :      3905      LST_BLK = ZEROES;              !INIT LAST BLOCK TO ZERO
5567 :      3906
5568 :      3907      if .((.PTBL_PTR) + 8) IS_SET      !CALCULATE ML11 16K MOS RAM PARAMETERS
5569 :      3908      then
5570 :      3909      begin
5571 :      3910      DRIVE_TYPE = %o'000110';          !EXPECTED DRIVE TYPE VALUE
5572 :      3911      W_C_SIZE = %o'140000';         !WORD COUNT SIZE FOR 16K WORD XFER
5573 :      3912      RAS_INC = %o'200';             !RAS INCREMENT FOR 16K RAMS
5574 :      3913      CHIP_SIZ = 16;                 !CHIP SIZE
5575 :      3914      ARR_INC = %o'1000';           !ARRAY INCREMENT
5576 :      3915      ARR_16<9, 4> = %o'17';        !ARRAY 16
5577 :      3916      LST_ARR<9, 4> = .OP_NUM_ARR;  !LAST ARRAY
5578 :      3917      LST_BLK<9, 4> = .OP_NUM_ARR;  !LAST BLOCK
5579 :      3918      LST_BLK = .LST_BLK or %o'777';
5580 :      3919      end
5581 :      3920      else
5582 :      3921      begin
5583 :      3922      DRIVE_TYPE = %o'000111';          !EXPECTED DRIVE TYPE VALUE
5584 :      3923      W_C_SIZE = %o'000000';         !WORD COUNT SIZE FOR 64K WORD XFER
5585 :      3924      RAS_INC = %o'1000';             !RAS INCREMENT FOR 64K RAMS
5586 :      3925      CHIP_SIZ = 64;                 !CHIP SIZE
5587 :      3926      ARR_INC = %o'4000';           !ARRAY INCREMENT
5588 :      3927      ARR_16<11, 4> = %o'74';        !ARRAY 16
5589 :      3928      LST_ARR<11, 4> = .OP_NUM_ARR;  !LAST ARRAY
5590 :      3929      LST_BLK<11, 4> = .OP_NUM_ARR;  !LAST BLOCK
5591 :      3930      LST_BLK = .LST_BLK or %o'3777';
5592 :      3931      end;
5593 :      3932
5594 :      3933      PRINTB (FMT_17, .ML_LUN);      !TELL OPERATOR WHICH UNIT IS BEING TESTED
5595 :      3934      CLR_MBUS;                       !CLEAR MASS BUS
5596 :      3935      end;
5597 :      3936
5598 :      3937      ENDINIT;
  
```

5603					:ML4			
5604					:			
5605								
5606								
5607						.GLOBL	L\$UNIT	
5608								
5609								
5613	020600	004167	163216		LINIT:	JSR	R1,\$SAVE3	: 3835
5614	020604	012700	000036			MOV	#36,R0	: 3853
5615	020610	104447				TRAP	47	
5616	020612	103001				BHIS	1\$	
5617	020614	000207				RTS	PC	
5618	020616	012700	000040		1\$:	MOV	#40,R0	: 3857
5619	020622	104447				TRAP	47	
5620	020624	103061				BHIS	5\$	
5621	020626	012767	177777	171332		MOV	#-1,ML.LUN	: 3860
5622	020634	005267	171326		2\$:	INC	ML.LUN	: 3864
5623	020640	026767	171322	161144		CMP	ML.LUN,L\$UNIT	: 3866
5624	020646	002401				BLT	3\$	
5625	020650	104444				TRAP	44	
5626	020652	016700	171310		3\$:	MOV	ML.LUN,R0	: 3869
5627	020656	104442				TRAP	42	
5628	020660	010067	167434			MOV	R0,PTBL.PTR	
5629	020664	005767	167430			TST	PTBL.PTR	
5630	020670	001761				BEQ	2\$	
5631	020672	017767	167422	171260		MOV	@PTBL.PTR,RH.ADD	: 3871
5632	020700	016701	167414			MOV	PTBL.PTR,R1	: 3872
5633	020704	016167	000002	171250		MOV	2(R1),RH.TYP	
5634	020712	016701	167402			MOV	PTBL.PTR,R1	: 3873
5635	020716	016167	000004	171240		MOV	4(R1),RH.VEC	
5636	020724	005002				CLR	R2	: OFFSET 3874
5637	020726	005001				CLR	R1	: COUNT 3876
5638	020730	010100			4\$:	MOV	R1,R0	: COUNT,* 3878
5639	020732	006300				ASL	R0	
5640	020734	006300				ASL	R0	
5641	020736	006300				ASL	R0	
5642	020740	016703	171214			MOV	RH.ADD,R3	
5643	020744	060203				ADD	R2,R3	: OFFSET,*
5644	020746	010360	011700			MOV	R3,ML.REG(R0)	
5645	020752	062702	000002			ADD	#2,R2	: *,OFFSET 3879
5646	020756	005201				INC	R1	: COUNT 3876
5647	020760	020127	000025			CMP	R1,#25	: COUNT,*
5648	020764	003761				BLE	4\$	
5649	020766	000426				BR	8\$: 3857
5650	020770	012700	000035		5\$:	MOV	#35,R0	: 3886
5651	020774	104447				TRAP	47	
5652	020776	103003				BHIS	6\$	
5653	021000	012767	177777	171160		MOV	#-1,ML.LUN	
5654	021006	005267	171154		6\$:	INC	ML.LUN	: 3890
5655	021012	026767	171150	160772		CMP	ML.LUN,L\$UNIT	: 3892
5656	021020	002401				BLT	7\$	

Address	Instruction	Source	Target	Offset	Symbol	Line
5658		:ML4				
5659		:				
5660						
5661	021022	104444			TRAP	44
5662	021024	016700	171136		MOV	ML.LUN,R0
5663	021030	104442			TRAP	42
5664	021032	010067	167262		MOV	R0,PTBL.PTR
5665	021036	005767	167256		TST	PTBL.PTR
5666	021042	001761			BEQ	6\$
5667	021044	016701	167250		MOV	PTBL.PTR,R1
5668	021050	016167	000014	167252	MOV	14(R1),PAR.DIS
5669	021056	016701	167236		MOV	PTBL.PTR,R1
5670	021062	016167	000012	171100	MOV	12(R1),ML.DUT
5671	021070	016701	167224		MOV	PTBL.PTR,R1
5672	021074	016103	000006		MOV	6(R1),R3
5673	021100	005303			DEC	R3
5674	021102	010367	167214		MOV	R3,OP.NUM.ARR
5675	021106	005067	167214		CLR	GOOD.BLK
5676	021112	005067	167220		CLR	ARR.16
5677	021116	005067	167216		CLR	LST.ARR
5678	021122	005067	167206		CLR	LST.BLK
5679	021126	016701	167166		MOV	PTBL.PTR,R1
5680	021132	026127	000010	000001	CMP	10(R1),#1
5681	021140	001054			BNE	9\$
5682	021142	012767	000110	170524	MOV	#110,DRIVE.TYPE
5683	021150	012767	140000	170506	MOV	#-40000,W.C.SIZE
5684	021156	012767	000200	170502	MOV	#200,RAS.INC
5685	021164	012767	000020	167140	MOV	#20,CHIP.SIZ
5686	021172	012767	001000	167124	MOV	#1000,ARR.INC
5687	021200	052767	017000	167130	BIS	#17000,ARR.16
5688	021206	016703	167110		MOV	OP.NUM.ARR,R3
5689	021212	000303			SWAB	R3
5690	021214	006303			ASL	R3
5691	021216	042703	160777		BIC	#160777,R3
5692	021222	042767	017000	167110	BIC	#17000,LST.ARR
5693	021230	050367	167104		BIS	R3,LST.ARR
5694	021234	016703	167062		MOV	OP.NUM.ARR,R3
5695	021240	000303			SWAB	R3
5696	021242	006303			ASL	R3
5697	021244	042703	160777		BIC	#160777,R3
5698	021250	042767	017000	167056	BIC	#17000,LST.BLK
5699	021256	050367	167052		BIS	R3,LST.BLK
5700	021262	052767	000777	167044	BIS	#777,LST.BLK
5701	021270	000461			BR	10\$
5702	021272	012767	000111	170374	MOV	#111,DRIVE.TYPE
5703	021300	005067	170360		CLR	W.C.SIZE
5704	021304	012767	001000	170354	MOV	#1000,RAS.INC
5705	021312	012767	000100	167012	MOV	#100,CHIP.SIZ
5706	021320	012767	004000	166776	MOV	#4000,ARR.INC
5707	021326	042767	074000	167002	BIC	#74000,ARR.16
5708	021334	052767	060000	166774	BIS	#60000,ARR.16
5709	021342	016703	166754		MOV	OP.NUM.ARR,R3
5710	021346	000303			SWAB	R3
5711	021350	006303			ASL	R3
5712	021352	006303			ASL	R3

22-Oct-1980 10:47:44 TOPS
22-Oct-1980 10:45:32 PA:<

3895
3899
3900
3901
3902
3903
3904
3905
3907
3910
3911
3912
3913
3914
3915
3916
3917
3918
3907
3922
3923
3924
3925
3926
3927
3928

```

5714      ;ML4
5715      ;
5716
5717 021354 006303          ASL      R3
5718 021356 042703 103777  BIC      #103777,R3
5719 021362 042767 074000 166750  BIC      #74000,LST.ARR
5720 021370 050367 166744          BIS      R3,LST.ARR
5721 021374 016703 166722          MOV      OP.NUM.ARR,R3
5722 021400 000303          SWAB     R3
5723 021402 006303          ASL      R3
5724 021404 006303          ASL      R3
5725 021406 006303          ASL      R3
5726 021410 042703 103777  BIC      #103777,R3
5727 021414 042767 074000 166712  BIC      #74000,LST.BLK
5728 021422 050367 166706          BIS      R3,LST.BLK
5729 021426 052767 003777 166700  BIS      #3777,LST.BLK
5730 021434 016746 170526          MOV      ML.LUN,-(SP)
5731 021440 012746 005244          MOV      #FMT.17,-(SP)
5732 021444 012746 000002          MOV      #2,-(SP)
5733 021450 010600          MOV      SP,R0
5734 021452 104414          TRAP     14
5735 021454 152777 000040 170256  BISB     #40,@ML.REG+40
5736 021462 016703 170502          MOV      ML.DUT,R3
5737 021466 042703 177770          BIC      #177770,R3
5738 021472 142777 000007 170240  BICB     #7,@ML.REG+40
5739 021500 150377 170234          BISB     R3,@ML.REG+40
5740 021504 062706 000006          ADD      #6,SP
5741 021510 000207          RTS      PC
5742
5743      ; Routine Size: 229 words
5744      ; Maximum stack depth per invocation: 7 words
5749
5750
5754
5758 021512 004767 177062  L$INIT::JSR  PC,LINIT
5759 021516 104411          TRAP     11
5760 021520 000207          RTS      PC
5761
5762      ; Routine Size: 4 words
5763      ; Maximum stack depth per invocation: 0 words

```

22-Oct-1980 10:47:44 TOPS
22-Oct-1980 10:45:32 PA:<

3929

3930
3933

3855
3835

3935

22-Oct-1980 10:47:44 TOPS-20 Bliss-16 V2(206)
 22-Oct-1980 10:45:32 PA:<NEALE>BL2ML4.BLI.2 (44)

```

5769 :ML4
5770 :
5771 :
5772 : 3938 :
5773 : 3939 :
5774 : 3940 BGNTST;
5775 : 3941 :
5776 : 3942 :++
5777 : 3943 : TEST NUMBER: TST 1
5778 : 3944 :
5779 : 3945 : TEST NAME: DRIVE PRESENT TEST
5780 : 3946 :
5781 : 3947 : TEST DESCRIPTION:
5782 : 3948 : THIS TEST READS THE DESIRED SECTOR
5783 : 3949 : ADDRESS REGISTER OF THE DRIVE UNDER
5784 : 3950 : TEST, DELAYS 100 US, THEN
5785 : 3951 : READS THE NED BIT OF MLCS2
5786 : 3952 :
5787 : 3953 : IF SET, AN ERROR MESSAGE IS
5788 : 3954 : PRINTED AND THE UNIT IS DROPPED
5789 : 3955 :
5790 : 3956 : IMPLICIT INPUTS: !NONE
5791 : 3957 :
5792 : 3958 :
5793 : 3959 :--
5794 : 3960 :
5795 : 3961 Local
5796 : 3962 DODU_FLG, !DROP UNIT FLAG
5797 : 3963 SAVE; !TEMP STORAGE LOCATION
5798 : 3964 :
5799 : 3965 BGNSUB;
5800 : 3966 CLR MBUS;
5801 : 3967 DODU_FLG = ZERO;
5802 : 3968 SAVE = .MLDA; !READ A DRIVE REGISTER
5803 : 3969 DELAY (ONE_US); !DELAY 1 US
5804 : 3970 :
5805 : 3971 if .NED IS_SET !TEST THE NED BIT
5806 : 3972 then
5807 : 3973 begin
5808 : 3974 ERRDF (1, ASYNC, 0); !IF SET THEN REPORT ERROR AND SET DODU_FLG
5809 : 3975 PRINTB (ONE_FMT, PHR_3);
5810 : 3976 DODU_FLG = ONE;
5811 : 3977 end;
5812 : 3978 :
5813 : 3979 ENDSUB;
5814 : 3980 :
5815 : 3981 if .DODU_FLG IS_SET !DROP THIS UNIT IF DODU IS_SET
5816 : 3982 then
5817 : 3983 begin
5818 : 3984 DODU (.ML_LUN);
5819 : 3985 DOCLN;
5820 : 3986 end;
5821 : 3987 :
5822 : 3988 ENDTST;
  
```

```

5830
5831          .GLOBL  L$DLY
5832
5833
5837 021522 004167 162312      $T1:  JSR    R1,$SAVE4          ;
5838 021526 005746              TST    -(SP)
5839 021530 104402              1$:   TRAP   2          ;
5840 021532 152777 000040 170200  BISB   #40,@ML.REG+40      ;
5841 021540 016702 170424      MOV    ML.DUT,R2
5842 021544 042702 177770      BIC    #177770,R2
5843 021550 142777 000007 170162  BICB   #7,@ML.REG+40
5844 021556 150277 170156      BISB   R2,@ML.REG+40
5845 021562 005003              CLR    R3          ; DODU.FLG
5846 021564 017704 170140      MOV    @ML.REG+30,R4    ; *,SAVE
5847 021570 012701 000001      MOV    #1,R1          ; *,$$TMP2
5848 021574 001410              2$:   BEQ    5$
5849 021576 016702 160314      MOV    L$DLY,R2       ; *,$$TMP1
5850 021602 001407              BEQ    4$
5851 021604 005016              3$:   CLR    (SP)        ; $$TMP
5852 021606 005302              DEC    R2            ; $$TMP1
5853 021610 001375              BNE    3$
5854 021612 005301              4$:   DEC    R1          ; $$TMP2
5855 021614 000767              BR     2$
5856 021616 032777 010000 170114  5$:   BIT    #10000,@ML.REG+40 ;
5857 021624 001420              BEQ    6$
5858 021626 104455              TRAP   55          ;
5859 021630 000001              .WORD  1
5860 021632 007444              .WORD  ASYNC
5861 021634 000000              .WORD  0
5862 021636 012746 006576      MOV    #PHR.3,-(SP)    ;
5863 021642 012746 005350      MOV    #ONE.FMT,-(SP)
5864 021646 012746 000002      MOV    #2,-(SP)
5865 021652 010600              MOV    SP,R0         ; SP,*
5866 021654 104414              TRAP   14
5867 021656 012703 000001      MOV    #1,R3         ; *,DODU.FLG
5868 021662 062706 000006      ADD    #6,SP
5869 021666 104467              6$:   TRAP   67          ;
5870 021670 006000              ROR    R0
5871 021672 103716              BLO    1$
5872 021674 005303              DEC    R3            ; DODU.FLG
5873 021676 001004              BNE    7$
5874 021700 016700 170262      MOV    ML.LUN,R0     ;
5875 021704 104451              TRAP   51
5876 021706 104444              TRAP   44
5877 021710 005726              7$:   TST    (SP)+
5878 021712 000207              RTS    PC

```

22-Oct-1980 10:47:44 TOPS
22-Oct-1980 10:45:32 PA:<

```
5880 ;ML4
5881 ;
5882
5883
5884 ; Routine Size: 61 words
5885 ; Maximum stack depth per invocation: 9 words
5890
5891
5895
5899 021714 T1::
5900 021714 004767 177602 1$: JSR PC,$T1 ;
5901 021720 104466 TRAP 66
5902 021722 006000 ROR R0
5903 021724 103773 BLO 1$
5904 021726 000207 RTS PC
5905
5906 ; Routine Size: 6 words
5907 ; Maximum stack depth per invocation: 0 words
5912
5913
5914 ; 3989 !<BLF/PAGE>
```

3986

5916 :ML4
5917 :
5918 :
5919 :
5920 :
5921 :
5922 :
5923 :
5924 :
5925 :
5926 :
5927 :
5928 :
5929 :
5930 :
5931 :
5932 :
5933 :
5934 :
5935 :
5936 :
5937 :
5938 :
5939 :
5940 :
5941 :
5942 :
5943 :
5944 :
5945 :
5946 :
5947 :
5948 :
5949 :
5950 :
5951 :
5952 :
5953 :
5954 :
5955 :
5956 :
5957 :
5958 :
5959 :
5960 :
5961 :
5962 :
5963 :
5964 :
5965 :
5966 :
5967 :
5968 :
5969 :
5970 :

```

3990 !
3991 BGNTST;
3992
3993 !++
3994 TEST NUMBER: TST 2
3995
3996 TEST NAME: DRIVE SELECTION TEST
3997
3998 TEST DESCRIPTION:
3999 THIS TEST TESTS FOR UNIQUE DRIVE SELECTION BY WRITING THE DRIVE
4000 UNDER TEST (DUT) DRIVE NUMBER INTO ITS DSA REG. THEN WRITING THE DRIVE
4001 NUMBERS OF OTHER DRIVES INTO THEIR RESPECTIVE DSA REGISTERS
4002 AND READING THE DUT DSA FOR ITS DRIVE NUMBER.
4003
4004 IMPLICIT INPUTS:
4005 ML_DUT
4006 LOADED DURING THE INITIALIZATION CODE AND CONTAINS THE DRIVE
4007 NUMBER OF THE DRIVE PRESENTLY BEING TESTED.
4008 --
4009
4010 Local
4011 DODU_FLG, !DROP UNIT FLAG
4012 SAVE; !TEMPORARY SAVE LOCATION
4013
4014 BGNSUB;
4015 CLR_MBUS;
4016 DODU_FLG = ZERO;
4017 MLDA = .ML_DUT; !LOAD THIS DRIVES DRIVE NO. INTO ITS DSA REG
4018 SAVE = .MLDA; !READ THE REGISTER BACK
4019
4020 if .SAVE neq .ML_DUT !SEE IF DSA HAS DRIVE NUMBER
4021 then
4022 begin
4023 ERRDF (2, INTER, 0); !ERROR AND EXIT TEST IF DSA NEQ DRIVE NUM
4024 PRINTB (SIX_FMT, PHR 4, WRD 12, FNC 3, WRD 37, WRD 13, REG_6);
4025 PRINTB (FMT_2, .ML_DUT, .SAVE, (.ML_DUT xor .SAVE));
4026 EXIT_TST;
4027 end;
4028
4029 incr DRV_SEL from 0 to 7 do !WRITE DRV NO OF OTHER DRIVES INTO THEIR RESPECTIVE DSA REG.
4030
4031 if .DRV_SEL neq .ML_DUT !SKIP IF .DRV_SEL EQL TO THE DRIVE UNDER TEST (DUT)
4032 then
4033 begin
4034 DRV_NUM = .DRV_SEL; !SELECT DRIVE TO BE WRITTEN TO
4035 MLDA = .DRV_SEL; !WRITE DRIVE SEL NO. INTO ITS DSA REG
4036 DELAY (ONE_US); !DELAY 1 US
4037 end;
4038
4039 DRV_NUM = .ML_DUT; !SELECT THE DUT
4040 SAVE = .MLDA; !READ ITS DSA REG
4041

```

22-Oct-1980 10:47:44
22-Oct-1980 10:45:32

TOPS-20 Bliss-16 V2(206)
PA:<NEALE>BL2ML4.BLI.2 (45)

22-Oct-1980 10:47:44 TOPS-20 Bliss-16 V2(206)
22-Oct-1980 10:45:32 PA:<NEALE>BL2ML4.BLI.2 (45)

```

5972 :ML4
5973 :
5974 :
5975 :      4042 if .SAVE neq .ML_DUT
5976 :      4043 then
5977 :      4044 begin
5978 :      4045 ERRDF (3, ASYNC, 0);
5979 :      4046 PRINTB (THR_FMT, FNC_3, WRD_37, WRD_14);
5980 :      4047 PRINTB (FMT_1, .ML_DUT, .SAVE);
5981 :      4048 DODU_FLG = ONE;
5982 :      4049 end;
5983 :      4050
5984 :      4051 ENDSUB;
5985 :      4052
5986 :      4053 if .DODU_FLG IS_SET
5987 :      4054 then
5988 :      4055 begin
5989 :      4056 DODU (.ML_LUN);
5990 :      4057 DOCLN;
5991 :      4058 end;
5992 :      4059
5993 :      4060 ENDTST;
5997 :

```

!SEE IF WRITTING TO OTHER DRIVES CHANGE ITS VALUE

!ERROR AND SET DODU_FLG IF CHANGED

!DROP UNIT IF DODU_FLG IS_SET

6001	021730	004167	162124	\$T2:	JSR	R1,\$SAVE5	:	3988
6002	021734	005746			TST	-(SP)	:	
6003	021736	104402		1\$:	TRAP	2	:	4012
6004	021740	152777	000040	167772	BISB	#40,@ML.REG+40	:	4014
6005	021746	016703	170216		MOV	ML.DUT,R3	:	
6006	021752	010302			MOV	R3,R2	:	
6007	021754	042702	177770		BIC	#177770,R2	:	
6008	021760	142777	000007	167752	BICB	#7,@ML.REG+40	:	
6009	021766	150277	167746		BISB	R2,@ML.REG+40	:	
6010	021772	005005			CLR	R5	:	DODU.FLG 4016
6011	021774	010377	167730		MOV	R3,@ML.REG+30	:	4017
6012	022000	017704	167724		MOV	@ML.REG+30,R4	:	*.SAVE 4018
6013	022004	020403			CMP	R4,R3	:	SAVE,* 4020
6014	022006	001451			BEQ	2\$:	
6015	022010	104455			TRAP	55	:	4023
6016	022012	000002			.WORD	2	:	
6017	022014	007622			.WORD	INTER	:	
6018	022016	000000			.WORD	0	:	
6019	022020	012746	007342		MOV	#REG.6,-(SP)	:	4024
6020	022024	012746	005770		MOV	#WRD.13,-(SP)	:	
6021	022030	012746	006232		MOV	#WRD.37,-(SP)	:	
6022	022034	012746	006774		MOV	#FNC.3,-(SP)	:	
6023	022040	012746	005760		MOV	#WRD.12,-(SP)	:	
6024	022044	012746	006630		MOV	#PHR.4,-(SP)	:	
6025	022050	012746	005432		MOV	#SIX.FMT,-(SP)	:	

Address	Op Code	Op Hex	Op Dec	Comments	Labels	Sequence
6027				:ML4		
6028				:		
6029						
6030	022054	012746	000007	MOV #7,-(SP)		
6031	022060	010600		MOV SP,R0	; SP,*	
6032	022062	104414		TRAP 14		
6033	022064	016716	170100	MOV ML.DUT,(SP)		4025
6034	022070	010403		MOV R4,R3	; SAVE,*	
6035	022072	041603		BIC (SP),R3		
6036	022074	040416		BIC R4,(SP)	; SAVE,*	
6037	022076	050316		BIS R3,(SP)		
6038	022100	010446		MOV R4,-(SP)	; SAVE,*	
6039	022102	016746	170062	MOV ML.DUT,-(SP)		
6040	022106	012746	004224	MOV #FMT.2,-(SP)		
6041	022112	012746	000004	MOV #4,-(SP)		
6042	022116	010600		MOV SP,R0	; SP,*	
6043	022120	104414		TRAP 14		
6044	022122	104463		TRAP 63		
6045	022124	062706	000030	ADD #30,SP		4020
6046	022130	000522		BR 10\$		4022
6047	022132	016700	170032	MOV ML.DUT,R0		4031
6048	022136	005003		CLR R3	; DRV.SEL	4029
6049	022140	020300		CMP R3,R0	; DRV.SEL,*	4031
6050	022142	001425		BEQ 7\$		
6051	022144	010302		MOV R3,R2	; DRV.SEL,*	4034
6052	022146	042702	177770	BIC #177770,R2		
6053	022152	142777	000007	BICB #7,@ML.REG+40		
6054	022160	150277	167554	BISB R2,@ML.REG+40		
6055	022164	010377	167540	MOV R3,@ML.REG+30	; DRV.SEL,*	4035
6056	022170	012701	000001	MOV #1,R1	; *,\$\$TMP2	4036
6057	022174	001410		BEQ 7\$		
6058	022176	016702	157714	MOV L\$DLY,R2	; *,\$\$TMP1	
6059	022202	001403		BEQ 6\$		
6060	022204	005016		CLR (SP)	; \$\$TMP	
6061	022206	005302		DEC R2	; \$\$TMP1	
6062	022210	001375		BNE 5\$		
6063	022212	005301		DEC R1	; \$\$TMP2	
6064	022214	000767		BR 4\$		
6065	022216	005203		INC R3	; DRV.SEL	4029
6066	022220	020327	000007	CMP R3,#7	; DRV.SEL,*	
6067	022224	003745		BLE 3\$		
6068	022226	010003		MOV R0,R3		4039
6069	022230	042703	177770	BIC #177770,R3		
6070	022234	142777	000007	BICB #7,@ML.REG+40		
6071	022242	150377	167472	BISB R3,@ML.REG+40		
6072	022246	017704	167456	MOV @ML.REG+30,R4	; *,SAVE	4040
6073	022252	020400		CMP R4,R0	; SAVE,*	4042
6074	022254	001435		BEQ 8\$		
6075	022256	104455		TRAP 55		4045
6076	022260	000003		.WORD 3		
6077	022262	007444		.WORD ASYNC		
6078	022264	000000		.WORD 0		
6079	022266	012746	005774	MOV #WRD.14,-(SP)		4046
6080	022272	012746	006232	MOV #WRD.37,-(SP)		
6081	022276	012746	006774	MOV #FNC.3,-(SP)		

```
6083      :ML4
6084      :
6085
6086 022302 012746 005366      MOV      #THR.FMT,-(SP)
6087 022306 012746 000004      MOV      #4,-(SP)
6088 022312 010600              MOV      SP,R0      ; SP,*
6089 022314 104414              TRAP     14
6090 022316 010416              MCV      R4,(SP)    ; SAVE,*      4047
6091 022320 016746 167644      MOV      ML.DUT,-(SP)
6092 022324 012746 004164      MOV      #FMT.1,-(SP)
6093 022330 012746 000003      MOV      #3,-(SP)
6094 022334 010600              MOV      SP,R0      ; SP,*
6095 022336 104414              TRAP     14
6096 022340 012705 000001      MOV      #1,R5      ; *,DODU.FLG    4048
6097 022344 062706 000020      ADD      #20,SP     ;              4044
6098 022350 104467              TRAP     67         ;              4049
6099 022352 006000
6100 022354 103002
6101 022356 000167 177354      JMP      1$
6102 022362 005305      9$: DEC      R5      ; DODU.FLG      4053
6103 022364 001004              BNE      10$
6104 022366 016700 167574      MOV      ML.LUN,R0 ;              4056
6105 022372 104451              TRAP     51
6106 022374 104444              TRAP     44
6107 022376 005726      10$: TST      (SP)+    ;              3988
6108 022400 000207      RTS      PC
6109
6110      ; Routine Size: 149 words
6111      ; Maximum stack depth per invocation: 19 words
6116
6117
6121
6125 022402      T2::
6126 022402 004767 177322      1$: JSR      PC,$T2 ;              4058
6127 022406 104466              TRAP     66
6128 022410 006000              ROR      R0
6129 022412 103773              BLO      1$
6130 022414 000207      RTS      PC
6131
6132      ; Routine Size: 6 words
6133      ; Maximum stack depth per invocation: 0 words
6141
6142
6143 ;      4061 !<BLF/PAGE>
```

22-Oct-1980 10:47:44 TOPS-20 Bliss-16 V2(206)
22-Oct-1980 10:45:32 PA:<NEALE>BL2ML4.BLI.2 (46)

```

6145 :ML4
6146 :
6147 :
6148 : 4062 !
6149 : 4063 BGNTST;
6150 : 4064
6151 : 4065 !++
6152 : 4066 TEST NUMBER: TST 3
6153 : 4067
6154 : 4068 TEST NAME: READ WRITE REG ONES/ZEROES TEST
6155 : 4069
6156 : 4070 TEST DESCRIPTION:
6157 : 4071 THIS TEST WRITES AND READS A DATA PATTERN OF ALL ONES AND ZEROES TO ALL
6158 : 4072 OF THE ML11'S READ / WRITE REGISTERS.
6159 : 4073
6160 : 4074 ROUTINES WRT_REG AND RD_REG ACCEPT ARGUMENTS TO FURTHER SELECT ROUTINES
6161 : 4075 WHICH ACTUALLY PERFORMS THE READING AND WRITING OF THE REGISTERS.
6162 : 4076
6163 : 4077 THE UNIT IS DROPPED ON DETECTED ERRORS.
6164 : 4078
6165 : 4079 IMPLICIT INPUTS:
6166 : 4080 WT_DATA
6167 : 4081 LOADED BY READ REGISTER ROUTINES AND CONTAINS THE DATA PATTERN WRITTEN
6168 : 4082 TO THE REGISTERS (REPRESENTS GOOD DATA).
6169 : 4083
6170 : 4084 RD_DATA
6171 : 4085 LOADED BY THE READ REGISTER ROUTINES AND CONTAINS THE DATA PATTERN
6172 : 4086 READ FROM THE REGISTER (REPRESENTS BAD DATA).
6173 : 4087 !--
6174 : 4088
6175 : 4089 local
6176 : 4090 ERR_FLG, !ERROR FLAG PASSED TO ROUTINES
6177 : 4091 TST_PAT, !TEST PATTERN
6178 : 4092 index, !POINTS TO REGISTER PRESENTLY BEING TESTED
6179 : 4093 DODU_FLG; !DROP UNIT FLAG
6180 : 4094
6181 : 4095 DODU_FLG = ZERO;
6182 : 4096 TST_PAT = ONES; !LOAD TEST PAT WITH ONES
6183 : 4097
6184 : 4098 incr TWICE from 0 to 1 do !REPEAT LOOP TWICE
6185 : 4099 begin
6186 : 4100
6187 : 4101 incr REG_SEL from 0 to 10 do !TEST ELEVEN WRITE/READ REGISTERS
6188 : 4102 begin
6189 : 4103 BGNSUB;
6190 : 4104 CLR_MBUS;
6191 : 4105 WRT_REG (.TST_PAT, .REG_SEL, index); !WRITE TO THE REGISTER
6192 : 4106 RD_REG (.TST_PAT, .REG_SEL, ERR_FLG); !READ THE REGISTER
6193 : 4107
6194 : 4108 if .ERR_FLG IS_SET !SEE IF READ FOUND AN ERROR
6195 : 4109 then
6196 : 4110 begin !IF ERROR FLAG IS_SET THEN ERROR AND SET DODU_FLG
6197 : 4111
6198 : 4112 selectone .REG_SEL of !SELECT WHICH MODULE FAILED
6199 : 4113 set

```


22-Oct-1980 10:47:44 TOPS-20 Bliss-16 V2(206) ;
22-Oct-1980 10:45:32 PA:<NEALE>BL2ML4.BLI.2 (46)

```

6201 ;ML4
6202 ;
6203
6204 : 4114
6205 : 4115 [0 to 3] :
6206 : 4116 ERRDF (4, ASYNC, 0); !ASYNC MODULE FAILURE
6207 : 4117
6208 : 4118 [4 to 5] :
6209 : 4119 ERRDF (4, SYNC, 0); !SYNC MODULE FAILURE
6210 : 4120
6211 : 4121 [6 to 10] :
6212 : 4122 ERRDF (4, ARR_DAT, 0); !ARRAY DATA MODULE FAILURE
6213 : 4123 tes;
6214 : 4124
6215 : 4125 PRINTB (SIX_FMT, PHR 4, WRD 12, FNC 5, FNC 6, WRD 52, WRD 56);
6216 : 4126 PRINTB (FMT_16, .ML_REG [.index, REGISTER_ADD], .WT_DATA, .RD_DATA);
6217 : 4127 DODU_FLG = ONE;
6218 : 4128 end;
6219 : 4129
6220 : 4130 ENDSUB;
6221 : 4131 end;
6222 : 4132
6223 : 4133 TST_PAT = not .TST_PAT; !REPEAT AGAIN WITH COMPLIMENT DATA
6224 : 4134 end;
6225 : 4135
6226 : 4136 if .DODU_FLG IS_SET !DROP THIS UNIT IF THE DODU_FLG IS_SET
6227 : 4137 then
6228 : 4138 begin
6229 : 4139 DODU (.ML_LUN);
6230 : 4140 DOCLN;
6231 : 4141 end;
6232 : 4142
6233 : 4143 ENDTST;
6237

```

```

6241 022416 004167 161416 $T3: JSR R1,$SAVE4 ; 4060
6242 022422 024646 CMP -(SP),-(SP) ;
6243 022424 005004 CLR R4 ; DODU.FLG 4095
6244 022426 012702 177777 MOV #-1,R2 ; *,TST.PAT 4096
6245 022432 005001 CLR R1 ; TWICE 4098
6246 022434 005003 1$: CLR R3 ; REG.SEL 4101
6247 022436 104402 2$: TRAP 2 ; 4102
6248 022440 152777 000040 167272 BISB #40,@ML.REG+40 ; 4103
6249 022446 016700 167516 MOV ML.DUT,R0
6250 022452 042700 177770 BIC #177770,R0
6251 022456 142777 000007 167254 BICB #7,@ML.REG+40
6252 022464 150077 167250 BISB R0,@ML.REG+40
6253 022470 010246 MOV R2,-(SP) ; TST.PAT,* 4105
6254 022472 010346 MOV R3,-(SP) ; REG.SEL,*

```

6256							22-Oct-1980 10:47:44	TOPS
6257							22-Oct-1980 10:45:32	PA:<
6258								
6259	022474	012746	000010		MOV	#10,-(SP)		
6260	022500	060616			ADD	SP,(SP)		
6261	022502	004767	175114		JSR	PC,WRT.REG		; INDEX,*
6262	022506	010216			MOV	R2,(SP)		; TST.PAT,*
6263	022510	010346			MOV	R3,-(SP)		; REG.SEL,*
6264	022512	012746	000012		MOV	#12,-(SP)		
6265	022516	060616			ADD	SP,(SP)		; ERR.FLG,*
6266	022520	004767	175464		JSR	PC,RD.REG		
6267	022524	026627	000012	000001	CMP	12(SP),#1		; ERR.FLG,*
6268	022532	001106			BNE	6\$		
6269	022534	005703			TST	R3		; REG.SEL
6270	022536	002410			BLT	3\$		
6271	022540	020327	000003		CMP	R3,#3		; REG.SEL,*
6272	022544	003005			BGT	3\$		
6273	022546	104455			TRAP	55		;
6274	022550	000004			.WORD	4		
6275	022552	007444			.WORD	ASYN		
6276	022554	000000			.WORD	0		
6277	022556	000425			BR	5\$;
6278	022560	020327	000004		CMP	R3,#4		; REG.SEL,*
6279	022564	002410			BLT	4\$		
6280	022566	020327	000005		CMP	R3,#5		; REG.SEL,*
6281	022572	003005			BGT	4\$		
6282	022574	104455			TRAP	55		;
6283	022576	000004			.WORD	4		
6284	022600	007500			.WORD	SYN		
6285	022602	000000			.WORD	0		
6286	022604	000412			BR	5\$;
6287	022606	020327	000006		CMP	R3,#6		; REG.SEL,*
6288	022612	002407			BLT	5\$		
6289	022614	020327	000012		CMP	R3,#12		; REG.SEL,*
6290	022620	003004			BGT	5\$		
6291	022622	104455			TRAP	55		;
6292	022624	000004			.WORD	4		
6293	022626	007534			.WORD	ARR.DAT		
6294	022630	000000			.WORD	0		
6295	022632	012746	006454		MOV	#WRD.56,-(SP)		;
6296	022636	012746	006420		MOV	#WRD.52,-(SP)		
6297	022642	012746	007030		MOV	#FNC.6,-(SP)		
6298	022646	012746	007020		MOV	#FNC.5,-(SP)		
6299	022652	012746	005760		MOV	#WRD.12,-(SP)		
6300	022656	012746	006630		MOV	#PHR.4,-(SP)		
6301	022662	012746	005432		MOV	#SIX.FMT,-(SP)		
6302	022666	012746	000007		MOV	#7,-(SP)		
6303	022672	010600			MOV	SP,R0		; SP,*
6304	022674	104414			TRAP	14		
6305	022676	016716	166770		MOV	RD.DATA,(SP)		;
6306	022702	016746	166762		MOV	WT.DATA,-(SP)		
6307	022706	016600	000036		MOV	36(SP),R0		; INDEX,*
6308	022712	006300			ASL	R0		
6309	022714	006300			ASL	R0		
6310	022716	006300			ASL	R0		

22-Oct-1980 10:47:44 TOPS
 22-Oct-1980 10:45:32 PA:<

```

6312      ;ML4
6313      ;
6314
6315 022720 016046 011700      MOV      ML.REG(R0),-(SP)
6316 022724 012746 005154      MOV      #FMT.16,-(SP)
6317 022730 012746 000004      MOV      #4,-(SP)
6318 022734 010600              MOV      SP,R0          ; SP,*
6319 022736 104414              TRAP     14
6320 022740 012704 000001      MOV      #1,R4          ; *,DODU.FLG
6321 022744 062706 000030      ADD      #30,SP          ;
6322 022750 062706 000012      6$:     ADD      #12,SP   ;
6323 022754 104467              TRAP     67             ;
6324 022756 00600C              ROR      R0
6325 022760 103626              BLO     2$
6326 022762 005203              INC      R3              ; REG.SEL
6327 022764 020327 000012      CMP      R3,#12         ; REG.SEL,*
6328 022770 003622              BLE     2$
6329 022772 005102              COM      R2              ; TST.PAT
6330 022774 005201              INC      R1              ; TWICE
6331 022776 020127 000001      CMP      R1,#1          ; TWICE,*
6332 023002 003614              BLE     1$
6333 023004 005304              DEC      R4              ; DODU.FLG
6334 023006 001004              BNE     7$
6335 023010 016700 167152      MOV      ML.LUN,R0      ;
6336 023014 104451              TRAP     51
6337 023016 104444              TRAP     44
6338 023020 022626      7$:     CMP      (SP)+,(SP)+ ;
6339 023022 000207              RTS      PC              ;
6340
6341      ; Routine Size: 131 words
6342      ; Maximum stack depth per invocation: 24 words
6347
6348
6352
6356 023024      T3::
6357 023024 004767 177366      1$:     JSR      PC,$T3    ;
6358 023030 104466              TRAP     66
6359 023032 006000              ROR      R0
6360 023034 103773              BLO     1$
6361 023036 000207              RTS      PC
6362
6363      ; Routine Size: 6 words
6364      ; Maximum stack depth per invocation: 0 words
6372
6373
6374 ;      4144 !<BLF/PAGE>
  
```

22-Oct-1980 10:47:44
22-Oct-1980 10:45:32

TOPS-20 Bliss-16 V2(206)
PA:<NEALE>BL2ML4.BLI.2 (47)

6376 :ML4
6377 :
6378 :
6379 :
6380 :
6381 :
6382 :
6383 :
6384 :
6385 :
6386 :
6387 :
6388 :
6389 :
6390 :
6391 :
6392 :
6393 :
6394 :
6395 :
6396 :
6397 :
6398 :
6399 :
6400 :
6401 :
6402 :
6403 :
6404 :
6405 :
6406 :
6407 :
6408 :
6409 :
6410 :
6411 :
6412 :
6413 :
6414 :
6415 :
6416 :
6417 :
6418 :
6419 :
6420 :
6421 :
6422 :
6423 :
6424 :
6425 :
6426 :
6427 :
6428 :
6429 :
6430 :

```
4145 !  
4146 BGNTST;  
4147  
4148 !++  
4149 TEST NUMBER: TST 4  
4150  
4151 TEST NAME: READ WRITE REG SHIFTING ONES/ZEROES TEST  
4152  
4153 TEST DESCRIPTION:  
4154 THIS TEST WRITES AND READS A  
4155 SHIFTING ONE'S AND SHIFTING ZEROE'S  
4156 PATTERN TO ALL THE ML11'S  
4157 READ/WRITE REGISTERS  
4158  
4159 ROUTINES WRT REG AND RD REG  
4160 ACCEPT ARGUMENTS TO FURTHER  
4161 SELECT ROUTINES WHICH ACTUALLY  
4162 PERFORMS THE READING AND  
4163 WRITING OF THE REGISTERS.  
4164  
4165 THE DRIVE IS DROPPED ON DETECTED  
4166 ERRORS.  
4167  
4168 IMPLICIT INPUTS:  
4169 WT DATA  
4170 [LOADED BY READ REGISTER ROUTINES AND  
4171 CONTAINS THE DATA PATTERN WRITTEN TO THE  
4172 REGISTERS (REPRESENTS GOOD DATA).  
4173  
4174 RD DATA  
4175 [LOADED BY THE READ REGISTER ROUTINES AND  
4176 CONTAINS THE DATA PATTERN READ FROM THE  
4177 REGISTER (REPRESENTS BAD DATA).  
4178  
4179  
4180 !--  
4181  
4182 Local  
4183 ERR_FLG,  
4184 TST_PAT,  
4185 index,  
4186 DODU_FLG;  
4187  
4188 DODU_FLG = ZERO;  
4189 TST_PAT = ONE;  
4190  
4191 incr SHIFT from 0 to 15 do  
4192 begin  
4193  
4194 incr TWICE from 0 to 1 do  
4195 begin  
4196
```

```
!ERROR FLAG PASSED TO ROUTINE  
!TEST PATTERN  
!POINTS TO REG PRESENTLY BEING TESTED  
!DROP UNIT FLAG  
  
!LOAD TST_PAT WITH A 1 IN A FILED OF 0'S  
  
!DO SHIFT 16 TIMES  
  
!REPEAT LOOP TWICE
```

6432 :ML4

6433 :
6434 :
6435 : 4197
6436 : 4198
6437 : 4199
6438 : 4200
6439 : 4201
6440 : 4202
6441 : 4203
6442 : 4204
6443 : 4205
6444 : 4206
6445 : 4207
6446 : 4208
6447 : 4209
6448 : 4210
6449 : 4211
6450 : 4212
6451 : 4213
6452 : 4214
6453 : 4215
6454 : 4216
6455 : 4217
6456 : 4218
6457 : 4219
6458 : 4220
6459 : 4221
6460 : 4222
6461 : 4223
6462 : 4224
6463 : 4225
6464 : 4226
6465 : 4227
6466 : 4228
6467 : 4229
6468 : 4230
6469 : 4231
6470 : 4232
6471 : 4233
6472 : 4234
6473 : 4235
6474 : 4236
6475 : 4237
6476 : 4238
6477 : 4239
6478 : 4240
6479 : 4241
6480 : 4242

```

incr REG_SEL from 0 to 10 do
begin
  BGNSUB;
  CLR_MBUS;
  WRT_REG (.TST_PAT, .REG_SEL, index);      !WRITE TO THE REGISTER
  RD_REG (.TST_PAT, .REG_SEL, ERR_FLG);     !READ THE REGISTER
  if .ERR_FLG IS_SET                        !SEE IF THE READ FOUND AN ERROR
  then
  begin
    !IF THE ERROR FLAG IS_SET THEN ERROR
    selectone .REG_SEL of
    set
    [0 to 3] :
      ERRDF (5, ASYNC, 0);      !ASYNC MODULE FAILURE
    [4 to 5] :
      ERRDF (5, SYNC, 0);      !SYNC MODULE FAILURE
    [6 to 10] :
      ERRDF (5, ARR_DAT, 0);   !ARRAY DATA MODULE FAILURE
  tes;
  PRINTB (SIX_FMT, PHR 4, WRD 12, FNC 5, FNC 6, WRD 52, WRD 56);
  PRINTB (FMT_16, .ML_REG [.index, REGISTER_ADD], .WT_DATA, .RD_DATA);
  DODU_FLG = ONE;
end;
ENDSUB;
end;
TST_PAT = not .TST_PAT;      !REPEAT WITH A 0 IN A FIELD OF 1'S
end;
TST_PAT = .TST_PAT^ONE;     !SHIFT THE 1 IN THE FIELD OF 0'S
end;
if .DODU_FLG IS_SET
then
begin
  DODU (.ML_LUN);
  DOCLN;
end;
ENDTST;

```

6491	023040	004167	161014	\$T4:	JSR	R1,\$SAVE5	:	4143
6492	023044	024646			CMP	-(SP),-(SP)		
6493	023046	005005			CLR	R5	: DODU.FLG	4188
6494	023050	012703	000001		MOV	#1,R3	: *,TST.PAT	4189
6495	023054	005001			CLR	R1	: SHIFT	4191
6496	023056	005002		1\$:	CLR	R2	: TWICE	4194
6497	023060	005004		2\$:	CLR	R4	: REG.SEL	4197

6498	023062	104402			3\$:	TRAP	2	:		
6499	023064	152777	000040	166646		BISB	#40,@ML.REG+40	:		
6500	023072	016700	167072			MOV	ML.DUT,RO	:		
6501	023076	042700	177770			BIC	#177770,RO	:		
6502	023102	142777	000007	166630		BICB	#7,@ML.REG+40	:		
6503	023110	150077	166624			BISB	RO,@ML.REG+40	:		
6504	023114	010346				MOV	R3,-(SP)	:	TST.PAT,*	4201
6505	023116	010446				MOV	R4,-(SP)	:	REG.SEL,*	
6506	023120	012746	000010			MOV	#10,-(SP)	:		
6507	023124	060616				ADD	SP,(SP)	:	INDEX,*	
6508	023126	004767	174470			JSR	PC,WRT.REG	:		
6509	023132	010316				MOV	R3,(SP)	:	TST.PAT,*	4202
6510	023134	010446				MOV	R4,-(SP)	:	REG.SEL,*	
6511	023136	012746	000012			MOV	#12,-(SP)	:		
6512	023142	060616				ADD	SP,(SP)	:	ERR.FLG,*	
6513	023144	004767	175040			JSR	PC,RD.REG	:		
6514	023150	026627	000012	000001		CMP	12(SP),#1	:	ERR.FLG,*	4204
6515	023156	001106				BNE	7\$:		
6516	023160	005704				TST	R4	:	REG.SEL	4208
6517	023162	002410				BLT	4\$:		
6518	023164	020427	000003			CMP	R4,#3	:	REG.SEL,*	
6519	023170	003005				BGT	4\$:		
6520	023172	104455				TRAP	55	:		4212
6521	023174	000005				.WORD	5	:		
6522	023176	007444				.WORD	ASYN	:		
6523	023200	000000				.WORD	0	:		
6524	023202	000425				BR	6\$:		4208
6525	023204	020427	000004		4\$:	CMP	R4,#4	:	REG.SEL,*	
6526	023210	002410				BLT	5\$:		
6527	023212	020427	000005			CMP	R4,#5	:	REG.SEL,*	
6528	023216	003005				BGT	5\$:		
6529	023220	104455				TRAP	55	:		4215
6530	023222	000005				.WORD	5	:		
6531	023224	007500				.WORD	SYN	:		
6532	023226	000000				.WORD	0	:		
6533	023230	000412				BR	6\$:		4208
6534	023232	020427	000006		5\$:	CMP	R4,#6	:	REG.SEL,*	
6535	023236	002407				BLT	6\$:		
6536	023240	020427	000012			CMP	R4,#12	:	REG.SEL,*	
6537	023244	003004				BGT	6\$:		
6538	023246	104455				TRAP	55	:		4218
6539	023250	000005				.WORD	5	:		
6540	023252	007534				.WORD	ARR.DAT	:		
6541					:ML4					
6542					:					
6543					:					
6544	023254	000000				.WORD	0	:		
6545	023256	012746	006454		6\$:	MOV	#WRD.56,-(SP)	:		4221
6546	023262	012746	006420			MOV	#WRD.52,-(SP)	:		
6547	023266	012746	007030			MOV	#FNC.6,-(SP)	:		
6548	023272	012746	007020			MOV	#FNC.5,-(SP)	:		
6549	023276	012746	005760			MOV	#WRD.12,-(SP)	:		
6550	023302	012746	006630			MOV	#PHR.4,-(SP)	:		
6551	023306	012746	005432			MOV	#SIX.FMT,-(SP)	:		
6552	023312	012746	000007			MOV	#7,-(SP)	:		
6553	023316	010600				MOV	SP,RO	:	SP,*	
6554	023320	104414				TRAP	14	:		

6555	023322	016716	166344	MOV	RD.DATA,(SP)	:	4222
6556	023326	016746	166336	MOV	WT.DATA,-(SP)	:	
6557	023332	016600	000036	MOV	36(SP),R0	: INDEX,*	
6558	023336	006300		ASL	R0	:	
6559	023340	006300		ASL	R0	:	
6560	023342	006300		ASL	R0	:	
6561	023344	016046	011700	MOV	ML.REG(R0),-(SP)	:	
6562	023350	012746	005154	MOV	#FMT.16,-(SP)	:	
6563	023354	012746	000004	MOV	#4,-(SP)	:	
6564	023360	010600		MOV	SP,R0	: SP,*	
6565	023362	104414		TRAP	14	:	
6566	023364	012705	000001	MOV	#1,R5	: *,DODU.FLG	4223
6567	023370	062706	000030	ADD	#30,SP	:	4206
6568	023374	062706	000012	7\$: ADD	#12,SP	:	4198
6569	023400	104467		TRAP	67	:	4224
6570	023402	006000		ROR	R0	:	
6571	023404	103626		BLO	3\$:	
6572	023406	005204		INC	R4	: REG.SEL	4197
6573	023410	020427	000012	CMP	R4,#12	: REG.SEL,*	
6574	023414	003622		BLE	3\$:	
6575	023416	005103		COM	R3	: TST.PAT	4229
6576	023420	005202		INC	R2	: TWICE	4194
6577	023422	020227	000001	CMP	R2,#1	: TWICE,*	
6578	023426	003614		BLE	2\$:	
6579	023430	006303		ASL	R3	: TST.PAT	4232
6580	023432	005201		INC	R1	: SHIFT	4191
6581	023434	020127	000017	CMP	R1,#17	: SHIFT,*	
6582	023440	003606		BLE	1\$:	
6583	023442	005305		DEC	R5	: DODU.FLG	4235
6584	023444	001004		BNE	8\$:	
6585	023446	016700	166514	MOV	ML.LUN,R0	:	4238
6586	023452	104451		TRAP	51	:	
6587	023454	104444		TRAP	44	:	
6588	023456	022626		8\$: CMP	(SP)+,(SP)+	:	4143
6589	023460	000207		RTS	PC	:	

; Routine Size: 137 words
 ; Maximum stack depth per invocation: 25 words

6590							
6591							
6592							
6600							
6601							
6605							
6609	023462			T4::			
6610	023462	004767	177352	1\$: JSR	PC,\$T4	:	4240
6611	023466	104466		TRAP	66	:	
6612	023470	006000		ROR	R0	:	
6613	023472	103773		BLO	1\$:	
6614	023474	000207		RTS	PC	:	

22-Oct-1980 10:47:44
22-Oct-1980 10:45:32

TOPS-20 Bliss-16 V2(206)
PA:<NEALE>BL2ML4.BLI.2 (48)

6620 :ML4
6621 :
6622 :
6623 :
6624 :
6625 :
6626 :
6627 :
6628 :
6629 :
6630 :
6631 :
6632 :
6633 :
6634 :
6635 :
6636 :
6637 :
6638 :
6639 :
6640 :
6641 :
6642 :
6643 :
6644 :
6645 :
6646 :
6647 :
6648 :
6649 :
6650 :
6651 :
6652 :
6653 :
6654 :
6655 :
6656 :
6657 :
6658 :
6659 :
6660 :
6661 :
6662 :
6663 :
6664 :
6665 :
6666 :
6667 :
6668 :
6669 :
6670 :
6671 :
6672 :
6673 :
6674 :

4245 !
4246 BGNTST;
4247
4248 !++
4249 TEST NUMBER: TST 5
4250
4251 TEST NAME: REGISTER INITIALIZATION TEST
4252
4253 TEST DESCRIPTION:
4254 THIS TEST TESTS THE ABILITY OF
4255 ALL ACCESSIBLE ML11 REGISTERS
4256 TO CLEAR OUT REGISTER DATA OF
4257 ONE'S AND ZEROES PATTERN.
4258
4259 ROUTINE WRT_REG WRITES A
4260 PATTERN TO THE SELECTED REGISTER.
4261
4262 A MASS BUS CLEAR IS DONE.
4263
4264 THEN ROUTINE RD_REG READS THE
4265 SELECTED REGISTER FOR CLEARED DATA
4266 THE DRIVE IS DROPPED ON DETECTED ERRORS.
4267
4268 THIS TEST WILL ALSO READ THE
4269 DRIVE TYPE REGISTER FOR ITS
4270 INITIAL REGISTER VALUE.
4271
4272 IMPLICIT INPUTS:
4273 REG_INIT_FLG
4274 THIS GLOBAL FLAG TELLS THE ROUTINES
4275 WHICH READ THE DATA DIAGNOSTIC
4276 REGISTERS (RD_D1, RD_D2, RD_D3)
4277 TO DO A MBUS CLEAR BEFORE READING
4278 THE REGISTER DURING THE REG_INIT
4279 TEST.
4280
4281 THIS FLAG IS NEEDED DUE TO THE UNIQUE
4282 MANNER IN WHICH THESE REGISTERS
4283 MUST BE READ.
4284
4285 RD_DATA
4286 LOADED BY THE READ REGISTER ROUTINE
4287 AND CONTAINS THE DATA PATTERN READ FROM
4288 THE REGISTER (REPRESENTS BAD DATA).
4289
4290 DRIVE_TYPE
4291 LOADED DURING THE INITIALIZATION CODE AND
4292 STORES THE EXPECTED CONTENTS OF THE DRIVE
4293 TYPE REGISTER.
4294 --
4295
4296 local

6676 :ML4

22-Oct-1980 10:47:44 TOPS-20 Bliss-16 V2(206)
22-Oct-1980 10:45:32 PA:<NEALE>BL2ML4.BLI.2 (48)

```

6677 :
6678 :
6679 :      4297      TST_PAT,
6680 :      4298      ERR_FLG,
6681 :      4299      index,
6682 :      4300      CLR_DATA,
6683 :      4301      SAVE,
6684 :      4302      DODU_FLG;
6685 :      4303
6686 :      4304      ML_REG [19, FORCE_HI] = %o'177777';
6687 :      4305      REG_INIT_FLG = ZERO;
6688 :      4306      DODU_FLG = ZERO;
6689 :      4307      TST_PAT = ONES;
6690 :      4308
6691 :      4309      incr TWICE from 0 to 1 do
6692 :      4310      begin
6693 :      4311
6694 :      4312      incr REG_SEL from 0 to 13 do
6695 :      4313      begin
6696 :      4314      BGNSUB;
6697 :      4315      CLR_MBUS;
6698 :      4316      WRT_REG (.TST_PAT, .REG_SEL, index);
6699 :      4317      CLR_DATA = (.HI) or (.IGNORE);
6700 :      4318
6701 :      4319      if (.REG_SEL neq 7) and (.REG_SEL neq 8) and (.REG_SEL neq 9)
6702 :      4320      then
6703 :      4321      begin
6704 :      4322      CLR_MBUS;
6705 :      4323      end
6706 :      4324      else
6707 :      4325      REG_INIT_FLG = ONE;
6708 :      4326
6709 :      4327      RD_REG (.CLR_DATA, .REG_SEL, ERR_FLG);
6710 :      4328
6711 :      4329      if .ERR_FLG IS_SET
6712 :      4330      then
6713 :      4331      begin
6714 :      4332
6715 :      4333      selectone .REG_SEL of
6716 :      4334      set
6717 :      4335
6718 :      4336      [0, 1, 2, 3, 12, 13] :
6719 :      4337      ERRDF (6, ASYNC, 0);
6720 :      4338
6721 :      4339      [4, 5, 11] :
6722 :      4340      ERRDF (6, SYNC, 0);
6723 :      4341
6724 :      4342      [6 to 10] :
6725 :      4343      ERRDF (6, ARR_DAT, 0);
6726 :      4344      tes;
6727 :      4345
6728 :      4346      PRINTB (SIX_FMT, PHR 4, WRD 12, WRD 52, WRD 23, WRD 52, WRD 56);
6729 :      4347      PRINTB (FMT_16, .ML_REG [.index, REGISTER_ADD], .CLR_DATA, .RD_DATA);
6730 :      4348      DODU_FLG = ONE;

```

```

!TEST PATTERN
!ERROR FLAG PASSED TO ROUTINE
!POINTS TO REG PRESENTLY BEING TESTED
!STORES CALCULATED REGISTER CLEAR DATA
!TEMPORARY STORAGE LOCATION
!DROP UNIT FLAG

!CLR DATA FOR MLPD IS ONES

!BACKGROUND PATTERN

!REPEAT LOOP TWICE

!TEST THIRTEEN REGISTERS

!WRITE REGISTER WITH BACKGROUND
!CALCULATE THE CLEARED DATA PATTERN

!OK TO CLEAR THESE REG'S HERE

!LET READ REG ROUTINE CLEAR THE REMAINING REG'S

!READ THE REGISTER FOR THE CLEARED DATA PAT

!SEE IF READ FOUND AN ERROR

!IF ERROR FLAG IS_SET THEN ERROR AND SET DODU_FLG

!FIND WHICH MODULE FAILED

!ASYNC MODULE FAILURE

!SYNC MODULE FAILURE

!ARRAY DATA MODULE

```

22-Oct-1980 10:47:44 TOPS-20 Bliss-16 V2(206)
22-Oct-1980 10:45:32 PA:<NEALE>BL2ML4.BLI.2 (48)

```

6732 :ML4
6733 :
6734 :
6735 :      4349          end;
6736 :      4350
6737 :      4351          ENDSUB;
6738 :      4352          end;
6739 :      4353
6740 :      4354          TST_PAT = not .TST_PAT;
6741 :      4355          end;
6742 :
6743 :      4357          ML_REG [19, FORCE_HI] = ZEROES;
6744 :      4358
6745 :      4359
6746 :      4360          !REPEAT WITH COMPLIMENT BACKGROUND PAT
6747 :      4361          !RESTORE MLPD FORCED_HI
6748 :      4362
6749 :      4363
6750 :      4364          if .MLDT neq .DRIVE_TYPE
6751 :      4365          then
6752 :      4366          begin
6753 :      4367          ERRDF (113, ASYNC, 0);
6754 :      4368          PRINTB (TWO_FMT, REG_7, PHR_4);
6755 :      4369          PRINTB (FMT_2, .DRIVE_TYPE, .SAVE);
6756 :      4370          end;
6757 :      4371
6758 :      4372          if .DODU_FLG IS_SET
6759 :      4373          then
6760 :      4374          begin
6761 :      4375          DODU (.ML_LUN);
6762 :      4376          DOCLN;
6763 :      4377          end;
6764 :      4378
6765 :      4379          ENDTST;
6769 :
6773 023476 004167 160356      $T5:   JSR      R1,$SAVE5          ;          4242
6774 023502 162706 000006      SUB      #6,SP          ;
6775 023506 012767 177777 166416  MOV      #-1,ML.REG+232 ;          4304
6776 023514 005067 166156      CLR      REG.INIT.FLG   ;          4305
6777 023520 005005      CLR      R5          ; DODU.FLG   4306
6778 023522 012702 177777      MOV      #-1,R2        ; *.TST.PAT 4307
6779 023526 005001      CLR      R1          ; TWICE     4309
6780 023530 005004      CLR      R4          ; REG.SEL   4312
6781 023532 104402      TRAP     2          ;          4313
6782 023534 152777 000040 166176  BISB    #40,@ML.REG+40 ;          4314
6783 023542 016703 166422      MOV      ML.DUT,R3
6784 023546 042703 177770      BIC     #177770,R3
6785 023552 142777 000007 166160  BICB    #7,@ML.REG+40

```

```

6787      ;ML4
6788      ;
6789
6790 023560 150377 166154      BISB    R3,@ML.REG+40
6791 023564 010246      MOV     R2,-(SP)          ; TST.PAT,*
6792 023566 010446      MOV     R4,-(SP)          ; REG.SEL,*
6793 023570 012746 000012      MOV     #12,-(SP)
6794 023574 060616      ADD     SP,(SP)          ; INDEX,*
6795 023576 004767 174020      JSR    PC,WRT.REG
6796 023602 016600 000012      MOV     12(SP),R0        ; INDEX,*
6797 023606 006300
6798 023610 006300
6799 023612 006300
6800 023614 010003
6801 023616 016300 011702      MOV     R0,R3
6802 023622 056300 011706      MOV     ML.REG+2(R3),R0
6803 023626 010066 000006      BIS     ML.REG+6(R3),R0
6804 023632 020427 000007      MOV     R0,6(SP)        ; *,CLR.DAIA
6805 023636 001423      CMP     R4,#7           ; REG.SEL,*
6806 023640 020427 000010      BEQ    3$
6807 023644 001420      CMP     R4,#10          ; REG.SEL,*
6808 023646 020427 000011      BEQ    3$
6809 023652 001415      CMP     R4,#11          ; REG.SEL,*
6810 023654 152777 000040 166056      BEQ    3$
6811 023662 016700 166302      BISB   #40,@ML.REG+40  ;
6812 023666 042700 177770      MOV     ML.DUT,R0
6813 023672 142777 000007 166040      BIC    #177770,R0
6814 023700 150077 166034      BICB   #7,@ML.REG+40
6815 023704 000403      BISB   R0,@ML.REG+40
6816 023706 012767 000001 165762 3$:    BR     4$
6817 023714 016616 000006 4$:    MOV     #1,REG.INIT.FLG
6818 023720 010446      MOV     6(SP),(SP)      ; CLR.DATA,*
6819 023722 012746 000014      MOV     R4,-(SP)        ; REG.SEL,*
6820 023726 060616      MOV     #14,-(SP)
6821 023730 004767 174254      ADD     SP,(SP)          ; ERR.FLG,*
6822 023734 026627 000014 000001      JSR    PC,RD.REG
6823 023742 001112      CMP     14(SP),#1      ; ERR.FLG,*
6824 023744 005704      BNE    12$
6825 023746 002403      TST    R4              ; REG.SEL
6826 023750 020427 000003      BLT    5$
6827 023754 003406      CMP     R4,#3           ; REG.SEL,*
6828 023756 020427 000014 5$:    BLE    6$
6829 023762 002410      CMP     R4,#14          ; REG.SEL,*
6830 023764 020427 000015      BLT    7$
6831 023770 003005      CMP     R4,#15          ; REG.SEL,*
6832 023772 104455 6$:    BGT    7$
6833 023774 000006      TRAP   55
6834 023776 007444      .WORD 6
6835 024000 000000      .WORD ASYNC
6836 024002 000430      .WORD 0
6837 024004 020427 000004 7$:    BR     11$
6838 024010 002403      CMP     R4,#4           ; REG.SEL,*
6839 024012 020427 000005 8$:    BLT    8$
6840 024016 003403      CMP     R4,#5           ; REG.SEL,*
6841 024020 020427 000013 8$:    BLE    9$
        CMP     R4,#13      ; REG.SEL,*
  
```

Address	OpCode	Operand 1	Operand 2	Label	Instruction	Comments	Address
6843				:ML4			22-Oct-1980 10:47:44 TOPS
6844				:			22-Oct-1980 10:45:32 PA:<
6845							
6846	024024	001005			BNE 10\$		
6847	024026	104455		9\$:	TRAP 55		4340
6848	024030	000006			.WORD 6		
6849	024032	007500			.WORD SYNC		
6850	024034	000000			.WORD 0		
6851	024036	000412			BR 11\$		4333
6852	024040	020427	000006	10\$:	CMP R4,#6	: REG.SEL,*	
6853	024044	002407			BLT 11\$		
6854	024046	020427	000012		CMP R4,#12	: REG.SEL,*	
6855	024052	003004			BGT 11\$		
6856	024054	104455			TRAP 55		4343
6857	024056	000006			.WORD 6		
6858	024060	007534			.WORD ARR.DAT		
6859	024062	000000			.WORD 0		
6860	024064	012746	006454	11\$:	MOV #WRD.56,-(SP)		4346
6861	024070	012746	006420		MOV #WRD.52,-(SP)		
6862	024074	012746	006076		MOV #WRD.23,-(SP)		
6863	024100	012746	006420		MOV #WRD.52,-(SP)		
6864	024104	012746	005760		MOV #WRD.12,-(SP)		
6865	024110	012746	006630		MOV #PHR.4,-(SP)		
6866	024114	012746	005432		MOV #SIX.FMT,-(SP)		
6867	024120	012746	000007		MOV #7,-(SP)		
6868	024124	010600			MOV SP,R0	: SP,*	
6869	024126	104414			TRAP 14		
6870	024130	016716	165536		MOV RD.DATA,(SP)		4347
6871	024134	016646	000032		MOV 32(SP),-(SP)	: CLR.DATA,*	
6872	024140	016346	011700		MOV ML.REG(R3),-(SP)		
6873	024144	012746	005154		MOV #FMT.16,-(SP)		
6874	024150	012746	000004		MOV #4,-(SP)		
6875	024154	010600			MOV SP,R0	: SP,*	
6876	024156	104414			TRAP 14		
6877	024160	012705	000001		MOV #1,R5	: *,DODU.FLG	4348
6878	024164	062706	000030		ADD #30,SP		4331
6879	024170	062706	000012	12\$:	ADD #12,SP		4313
6880	024174	104467			TRAP 67		4349
6881	024176	006000			ROR R0		
6882	024200	103002			BHIS 14\$		
6883	024202	000167	177324	13\$:	JMP 2\$		
6884	024206	005204		14\$:	INC R4	: REG.SEL	4312
6885	024210	020427	000015		CMP R4,#15	: REG.SEL,*	
6886	024214	003772			BLE 13\$		
6887	024216	005102			COM R2	: TST.PAT	4354
6888	024220	005201			INC R1	: TWICE	4309
6889	024222	020127	000001		CMP R1,#1	: TWICE,*	
6890	024226	003002			BGT 15\$		
6891	024230	000167	177274		JMP 1\$		
6892	024234	005067	165672	15\$:	CLR ML.REG+232		4357
6893	024240	027767	165564	165426	CMP @ML.REG+130,DRIVE.TYPE		4364
6894	024246	001431			BEQ 16\$		
6895	024250	104455			TRAP 55		4367
6896	024252	000161			.WORD 161		
6897	024254	007444			.WORD ASYNC		

```
6899 ;ML4
6900 ;
6901
6902 024256 000000 .WORD 0
6903 024260 012746 006630 MOV #PHR.4,-(SP) ;
6904 024264 012746 007350 MOV #REG.7,-(SP) ;
6905 024270 012746 005356 MOV #TWO.FMT,-(SP) ;
6906 024274 012746 000003 MOV #3,-(SP) ;
6907 024300 010600 MOV SP,R0 ; SP,*
6908 024302 104414 TRAP 14
6909 024304 010416 MOV R4,(SP) ; SAVE,*
6910 024306 016746 165362 MOV DRIVE.TYPE,-(SP)
6911 024312 012746 004224 MOV #FMT.2,-(SP)
6912 024316 012746 000003 MCV #3,-(SP)
6913 024322 010600 MOV SP,R0 ; SP,*
6914 024324 104414 TRAP 14
6915 024326 062706 000016 ADD #16,SP ;
6916 024332 005305 16$: DEC R5 ; DODU.FLG
6917 024334 001004 BNE 17$
6918 024336 016700 165624 MOV ML.LUN,R0 ;
6919 024342 104451 TRAP 51
6920 024344 104444 TRAP 44
6921 024346 062706 000006 17$: ADD #6,SP ;
6922 024352 000207 RTS PC ;
6923
6924 ; Routine Size: 215 words
6925 ; Maximum stack depth per invocation: 26 words
6930
6931
6935
6939 024354 T5::
6940 024354 004767 177116 1$: JSR PC,$T5 ;
6941 024360 104466 TRAP 66
6942 024362 006000 ROR R0
6943 024364 103773 BLO 1$
6944 024366 000207 RTS PC
6945
6946 ; Routine Size: 6 words
6947 ; Maximum stack depth per invocation: 0 words
6952 ; 4380 !<BLF/PAGE>
```

22-Oct-1980 10:47:44
22-Oct-1980 10:45:32

TOPS-20 BLiss-16 V2(206)
PA:<NEALE>BL2ML4.BLI.2 (49)

```

6954 :ML4
6955 :
6956 :
6957 : 4381 !
6958 : 4382 !
6959 : 4383 !
6960 : 4384 BGNTST;
6961 : 4385
6962 : 4386 !++
6963 : 4387 TEST NUMBER: TST 6
6964 : 4388
6965 : 4389 TEST NAME: REGISTER SELECTION TEST
6966 : 4390
6967 : 4391 TEST DESCRIPTION:
6968 : 4392 THIS TEST TESTS FOR UNIQUE REGISTER
6969 : 4393 SELECTION BY FIRST WRITING A BACKGROUND
6970 : 4394 PATTERN INTO ALL READ/WRITE REGISTERS
6971 : 4395
6972 : 4396 IT THEN WRITES A COMPLIMENT
6973 : 4397 BACKGROUND PATTERN INTO ONE REGISTER
6974 : 4398 AND READS THE REMAINING UNWRITTEN
6975 : 4399 REGISTERS FOR AN UNCHANGED BACKGROUND
6976 : 4400 PAT
6977 : 4401
6978 : 4402 THIS PROCEDURE IS REPEATED UNTIL ALL
6979 : 4403 REGISTERS HAVE BEEN WRITTEN WITH A
6980 : 4404 COMPLIMENT BACKGROUND PATTERN.
6981 : 4405
6982 : 4406 IMPLICIT INPUTS: NONE
6983 : 4407
6984 : 4408
6985 : 4409 !--
6986 : 4410
6987 : 4411 local
6988 : 4412 DODU FLG,
6989 : 4413 TST_PAT,
6990 : 4414 ERR_FLG,
6991 : 4415 index;
6992 : 4416
6993 : 4417 BGNSUB;
6994 : 4418 CLR MBUS;
6995 : 4419 DODU FLG = ZERO;
6996 : 4420 TST_PAT = %o'125252';
6997 : 4421 WRT_CS1 (.TST_PAT, 0);
6998 : 4422 WRT_ER (.TST_PAT, 6);
6999 : 4423 WRT_DA (.TST_PAT, 3);
7000 : 4424 WRT_PA (.TST_PAT, 8);
7001 : 4425 WRT_E1 (.TST_PAT, 13);
7002 : 4426 WRT_E2 (.TST_PAT, 14);
7003 : 4427
7004 : 4428 incr CNT_1 from 0 to 4 do
7005 : 4429 begin
7006 : 4430 TST_PAT = not .TST_PAT;
7007 : 4431
7008 : 4432 case .CNT_1 from 0 to 4 of

```

```

!DROP UNIT FLAG
!TEST PATTERN
!ERROR FLAG PASSED TO ROUTINE;
!POINTS TO REGISTER PRESENTLY BEING TESTED.

```

```

!LOAD TST PAT WITH ALTERNATE 1'S & 0'S
!WRITE A BACKGROUND INTO ALL THE DIRECTLY
!ACCESSABLE READ WRITE REGISTERS

```

```

!WRITE A COMPLIMENT PATTERN INTO ONE REGISTER
!GENERATE THE COMPLIMENT PAT
!SELECT THE REGISTER TO WRITE INTO

```

```

7010 :ML4
7011 :
7012 :
7013 : 4433 set
7014 : 4434
7015 : 4435 [0] : WRT_CS1 (.TST_PAT 0); !FIRST PASS WRITE COMP PAT TO MLCS1
7016 : 4436
7017 : 4437
7018 : 4438 [1] : WRT_ER (.TST_PAT, 6); !SECOND PASS WRITE COMP PAT TO MLER
7019 : 4439
7020 : 4440
7021 : 4441 [2] : WRT_DA (.TST_PAT, 3); !THIRD PASS WRITE COMP PAT TO MLDA
7022 : 4442
7023 : 4443
7024 : 4444 [3] : WRT_PA (.TST_PAT, 8); !FORTH PASS WRITE COMP PAT TO MLPA
7025 : 4445
7026 : 4446
7027 : 4447 [4] : WRT_E1 (.TST_PAT, 13); !FIFTH PASS WRITE COMP PAT TO MLE1
7028 : 4448 tes;
7029 : 4449
7030 : 4450 TST_PAT = not .TST_PAT; !COMPLIMENT TST_PAT BACK TO BACKGROUND
7031 : 4451
7032 : 4452 incr CNT_2 from .CNT_1 + 1 to 5 do !NOW READ THE REMAINING UNWRITTEN REGISTERS FOR AN
7033 : 4453 begin !UNCHANGED BACKGROUND
7034 : 4454
7035 : 4455 case .CNT_2 from 0 to 5 of !SELECT THE REGISTER TO READ
7036 : 4456 set
7037 : 4457
7038 : 4458 [0] : RD_CS1 (.TST_PAT, index = 0, ERR_FLG); !READ MLCS1
7039 : 4459
7040 : 4460 [1] : RD_ER (.TST_PAT, index = 6, ERR_FLG); !READ MLER
7041 : 4461
7042 : 4462 [2] : RD_DA (.TST_PAT, index = 3, ERR_FLG); !READ MLDA
7043 : 4463
7044 : 4464 [3] : RD_PA (.TST_PAT, index = 8, ERR_FLG); !READ MLPA
7045 : 4465
7046 : 4466 [4] : RD_E1 (.TST_PAT, index = 13, ERR_FLG); !READ MLE1
7047 : 4467
7048 : 4468 [5] : RD_E2 (.TST_PAT, index = 14, ERR_FLG); !READ MLE2
7049 : 4469
7050 : 4470 tes;
7051 : 4471
7052 : 4472 if .ERR_FLG IS_SET !SEE IF READ FOUND AN ERROR
7053 : 4473 then
7054 : 4474 begin
7055 : 4475
7056 : 4476 selectone .CNT_2 of
7057 : 4477 set
7058 : 4478
7059 : 4479
7060 : 4480
7061 : 4481
7062 : 4482
7063 : 4483
7064 : 4484
  
```

22-Oct-1980 10:47:44 TOPS-20 Bliss-16 V2(206)
 22-Oct-1980 10:45:32 PA:<NEALE>BL2ML4.BLI.2 (49)

7066 :ML4

22-Oct-1980 10:47:44

TOPS-20 Bliss-16 V2(206)

7067 :

22-Oct-1980 10:45:32

PA:<NEALE>BL2ML4.BLI.2 (49)

7068 : 4485

[0 to 2] :

7069 : 4486

ERRDF (110, ASYNC, 0);

!ASYNC MODULE FAILURE

7070 : 4487

7071 : 4488

[3 to 5] :

7072 : 4489

ERRDF (110, ARR_DAT, 0);

!ARRAY DATA MODULE FAILURE

7073 : 4490

7074 : 4491

tes;

7075 : 4492

PRINTB (THR_FMT, WRD 38, WRD 37, WRD 10);

7076 : 4493

PRINTB (FMT_16, .ML_REG [.index, REGISTER_ADD], .WT_DATA, .RD_DATA);

7077 : 4494

DODU_FLG = ONE;

7078 : 4495

end;

7079 : 4496

7080 : 4497

end;

7081 : 4498

7082 : 4499

end;

7083 : 4500

7084 : 4501

ENDSUB;

7085 : 4502

7086 : 4503

if .DODU_FLG IS_SET

!DROP THIS UNIT IF DODU_FLG IS_SET

7087 : 4504

then

7088 : 4505

begin

7089 : 4506

DODU (.ML_LUN);

7090 : 4507

DOCLN;

7091 : 4508

end;

7092 : 4509

7093 : 4510

ENDTST;

7098 : 024370 004167 157464

\$T6:

JSR R1,\$SAVE5

:

4379

7103 024374 005746

TST -(SP)

7104 024376 104402

1\$:

TRAP 2

:

4415

7105 024400 152777 000040 165332

BISB #40,@ML.REG+40

:

4417

7106 024406 016703 165556

MOV ML.DUT,R3

7107 024412 042703 177770

BIC #177770,R3

7108 024416 142777 000007 165314

BICB #7,@ML.REG+40

7109 024424 150377 165310

BISB R3,@ML.REG+40

7110 024430 005005

CLR R5

: DODU.FLG

4419

7111 024432 012704 125252

MOV #-52526,R4

: *,TST.PAT

4420

7112 024436 010446

MOV R4,-(SP)

: TST.PAT,*

4421

7113 024440 005046

CLR -(SP)

7114 024442 004767 167526

JSR PC,WRT.CS1

7115 024446 010416

MOV R4,(SP)

: TST.PAT,*

4422

7116 024450 012746 000006

MOV #6,-(SP)

7117 024454 004767 167672

JSR PC,WRT.ER

7118 024460 010416

MOV R4,(SP)

: TST.PAT,*

4423

7119 024462 012746 000003

MOV #3,-(SP)

Address	OpCode	Operand 1	Operand 2	Operand 3	Instruction	Comments	Address
7121							22-Oct-1980 10:47:44 TOPS
7122							22-Oct-1980 10:45:32 PA:<
7123							
7124	024466	004767	170036		JSR	PC,WRT.DA	
7125	024472	010416			MOV	R4,(SP) ; TST.PAT,*	4424
7126	024474	012746	000010		MOV	#10,-(SP)	
7127	024500	004767	170360		JSR	PC,WRT.PA	
7128	024504	010416			MOV	R4,(SP) ; TST.PAT,*	4425
7129	024506	012746	000015		MOV	#15,-(SP)	
7130	024512	004767	170554		JSR	PC,WRT.E1	
7131	024516	010416			MOV	R4,(SP) ; TST.PAT,*	4426
7132	024520	012746	000016		MOV	#16,-(SP)	
7133	024524	004767	170750		JSR	PC,WRT.E2	
7134	024530	005001			CLR	R1 ; CNT.1	4428
7135	024532	005104		2\$:	COM	R4 ; TST.PAT	4430
7136	024534	010103			MOV	R1,R3 ; CNT.1,*	4432
7137	024536	006303			ASL	R3	
7138	024540	066307	024544		ADD	3\$(R3),PC	
7139	024544	000012		3\$:	.WORD	4\$-3\$	
7140	024546	000024			.WORD	5\$-3\$	
7141	024550	000040			.WORD	6\$-3\$	
7142	024552	000054			.WORD	7\$-3\$	
7143	024554	000070			.WORD	8\$-3\$	
7144	024556	010446		4\$:	MOV	R4,-(SP) ; TST.PAT,*	4436
7145	024560	005046			CLR	-(SP)	
7146	024562	004767	167406		JSR	PC,WRT.CS1	
7147	024566	000427			BR	9\$;	4432
7148	024570	010446		5\$:	MOV	R4,-(SP) ; TST.PAT,*	4439
7149	024572	012746	000006		MOV	#6,-(SP)	
7150	024576	004767	167550		JSR	PC,WRT.ER	
7151	024602	000421			BR	9\$;	4432
7152	024604	010446		6\$:	MOV	R4,-(SP) ; TST.PAT,*	4442
7153	024606	012746	000003		MOV	#3,-(SP)	
7154	024612	004767	167712		JSR	PC,WRT.DA	
7155	024616	000413			BR	9\$;	4432
7156	024620	010446		7\$:	MOV	R4,-(SP) ; TST.PAT,*	4445
7157	024622	012746	000010		MOV	#10,-(SP)	
7158	024626	004767	170232		JSR	PC,WRT.PA	
7159	024632	000405			BR	9\$;	4432
7160	024634	010446		8\$:	MOV	R4,-(SP) ; TST.PAT,*	4448
7161	024636	012746	000015		MOV	#15,-(SP)	
7162	024642	004767	170424		JSR	PC,WRT.E1	
7163	024646	005104		9\$:	COM	R4 ; TST.PAT	4451
7164	024650	010103			MOV	R1,R3 ; CNT.1,CNT.2	4453
7165	024652	000576			BR	22\$	
7166	024654	010300		10\$:	MOV	R3,R0 ; CNT.2,*	4456
7167	024656	006300			ASL	R0	
7168	024660	066007	024664		ADD	11\$(R0),PC	
7169	024664	000014		11\$:	.WORD	12\$-11\$	
7170	024666	000036			.WORD	13\$-11\$	
7171	024670	000062			.WORD	14\$-11\$	
7172	024672	000106			.WORD	15\$-11\$	
7173	024674	000132			.WORD	16\$-11\$	
7174	024676	000156			.WORD	17\$-11\$	
7175	024700	010446		12\$:	MOV	R4,-(SP) ; TST.PAT,*	4460

Line No	Op	Op2	Op3	Op4	Op5	Op6	Op7	Op8	Op9	Op10	Op11	Op12
7177												
7178												
7179												
7180	024702	005002			CLR	R2			; INDEX			
7181	024704	005046			CLR	-(SP)						
7182	024706	012746	000030		MOV	#30, -(SP)						
7183	024712	060616			ADD	SP, (SP)			; ERR.FLG,*			
7184	024714	004767	167324		JSR	PC,RD.CS1						
7185	024720	000461			BR	18\$					4456	
7186	024722	010446		13\$:	MOV	R4, -(SP)			; TST.PAT,*		4463	
7187	024724	012702	000006		MOV	#6, R2			; *INDEX			
7188	024730	010246			MOV	R2, -(SP)			; INDEX,*			
7189	024732	012746	000030		MOV	#30, -(SP)						
7190	024736	060616			ADD	SP, (SP)			; ERR.FLG,*			
7191	024740	004767	167456		JSR	PC,RD.ER						
7192	024744	000447			BR	18\$					4456	
7193	024746	010446		14\$:	MOV	R4, -(SP)			; TST.PAT,*		4466	
7194	024750	012702	000003		MOV	#3, R2			; *INDEX			
7195	024754	010246			MOV	R2, -(SP)			; INDEX,*			
7196	024756	012746	000030		MOV	#30, -(SP)						
7197	024762	060616			ADD	SP, (SP)			; ERR.FLG,*			
7198	024764	004767	167610		JSR	PC,RD.DA						
7199	024770	000435			BR	18\$					4456	
7200	024772	010446		15\$:	MOV	R4, -(SP)			; TST.PAT,*		4469	
7201	024774	012702	000010		MOV	#10, R2			; *INDEX			
7202	025000	010246			MOV	R2, -(SP)			; INDEX,*			
7203	025002	012746	000030		MOV	#30, -(SP)						
7204	025006	060616			ADD	SP, (SP)			; ERR.FLG,*			
7205	025010	004767	170134		JSR	PC,RD.PA						
7206	025014	000423			BR	18\$					4456	
7207	025016	010446		16\$:	MOV	R4, -(SP)			; TST.PAT,*		4472	
7208	025020	012702	000015		MOV	#15, R2			; *INDEX			
7209	025024	010246			MOV	R2, -(SP)			; INDEX,*			
7210	025026	012746	000030		MOV	#30, -(SP)						
7211	025032	060616			ADD	SP, (SP)			; ERR.FLG,*			
7212	025034	004767	170316		JSR	PC,RD.E1						
7213	025040	000411			BR	18\$					4456	
7214	025042	010446		17\$:	MOV	R4, -(SP)			; TST.PAT,*		4475	
7215	025044	012702	000016		MOV	#16, R2			; *INDEX			
7216	025050	010246			MOV	R2, -(SP)			; INDEX,*			
7217	025052	012746	000030		MOV	#30, -(SP)						
7218	025056	060616			ADD	SP, (SP)			; ERR.FLG,*			
7219	025060	004767	170512		JSR	PC,RD.E2						
7220	025064	026627	000030	000001	18\$:	CMP	30(SP), #1		; ERR.FLG,*		4478	
7221	025072	001064			BNE	21\$						
7222	025074	005703			TST	R3			; CNT.2		4482	
7223	025076	002410			BLT	19\$						
7224	025100	020327	000002		CMP	R3, #2			; CNT.2,*			
7225	025104	003005			BGT	19\$						
7226	025106	104455			TRAP	55					4486	
7227	025110	000156			.WORD	156						
7228	025112	007444			.WORD	ASYN						
7229	025114	000000			.WORD	0						
7230	025116	000412			BR	20\$					4482	
7231	025120	020327	000003	19\$:	CMP	R3, #3			; CNT.2,*			

22-Oct-1980 10:47:44 TOPS
22-Oct-1980 10:45:32 PA:<

7233								
7234				:ML4				
7235				:				
7236	025124	002407			BLT	20\$		
7237	025126	020327	000005		CMP	R3,#5	: CNT.2,*	
7238	025132	003004			BGT	20\$		
7239	025134	104455			TRAP	55	:	4489
7240	025136	000156			.WORD	156		
7241	025140	007534			.WORD	ARR.DAT		
7242	025142	000000			.WORD	0		
7243	025144	012746	005740	20\$:	MOV	#WRD.10,-(SP)	:	4492
7244	025150	012746	006232		MOV	#WRD.37,-(SP)		
7245	025154	012746	006242		MOV	#WRD.38,-(SP)		
7246	025160	012746	005366		MOV	#THR.FMT,-(SP)		
7247	025164	012746	000004		MOV	#4,-(SP)		
7248	025170	010600			MOV	SP,R0	: SP,*	
7249	025172	104414			TRAP	14	:	
7250	025174	016716	164472		MOV	RD.DATA,(SP)	:	4493
7251	025200	016746	164464		MOV	WT.DATA,-(SP)		
7252	025204	010200			MOV	R2,R0	: INDEX,*	
7253	025206	006300			ASL	R0		
7254	025210	006300			ASL	R0		
7255	025212	006300			ASL	R0		
7256	025214	016046	011700		MOV	ML.REG(R0),-(SP)		
7257	025220	012746	005154		MOV	#FMT.16,-(SP)		
7258	025224	012746	000004		MOV	#4,-(SP)		
7259	025230	010600			MOV	SP,R0	: SP,*	
7260	025232	104414			TRAP	14		
7261	025234	012705	000001		MOV	#1,R5	: *,DODU.FLG	4494
7262	025240	062706	000022		ADD	#22,SP	:	4480
7263	025244	062706	000006	21\$:	ADD	#6,SP	:	4454
7264	025250	005203		22\$:	INC	R3	: CNT.2	4453
7265	025252	020327	000005		CMP	R3,#5	: CNT.2,*	
7266	025256	003002			BGT	23\$		
7267	025260	000167	177370		JMP	10\$		
7268	025264	022626		23\$:	CMP	(SP)+,(SP)+	:	4429
7269	025266	005201			INC	R1	: CNT.1	4428
7270	025270	020127	000004		CMP	R1,#4	: CNT.1,*	
7271	025274	003002			BGT	24\$		
7272	025276	000167	177230		JMP	2\$		
7273	025302	062706	000016	24\$:	ADD	#16,SP	:	4415
7274	025306	104467			TRAP	67	:	4499
7275	025310	006000			ROR	R0		
7276	025312	103002			BHIS	25\$		
7277	025314	000167	177056		JMP	1\$		
7278	025320	005305		25\$:	DEC	R5	: DODU.FLG	4503
7279	025322	001004			BNE	26\$		
7280	025324	016700	164636		MOV	ML.LUN,R0	:	4506
7281	025330	104451			TRAP	51		
7282	025332	104444			TRAP	44		
7283	025334	005726		26\$:	TST	(SP)+	:	4379
7284	025336	000207			RTS	PC		

7285
7286
7287
: Routine Size: 244 word-
: Maximum stack depth per invocation: 28 words

7289
7290
7291
7296
7297
7301
7305 025340
7306 025340 004767 177024
7307 025344 104466
7308 025346 006000
7309 025350 103773
7310 025352 000207

:ML4
:

T6::
1\$: JSR PC,\$T6
TRAP 66
ROR R0
BLO 1\$
RTS PC

22-Oct-1980 10:47:44 TOPS
22-Oct-1980 10:45:32 PA:<

4508

7311
7312 ; Routine Size: 6 words
7313 ; Maximum stack depth per invocation: 0 words
7318
7319
7320 ; 4511 !<BLF/PAGE>

22-Oct-1980 10:47:44
22-Oct-1980 10:45:32

TOPS-20 Bliss-16 V2(206)
PA:<NEALE>BL2ML4.BLI.2 (50)

```

7322 :ML4
7323 :
7324 :
7325 : 4512 |
7326 : 4513 |
7327 : 4514 | BGNTST;
7328 : 4515 |
7329 : 4516 | ++
7330 : 4517 | TEST NUMBER: TST 7
7331 : 4518 |
7332 : 4519 | TEST NAME: PRINT SERIAL NUMBER
7333 : 4520 |
7334 : 4521 | TEST DESCRIPTION:
7335 : 4522 | UPON A YES RESPONSE TO THE
7336 : 4523 | SOFTWARE QUESTION 'PRINT SERIAL NO?'
7337 : 4524 | PRINT OUT THE DRIVE UNDER TEST
7338 : 4525 | SERIAL NUMBER.
7339 : 4526 |
7340 : 4527 | IMPLICIT INPUTS: NONE
7341 : 4528 |
7342 : 4529 |
7343 : 4530 | --
7344 : 4531 |
7345 : 4532 | external
7346 : 4533 | PRSN;
7347 : 4534 |
7348 : 4535 | if .PRSN IS_SET then PRINTB (FMT_3, .MLSN);
7349 : 4536 |
7350 : 4537 | ENDTST;

```

!LOCATION WHERE ANSWER TO SW QUESTION IS STORED
!PRINT DRIVE SERIAL NO. IF ANS IS YES.

```
7354 :
7355 : .GLOBL PRSN
7356 :
7357 :
```

```

7361 025354 026727 154760 000001 $T7:  CMP  PRSN,#1           ; 4535
7362 025362 001012          BNE  1$                ;
7363 025364 017746 164450  MOV  @ML.REG+140,-(SP)
7364 025370 012746 004312  MOV  #FMT.3,-(SP)
7365 025374 012746 000002  MOV  #2,-(SP)
7366 025400 010600          MOV  SP,R0             ; SP,*
7367 025402 104414          TRAP 14
7368 025404 062706 000006  ADD  #6,SP
7369 025410 000207          RTS   PC              ; 4510
7377 :
7378 :
7382 :
7386 025412          T7::
7387 025412 004767 177736 1$:  JSR  PC,$T7           ; 4535
7388 025416 104466          TRAP 66
7389 025420 006000          ROR  R0
7390 025422 103773          BLO  1$
7391 025424 000207          RTS   PC

```

22-Oct-1980 10:47:44
22-Oct-1980 10:45:32

TOPS-20 Bliss-16 V2(206)
PA:<NEALE>BL2ML4.BLI.2 (51)

7397 :ML4
7398 :
7399 :
7400 :
7401 :
7402 :
7403 :
7404 :
7405 :
7406 :
7407 :
7408 :
7409 :
7410 :
7411 :
7412 :
7413 :
7414 :
7415 :
7416 :
7417 :
7418 :
7419 :
7420 :
7421 :
7422 :
7423 :
7424 :
7425 :
7426 :
7427 :
7428 :
7429 :
7430 :
7431 :
7432 :
7433 :
7434 :
7435 :
7436 :
7437 :
7438 :
7439 :
7440 :
7441 :
7442 :
7443 :
7444 :
7445 :
7446 :
7447 :
7448 :
7449 :
7450 :
7451 :

4539
4540
4541
4542
4543
4544
4545
4546
4547
4548
4549
4550
4551
4552
4553
4554
4555
4556
4557
4558
4559
4560
4561
4562
4563
4564
4565
4566
4567
4568
4569
4570
4571
4572
4573
4574
4575
4576
4577
4578
4579
4580
4581
4582
4583
4584
4585
4586
4587
4588
4589
4590

BGNTST;

!++

TEST NUMBER: TST 8

TEST NAME: C-BUS PARITY TEST

TEST DESCRIPTION:

TEST THE CONTROL BUS PARITY
DETECTION AND GENERATING BY:

1. WRITING BAD PARITY TO DEVICE
AND TEST CPAR SET.
2. WRITING GOOD PARITY TO DEVICE
AND TEST CPAR CLR.
3. READING DEVICE AND TEST GOOD
PARITY GENERATION BY READING
MCPE CLR.

IMPLICIT INPUTS: NONE

!--

local

SAVE,
TST_PAT;

!TEMPORARY SAVE LOCATION
!TEST PATTERN

if .PAR_DIS IS_NOT_SET
then

!SEE IF PARITY IS DISABLED

begin
TST_PAT = %0'125252';

!BEGIN IF PARITY IS ENABLE
!ALTERNATING 1, 0 PATTERN

incr TWICE from 1 to 2 do

!REPEAT LOOP TWICE

begin
CLR_MBUS;
PAT = ONE;
MLDA = .TST_PAT;

!GENERATE EVEN PARITY BY SETTING THE PAT BIT
!WRITE BAD PARITY ON CONTROL BUS

if .CPAR IS_NOT_SET
then

!READ CPAR BIT SET

begin
ERRDF (7, ASYNC, 0);
PRINTB (THR_FMT, WRD_5, WRD_7, WRD_9);
end;

!ERROR IF NOT SET

CLR_MBUS;
MLDA = .TST_PAT;

!CLEAR OUT PAT BIT
!WRITE ODD PARITY CONTROL BUS

22-Oct-1980 10:47:44 TOPS-20 Bliss-16 V2(206)
 22-Oct-1980 10:45:32 PA:<NEALE>BL2ML4.BLI.2 (51)

```

7453 :ML4
7454 :
7455 :
7456 : 4591
7457 : 4592
7458 : 4593
7459 : 4594
7460 : 4595
7461 : 4596
7462 : 4597
7463 : 4598
7464 : 4599
7465 : 4600
7466 : 4601
7467 : 4602
7468 : 4603
7469 : 4604
7470 : 4605
7471 : 4606
7472 : 4607
7473 : 4608
7474 : 4609
7475 : 4610
7476 : 4611
7477 : 4612
7478 : 4613
7479 : 4614
7480 : 4615
7481 : 4616
7485 :
  if .CPAR IS_SET
  then
  begin
  ERRDF (8, ASYNC, 0);
  PRINTB (THR_FMT, WRD_6, WRD_7, WRD_9);
  end;
  CLR MBUS;
  SAVE = .MLDA;
  if .MCPE IS_SET
  then
  begin
  ERRDF (9, ASYNC, 0);
  PRINTB (THR_FMT, WRD_6, WRD_7, WRD_8);
  end;
  TST_PAT = .TST_PAT^ONE;
  end;
  else
  PRINTB (TWO_FMT, WRD_7, WRD_53);
  ENDTST;

```

```

7489 025426 004167 156370 $T8: JSR R1,$SAVE3 ; 4537
7490 025432 005767 162672 TST PAR.DIS ; 4571
7491 025436 001171 BNE 5$ ;
7492 025440 012701 125252 MOV #-52526,R1 ; *,TST.PAT 4574
7493 025444 012702 000001 MOV #1,R2 ; *,TWICE 4576
7494 025450 152777 000040 164262 1$: BISB #40,@ML.REG+40 ; 4577
7495 025456 016700 164506 MOV ML.DUT,RO
7496 025462 042700 177770 BIC #177770,RO
7497 025466 142777 000007 164244 BICB #7,@ML.REG+40
7498 025474 150077 164240 BISB RO,@ML.REG+40
7499 025500 152777 000020 164232 BISB #20,@ML.REG+40 ; 4579
7500 025506 010177 164216 MOV R1,@ML.REG+30 ; TST.PAT,* 4580
7501 025512 132777 000010 164240 BITB #10,@ML.REG+60 ; 4582
7502 025520 001022 BNE 2$ ;
7503 025522 104455 TRAP 55 ; 4585
7504 025524 000007 .WORD 7
7505 025526 007444 .WORD ASYNC
7506 025530 000000 .WORD 0

```

7508										22-Oct-1980 10:47:44	TOPS
7509										22-Oct-1980 10:45:32	PA:<
7510											
7511	025532	012746	005726				MOV	#WRD.9,-(SP)			
7512	025536	012746	005676				MOV	#WRD.7,-(SP)			4586
7513	025542	012746	005662				MOV	#WRD.5,-(SP)			
7514	025546	012746	005366				MOV	#THR.FMT,-(SP)			
7515	025552	012746	000004				MOV	#4,-(SP)			
7516	025556	010600					MOV	SP,RO		; SP,*	
7517	025560	104414					TRAP	14			
7518	025562	062706	000012				ADD	#12,SP			4584
7519	025566	152777	000040	164144	2\$:		BISB	#40,@ML.REG+40			4587
7520	025574	016700	164370				MOV	ML.DUT,RO			
7521	025600	042700	177770				BIC	#177770,RO			
7522	025604	142777	000007	164126			BICB	#7,@ML.REG+40			
7523	025612	150077	164122				BISB	RO,@ML.REG+40			
7524	025616	010177	164106				MOV	R1,@ML.REG+30		; TST.PAT,*	4590
7525	025622	132777	000010	164130			BITB	#10,@ML.REG+60			4592
7526	025630	001422					BEQ	3\$			
7527	025632	104455					TRAP	55			4595
7528	025634	000010					.WORD	10			
7529	025636	007444					.WORD	ASYNC			
7530	025640	000000					.WORD	0			
7531	025642	012746	005726				MOV	#WRD.9,-(SP)			4596
7532	025646	012746	005676				MOV	#WRD.7,-(SP)			
7533	025652	012746	005670				MOV	#WRD.6,-(SP)			
7534	025656	012746	005366				MOV	#THR.FMT,-(SP)			
7535	025662	012746	000004				MOV	#4,-(SP)			
7536	025666	010600					MOV	SP,RO		; SP,*	
7537	025670	104414					TRAP	14			
7538	025672	062706	000012				ADD	#12,SP			4594
7539	025676	152777	000040	164034	3\$:		BISB	#40,@ML.REG+40			4597
7540	025704	016700	164260				MOV	ML.DUT,RO			
7541	025710	042700	177770				BIC	#177770,RO			
7542	025714	142777	000007	164016			BICB	#7,@ML.REG+40			
7543	025722	150077	164012				BISB	RO,@ML.REG+40			
7544	025726	017703	163776				MOV	@ML.REG+30,R3		; *,SAVE	4600
7545	025732	032777	020000	163740			BIT	#20000,@ML.REG			4602
7546	025740	001422					BEQ	4\$			
7547	025742	104455					TRAP	55			4605
7548	025744	000011					.WORD	11			
7549	025746	007444					.WORD	ASYNC			
7550	025750	000000					.WORD	0			
7551	025752	012746	005712				MOV	#WRD.8,-(SP)			4606
7552	025756	012746	005676				MOV	#WRD.7,-(SP)			
7553	025762	012746	005670				MOV	#WRD.6,-(SP)			
7554	025766	012746	005366				MOV	#THR.FMT,-(SP)			
7555	025772	012746	000004				MOV	#4,-(SP)			
7556	025776	010600					MOV	SP,RO		; SP,*	
7557	026000	104414					TRAP	14			
7558	026002	062706	000012				ADD	#12,SP			4604
7559	026006	006301			4\$:		ASL	R1		; TST.PAT	4609
7560	026010	005202					INC	R2		; TWICE	4576
7561	026012	020227	000002				CMP	R2,#2		; TWICE,*	
7562	026016	003614					BLE	1\$			


```
7564 ;ML4
7565 ;
7566
7567 026020 000207
7568 026022 012746 006426
7569 026026 012746 005676
7570 026032 012746 005356
7571 026036 012746 000003
7572 026042 010600
7573 026044 104414
7574 026046 062706 000010
7575 026052 000207
7576
7577 ; Routine Size: 139 words
7578 ; Maximum stack depth per invocation: 9 words
7583
7584
7588
7592 026054
7593 026054 004767 177346
7594 026060 104466
7595 026062 006000
7596 026064 103773
7597 026066 000207
7598
7599 ; Routine Size: 6 words
7600 ; Maximum stack depth per invocation: 0 words
7605
7606
7607 ; 4617 !<BLF/PAGE>
```

22-Oct-1980 10:47:44 TOPS
22-Oct-1980 10:45:32 PA:<

4571
4614

4537

4614

22-Oct-1980 10:47:44
 22-Oct-1980 10:45:32

TOPS-20 BLISS-16 V2(206)
 PA:<NEALE>BL2ML4.BLI.2 (52)

```

7609 ;ML4
7610 :
7611 :
7612 : 4618 !
7613 : 4619 !
7614 : 4620 BGNTST;
7615 : 4621
7616 : 4622 !++
7617 : 4623 TEST NUMBER: TST 9
7618 : 4624
7619 : 4625 TEST NAME: MEMORY SIZING TEST
7620 : 4626
7621 : 4627 TEST DESCRIPTION:
7622 : 4628
7623 : 4629 THIS TESTS THE ML11'S SIZING
7624 : 4630 LOGIC BY COMPARING THE
7625 : 4631 OPERATORS INPUTED NUMBER OF ARRAYS
7626 : 4632 PRESENT TO THE ML11 SIZING
7627 : 4633 NUMBER OF ARRAYS PRESENT
7628 : 4634 THE DRIVE IS DROPPED ON DETECTED ERRORS.
7629 : 4635 IMPLICIT INPUTS:
7630 : 4636
7631 : 4637 OP_NUM_ARR:
7632 : 4638 LOADED DURING INIT CODE AND
7633 : 4639 STORES OPERATORS INPUT TO THE
7634 : 4640 SOFTWARE QUESTION:
7635 : 4641 NUMBER OF ARRAYS PRESENT?
7636 : 4642
7637 : 4643
7638 : 4644 !--
7639 : 4645
7640 : 4646 Local
7641 : 4647 DODU_FLG; !DROP UNIT FLAG
7642 : 4648
7643 : 4649 BGNSUB;
7644 : 4650 CLR MBUS;
7645 : 4651 DODU_FLG = ZERO;
7646 : 4652
7647 : 4653 if (.OP_NUM_ARR + 1) neq .ML_NUM_ARR !SEE IF DRIVE SIZED SAME NO. OF ARRAYS AS UP INPUTED
7648 : 4654 then
7649 : 4655 begin
7650 : 4656 ERRDF (10, ASYNC, 0); !IF NOT EQL THEN ERROR AND SET DODU_FLG
7651 : 4657 PRINTB (TWO_FMT, FNC 1, WRD 14);
7652 : 4658 PRINTB (FMT 2, (.OP_NUM_ARR + 1), .ML_NUM_ARR);
7653 : 4659 DODU_FLG = ONE;
7654 : 4660 end;
7655 : 4661
7656 : 4662 ENDSUB;
7657 : 4663
7658 : 4664 if .DODU_FLG IS_SET !DROP THIS UNIT IF DODU_FLG IS_SET
7659 : 4665 then
7660 : 4666 begin
7661 : 4667 DODU (.ML_LUN);
7662 : 4668 DOCLN;
7663 : 4669 end;
  
```

22-Oct-1980 10:47:44 TOPS-20 Bliss-16 V2(206)
22-Oct-1980 10:45:32 PA:<NEALE>BL2ML4.BLI.2 (52)

Address	Instruction	Comments	Label
7665	:ML4		
7666	:		
7667	:		
7668	:	4670	
7669	:	4671	ENDTST:
7673			
7677	026070 004167 155726	\$T9: JSR R1,\$SAVE3	4616
7678	026074 104402	1\$: TRAP 2	4647
7679	026076 152777 000040 163634	BISB #40,@ML.REG+40	4649
7680	026104 016701 164060	MOV ML.DUT,R1	
7681	026110 042701 177770	BIC #177770,R1	
7682	026114 142777 000007 163616	BICB #7,@ML.REG+40	
7683	026122 150177 163612	BISB R1,@ML.REG+40	
7684	026126 005002	CLR R2 ; DODU.FLG	4651
7685	026130 016701 162166	MOV OP.NUM.ARR,R1	4653
7686	026134 005201	INC R1	
7687	026136 017703 163656	MOV @ML.REG+120,R3	
7688	026142 006203	ASR R3	
7689	026144 006203	ASR R3	
7690	026146 006203	ASR R3	
7691	026150 000303	SWAB R3	
7692	026152 042703 177740	BIC #177740,R3	
7693	026156 010100	MOV R1,R0	
7694	026160 020003	CMP R0,R3	
7695	026162 001444	BEQ 2\$	
7696	026164 104455	TRAP 55 ;	4656
7697	026166 000012	.WORD 12	
7698	026170 007444	.WORD ASYNC	
7699	026172 000000	.WORD 0	
7700	026174 012746 005774	MOV #WRD.14,-(SP)	4657
7701	026200 012746 006752	MOV #FNC.1,-(SP)	
7702	026204 012746 005356	MOV #TWO.FMT,-(SP)	
7703	026210 012746 000003	MOV #3,-(SP)	
7704	026214 010600	MOV SP,R0 ; SP,*	
7705	026216 104414	TRAP 14	
7706	026220 017703 163574	MOV @ML.REG+120,R3 ;	4658
7707	026224 006203	ASR R3	
7708	026226 006203	ASR R3	
7709	026230 006203	ASR R3	
7710	026232 000303	SWAB R3	
7711	026234 042703 177740	BIC #177740,R3	
7712	026240 010316	MOV R3,(SP)	
7713	026242 016746 162054	MOV OP.NUM.ARR,-(SP)	
7714	026246 005216	INC (SP)	
7715	026250 012746 004224	MOV #FMT.2,-(SP)	
7716	026254 012746 000003	MOV #3,-(SP)	
7717	026260 010600	MOV SP,R0 ; SP,*	
7718	026262 104414	TRAP 14	

```
7720 ;ML4
7721 ;
7722 ;
7723 026264 012702 000001 MOV #1,R2 ; *,DODU.FLG 4659
7724 026270 062706 000016 ADD #16,SP ; 4655
7725 026274 104467 2$: TRAP 67 ; 4660
7726 026276 006000 ROR R0 ;
7727 026300 103675 BLO 1$ ;
7728 026302 005302 DEC R2 ; DODU.FLG 4664
7729 026304 001004 BNE 3$ ;
7730 026306 016700 163654 MOV ML.LUN,R0 ; 4667
7731 026312 104451 TRAP 51 ;
7732 026314 104444 TRAP 44 ;
7733 026316 000207 3$: RTS PC ; 4616
7734
7735 ; Routine Size: 76 words
7736 ; Maximum stack depth per invocation: 11 words
7741
7742 /
7746
7750 026320 T9::
7751 026320 004767 177544 1$: JSR PC,$T9 ; 4669
7752 026324 104466 TRAP 66 ;
7753 026326 006000 ROR R0 ;
7754 026330 103773 BLO 1$ ;
7755 026332 000207 RTS PC ;
7756
7757 ; Routine Size: 6 words
7758 ; Maximum stack depth per invocation: 0 words
7763
7764
7765 ; 4672 !<BLF/PAGE>
```

22-Oct-1980 10:47:44
 22-Oct-1980 10:45:32

TOPS-20 Bliss-16 V2(206)
 PA:<NEALE>BL2ML4.BLI.2 (53)

```

7767 :ML4
7768 :
7769 :
7770 :      4673 !
7771 :      4674 !
7772 :      4675 BGNTST;
7773 :      4676
7774 :      4677 !++
7775 :      4678 ! TEST NUMBER: TST 10
7776 :      4679
7777 :      4680 ! TEST NAME: NO-OP FUNCTION TEST
7778 :      4681
7779 :      4682 ! TEST DESCRIPTION:
7780 :      4683
7781 :      4684 ! TEST IF THE DRIVE CAN PERFORM
7782 :      4685 ! A NO_OP FUNCTION WITH OUT
7783 :      4686 ! HANGING THE DRIVE.
7784 :      4687
7785 :      4688 ! A NO_OP FUNCTION IS WRITTEN
7786 :      4689 ! INTO MLCST.
7787 :      4690 ! THEN GO AND ERROR BITS ARE
7788 :      4691 ! CHECKED FOR CORRECT STATES.
7789 :      4692
7790 :      4693 ! THIS UNIT IS DROPPED ON DETECTED
7791 :      4694 ! ERRORS.
7792 :      4695
7793 :      4696 ! IMPLICIT INPUTS: NONE
7794 :      4697
7795 :      4698
7796 :      4699 !--
7797 :      4700
7798 :      4701 local
7799 :      4702 DODU_FLG; !DROP UNIT FLAG
7800 :      4703
7801 :      4704 BGNSUB;
7802 :      4705 CLR MBUS;
7803 :      4706 DODU_FLG = ZERO;
7804 :      4707 MLCST = NOOP; !DO A NOOP FUNCTION
7805 :      4708 DELAY (ONE_US); !DELAY 1 US
7806 :      4709
7807 :      4710 if .GO IS_SET !SEE IF GO STILL SET
7808 :      4711 then
7809 :      4712 begin
7810 :      4713 ERRDF (11, ASYNC, 0); !ERROR AND SET DODU_FLG IF STILL SET
7811 :      4714 PRINTB (FIV_FMT, WRD_1, PHR_2, WRD_11, FNC_2, WRD_19);
7812 :      4715 DODU_FLG = ONE;
7813 :      4716 end;
7814 :      4717
7815 :      4718 if .ILF IS_SET !SEE ILF SET
7816 :      4719 then
7817 :      4720 begin
7818 :      4721 ERRDF (12, ASYNC, 0); !ERROR AND SET DODU_FLG IF SET
7819 :      4722 PRINTB (FIV_FMT, WRD_3, PHR_5, WRD_11, FNC_2, WRD_19);
7820 :      4723 DODU_FLG = ONE;
7821 :      4724 end;

```

22-Oct-1980 10:47:44 TOPS-20 Bliss-16 V2(206)
22-Oct-1980 10:45:32 PA:<NEALE>BL2ML4.BLI.2 (53)

```

7823 :ML4
7824 :
7825 :
7826 : 4725
7827 : 4726 if .OPI IS_SET !SEE IF CPI SET
7828 : 4727 then
7829 : 4728 begin
7830 : 4729 ERRDF (13, ASYNC, 0); !ERROR AND SET DODU_FLG IF SET
7831 : 4730 PRINTB (FIV_FMT, WRD_4, PHR_5, WRD_11, FNC_2, WRD_19);
7832 : 4731 DODU_FLG = ONE;
7833 : 4732 end;
7834 : 4733
7835 : 4734 ENDSUB;
7836 : 4735
7837 : 4736 if .DODU_FLG IS_SET !DROP THIS UNIT IF DODU_FLG IS_SET
7838 : 4737 then
7839 : 4738 begin
7840 : 4739 DODU (.ML_LUN);
7841 : 4740 DOCLN;
7842 : 4741 end;
7843 : 4742
7844 : 4743 ENDTST;

```

```

7848
7852 026334 004167 155446 $T10: JSR R1,$SAVE2 ; 4671
7853 026340 005746 TST -(SP) ;
7854 026342 104402 1$: TRAP 2 ; 4702
7855 026344 152777 000040 163366 BISB #40,@ML.REG+40 ; 4704
7856 026352 016701 163612 MOV ML,DUT,R1
7857 026356 042701 177770 BIC #177770,R1
7858 026362 142777 000007 163350 BICB #7,@ML.REG+40
7859 026370 150177 163344 BISB R1,@ML.REG+40
7860 026374 005002 CLR R2 ; DODU.FLG 4706
7861 026376 012777 000001 163274 MOV #1,@ML.REG ; 4707
7862 026404 012700 000001 MOV #1,R0 ; *,$STMP2 4708
7863 026410 001410 2$: BEQ 5$
7864 026412 016701 153500 MOV L$DLY,R1 ; *,$STMP1
7865 026416 001403 BEQ 4$
7866 026420 005016 3$: CLR (SP) ; $STMP
7867 026422 005301 DEC R1 ; $STMP1
7868 026424 001375 BNE 3$
7869 026426 005300 4$: DEC R0 ; $STMP2
7870 026430 000767 BR 2$
7871 026432 132777 000001 163240 5$: BITB #1,@ML.REG ; 4710
7872 026440 001430 BEQ 6$
7873 026442 104455 TRAP 55 ; 4713
7874 026444 000013 .WORD 13
7875 026446 007444 .WORD ASYNC
7876 026450 000000 .WORD 0

```

Line No.	Address	Instruction	Comments	Sequence No.
7878			:ML4	
7879			:	
7880			:	
7881	026452	012746	006040 MOV #WRD.19,-(SP)	4714
7882	026456	012746	006766 MOV #FNC.2,-(SP)	
7883	026462	012746	005750 MOV #WRD.11,-(SP)	
7884	026466	012746	006560 MOV #PHR.2,-(SP)	
7885	026472	012746	005630 MOV #WRD.1,-(SP)	
7886	026476	012746	005414 MOV #FIV.FMT,-(SP)	
7887	026502	012746	000006 MOV #6,-(SP)	
7888	026506	010600	MOV SP,RO	: SP,*
7889	026510	104414	TRAP 14	
7890	026512	012702	000001 MOV #1,R2	: *,DODU.FLG 4715
7891	026516	062706	000016 ADD #16,SP	: 4712
7892	026522	132777	000001 163230 6\$: BIT #1,@ML.REG+60	: 4718
7893	026530	001430	BEQ 7\$	
7894	026532	104455	TRAP 55	: 4721
7895	026534	000014	.WORD 14	
7896	026536	007444	.WORD ASYNC	
7897	026540	000000	.WORD 0	
7898	026542	012746	006040 MOV #WRD.19,-(SP)	: 4722
7899	026546	012746	006766 MOV #FNC.2,-(SP)	
7900	026552	012746	005750 MOV #WRD.11,-(SP)	
7901	026556	012746	006646 MOV #PHR.5,-(SP)	
7902	026562	012746	005646 MOV #WRD.3,-(SP)	
7903	026566	012746	005414 MOV #FIV.FMT,-(SP)	
7904	026572	012746	000006 MOV #6,-(SP)	
7905	026576	010600	MOV SP,RO	: SP,*
7906	026600	104414	TRAP 14	
7907	026602	012702	000001 MOV #1,R2	: *,DODU.FLG 4723
7908	026606	062706	000016 ADD #16,SP	: 4720
7909	026612	032777	020000 163140 7\$: BIT #20000,@ML.REG+60	: 4726
7910	026620	001430	BEQ 8\$	
7911	026622	104455	TRAP 55	: 4729
7912	026624	000015	.WORD 15	
7913	026626	007444	.WORD ASYNC	
7914	026630	000000	.WORD 0	
7915	026632	012746	006040 MOV #WRD.19,-(SP)	: 4730
7916	026636	012746	006766 MOV #FNC.2,-(SP)	
7917	026642	012746	005750 MOV #WRD.11,-(SP)	
7918	026646	012746	006646 MOV #PHR.5,-(SP)	
7919	026652	012746	005654 MOV #WRD.4,-(SP)	
7920	026656	012746	005414 MOV #FIV.FMT,-(SP)	
7921	026662	012746	000006 MOV #6,-(SP)	
7922	026666	010600	MOV SP,RO	: SP,*
7923	026670	104414	TRAP 14	
7924	026672	012702	000001 MOV #1,R2	: *,DODU.FLG 4731
7925	026676	062706	000016 ADD #16,SP	: 4728
7926	026702	104467	TRAP 67	: 4732
7927	026704	006000	ROR RO	
7928	026706	103615	BLO 1\$	
7929	026710	005302	DEC R2	: DODU.FLG 4736
7930	026712	001004	BNE 9\$	
7931	026714	016700	163246 MOV ML.LUN,RO	: 4739
7932	026720	104451	TRAP 51	

22-Oct-1980 10:47:44 TOPS.
22-Oct-1980 10:45:32 PA:<

7934
7935
7936
7937 026722 104444
7938 026724 005726
7939 026726 000207
7940
7941
7942
7947
7948
7952
7956 026730
7957 026730 004767 177400
7958 026734 104466
7959 026736 006000
7960 026740 103773
7961 026742 000207
7962
7963
7964
7969
7970
7971 ; 4744 !<BLF/PAGE>

;ML4

;

9\$: TRAP 44
TST (SP)+
RTS PC

; Routine Size: 126 words
; Maximum stack depth per invocation: 11 words

T10::

1\$: JSR PC,\$T10
TRAP 66
ROR R0
BLO 1\$
RTS PC

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

4671

4741

22-Oct-1980 10:47:44
22-Oct-1980 10:45:32

TOPS-20 Bliss-16 V2(206)
PA:<NEALE>BL2ML4.BLI.2 (54)

7973 :ML4
7974 :
7975 :
7976 :
7977 :
7978 :
7979 :
7980 :
7981 :
7982 :
7983 :
7984 :
7985 :
7986 :
7987 :
7988 :
7989 :
7990 :
7991 :
7992 :
7993 :
7994 :
7995 :
7996 :
7997 :
7998 :
7999 :
8000 :
8001 :
8002 :
8003 :
8004 :
8005 :
8006 :
8007 :
8008 :
8009 :
8010 :
8011 :
8012 :
8013 :
8014 :
8015 :
8016 :
8017 :
8018 :
8019 :
8020 :
8021 :
8022 :
8023 :
8024 :
8025 :
8026 :
8027 :

```

4745 :
4746 :
4747 : BGNTST;
4748 :
4749 : ++
4750 : TEST NUMBER: TST 11
4751 :
4752 : TEST NAME: WRITE CHECK FUNCTION TEST
4753 :
4754 : TEST DESCRIPTION:
4755 :
4756 :     TEST IF THE DRIVE CAN PERFORM
4757 :     A WRITE CHECK FUNCTION WITHOUT
4758 :     HANGING THE DRIVE
4759 :
4760 :
4761 :     A WRITE CHECK FUNCTION IS WRITTEN
4762 :     INTO MLCS1.
4763 :     THEN GO AND ERROR BITS ARE
4764 :     CHECKED FOR CORRECT STATES
4765 :
4766 :     THIS UNIT IS DROPPED ON DETECTED
4767 :     ERRORS.
4768 :
4769 : IMPLICIT INPUTS: NONE
4770 :
4771 : --
4772 :
4773 : local
4774 :     DODU_FLG;                                !DROP UNIT FLAG
4775 :
4776 : BGNSUB;
4777 : CLR MBUS;
4778 : DODU_FLG = ZERO;
4779 : FIRST_BLK_XFER ();                            !SET UP A FIRST BLK XFERR
4780 : MLCS1 = WRT_CHK;                             !DO A WRITE CHECK FUNCTION
4781 :
4782 : if .GO IS_NOT_SET                            !SEE IF THE GO BIT GOT SET
4783 : then
4784 :     begin
4785 :         ERRDF (14, ASYNC, 0);                !ERROR IF NOT SET
4786 :         PRINTB (FIV_FMT, WRD_1, PHR_1, WRD_12, FNC_4, WRD_19);
4787 :
4788 :         if .DRY IS_NOT_SET                    !SEE IF THE DRY IS SET WHILE GO IS CLEARED
4789 :         then
4790 :             begin
4791 :                 ERRDF (15, ASYNC, 0);        !ERROR IF CLEARED
4792 :                 PRINTB (FIV_FMT, WRD_2, PHR_1, WRD_43, WRD_1, PHR_6);
4793 :             end;
4794 :
4795 :         DODU_FLG = ONE;                       !SET DODU_FLG
4796 :     end

```

22-Oct-1980 10:47:44 TOPS-20 Bliss-16 V2(206)
 22-Oct-1980 10:45:32 PA:<NEALE>BL2ML4.BLI.2 (54)

```

8029 :ML4
8030 :
8031 :
8032 :      4797 else
8033 :      4798
8034 :      4799     if .DRY IS_SET                !THE GO IS SET SO SEE IF DRY IS CLEARED
8035 :      4800     then
8036 :      4801     begin
8037 :      4802     ERRDF (16, ASYNC, 0);          !ERROR IF DRY IS SET
8038 :      4803     PRINTB (FIV_FMT, WRD_2, PHR_2, WRD_43, WRD_1, PHR_5);
8039 :      4804     DODU_FLG = ONE;
8040 :      4805     end;
8041 :      4806
8042 :      4807     if .ILF IS_SET                !SEE IF ILF IS SET
8043 :      4808     then
8044 :      4809     begin
8045 :      4810     ERRDF (17, ASYNC, 0);          !ERROR IF SET
8046 :      4811     PRINTB (FIV_FMT, WRD_3, PHR_5, WRD_12, FNC_4, WRD_19);
8047 :      4812     DODU_FLG = ONE;
8048 :      4813     end;
8049 :      4814
8050 :      4815     if .OPI IS_SET                !SEE IF OPI IS SET
8051 :      4816     then
8052 :      4817     begin
8053 :      4818     ERRDF (18, ASYNC, 0);          !ERROR IF SET
8054 :      4819     PRINTB (FIV_FMT, WRD_4, PHR_5, WRD_12, FNC_4, WRD_19);
8055 :      4820     DODU_FLG = ONE;
8056 :      4821     end;
8057 :      4822
8058 :      4823     DELAY (FRTY_US);                !WAIT FO XFERR TO COMPLTE
8059 :      4824
8060 :      4825     if .DRY IS_NOT_SET            !SEE IF DRY IS SETS AFTER XFERR
8061 :      4826     then
8062 :      4827     begin
8063 :      4828
8064 :      4829     if .GO IS_SET                    !IF DRY IS NOT SET THEN SEE IF GO IS SET
8065 :      4830     then
8066 :      4831     begin
8067 :      4832     CLR_MBUS;                        !IF THE GO IS SET THEN TRY TO CLR GO
8068 :      4833
8069 :      4834     if .GO IS_SET then ERRDF (19, ASYNC, 0) else ERRDF (20, SYNC, 0);
8070 :      4835
8071 :      4836     !IF GO IS STILL SET THEN ASYNC FAILURE
8072 :      4837     PRINTB (FIV_FMT, WRD_1, PHR_2, WRD_11, FNC_4, WRD_19); !ELSE SYNC MODLE FAILURE
8073 :      4838     end
8074 :      4839     else
8075 :      4840     begin
8076 :      4841     ERRDF (21, ASYNC, 0);          !ERROR GO AND DRY BOTH CLEARED
8077 :      4842     PRINTB (FIV_FMT, WRD_2, PHR_1, WRD_11, FNC_4, WRD_19);
8078 :      4843     end;
8079 :      4844
8080 :      4845     DODU_FLG = ONE;                !SET THE DODU_FLG
8081 :      4846     end;
8082 :      4847
8083 :      4848     if .GO IS_SET                    !SEE IF THE GO IS STILL SET
  
```

22-Oct-1980 10:47:44 TOPS-20 Bliss-16 V2(206)
22-Oct-1980 10:45:32 PA:<NEALE>BL2ML4.BLI.2 (54)

```

8085 :ML4
8086 :
8087 :
8088 :      4849 then
8089 :      4850 begin
8090 :      4851 CLR_MBUS;                !TRY TO CLR GO IF STILL SET
8091 :      4852
8092 :      4853 if .GO IS_SET then ERRDF (22, ASYNC, 0) else ERRDF (23, SYNC, 0);  !IF GO IS STILL SET THE ASYNC FAILUR
8093 :      4854
8094 :      4855 PRINTB (FIV_FMT, WRD_1, PHR_2, WRD_11, FNC_4, WRD_19);        !ELSE SYNC MODULE FAILURE
8095 :      4856 DODU_FLG = ONE;
8096 :      4857 end;
8097 :      4858
8098 :      4859 ENDSUB;
8099 :      4860
8100 :      4861 if .DODU_FLG IS_SET                !DROP THIS UNIT IF DODU_FLG IS_SET
8101 :      4862 then
8102 :      4863 begin
8103 :      4864 DODU (.ML_LUN);
8104 :      4865 DOCLN;
8105 :      4866 end;
8106 :      4867
8107 :      4868 ENDTST;
8111
8115 026744 004167 155036      $T11: JSR R1,$SAVE2 ; 4743
8116 026750 005746          TST -(SP)
8117 026752 104402          1$: TRAP 2 ; 4774
8118 026754 152777 000040 162756 BISB #40,@ML.REG+40 ; 4776
8119 026762 016701 163202 MOV ML.DUT,R1
8120 026766 042701 177770 BIC #177770,R1
8121 026772 142777 000007 162740 BICB #7,@ML.REG+40
8122 027000 150177 162734 BISB R1,@ML.REG+40
8123 027004 005002 CLR R2 ; DODU.FLG 4778
8124 027006 004767 163506 JSR PC,FIRST.BLK.XFER ; 4779
8125 027012 012777 000051 162660 MOV #51,@ML.REG ; 4780
8126 027020 132777 000001 162652 BITB #1,@ML.REG ; 4782
8127 027026 001057 BNE 2$
8128 027030 104455 TRAP 55 ; 4785
8129 027032 000016 .WORD 16
8130 027034 007444 .WORD ASYNC
8131 027036 000000 .WORD 0
8132 027040 012746 006040 MOV #WRD.19,-(SP) ; 4786
8133 027044 012746 007002 MOV #FNC.4,-(SP)
8134 027050 012746 005760 MOV #WRD.12,-(SP)
8135 027054 012746 006542 MOV #PHR.1,-(SP)
8136 027060 012746 005630 MOV #WRD.1,-(SP)
8137 027064 012746 005414 MOV #FIV_FMT,-(SP)
8138 027070 012746 000006 MOV #6,-(SP)

```

8140										22-Oct-1980 10:47:44	TOPS
8141										22-Oct-1980 10:45:32	PA:<
8142											
8143	027074	010600					MOV	SP,R0		; SP,*	
8144	027076	104414					TRAP	14			
8145	027100	132777	000200	162642			BITB	#200,@ML.REG+50		:	4788
8146	027106	001056					BNE	3\$			
8147	027110	104455					TRAP	55		:	4791
8148	027112	000017					.WORD	17			
8149	027114	007444					.WORD	ASync			
8150	027116	000000					.WORD	0			
8151	027120	012746	006660				MOV	#PHR.6,-(SP)		:	4792
8152	027124	012746	005630				MOV	#WRD.1,-(SP)			
8153	027130	012746	006306				MOV	#WRD.43,-(SP)			
8154	027134	012746	006542				MOV	#PHR.1,-(SP)			
8155	027140	012746	005634				MOV	#WRD.2,-(SP)			
8156	027144	012746	005414				MOV	#FIV.FMT,-(SP)			
8157	027150	012746	000006				MOV	#6,-(SP)			
8158	027154	010600					MOV	SP,R0		; SP,*	
8159	027156	104414					TRAP	14			
8160	027160	062706	000016				ADD	#16,SP		:	4790
8161	027164	000427					BR	3\$:	4795
8162	027166	105777	162556		2\$:		TSTB	@ML.REG+50		:	4799
8163	027172	100030					BPL	4\$			
8164	027174	104455					TRAP	55		:	4802
8165	027176	000020					.WORD	20			
8166	027200	007444					.WORD	ASync			
8167	027202	000000					.WORD	0			
8168	027204	012746	006646				MOV	#PHR.5,-(SP)		:	4803
8169	027210	012746	005630				MOV	#WRD.1,-(SP)			
8170	027214	012746	006306				MOV	#WRD.43,-(SP)			
8171	027220	012746	006560				MOV	#PHR.2,-(SP)			
8172	027224	012746	005634				MOV	#WRD.2,-(SP)			
8173	027230	012746	005414				MOV	#FIV.FMT,-(SP)			
8174	027234	012746	000006				MOV	#6,-(SP)			
8175	027240	010600					MOV	SP,R0		; SP,*	
8176	027242	104414					TRAP	14			
8177	027244	012702	000001		3\$:		MOV	#1,R2		; *.DODU.FLG	4804
8178	027250	062706	000016				ADD	#16,SP		:	4801
8179	027254	132777	000001	162476	4\$:		BITB	#1,@ML.REG+60		:	4807
8180	027262	001430					BEQ	5\$			
8181	027264	104455					TRAP	55		:	4810
8182	027266	000021					.WORD	21			
8183	027270	007444					.WORD	ASync			
8184	027272	000000					.WORD	0			
8185	027274	012746	006040				MOV	#WRD.19,-(SP)		:	4811
8186	027300	012746	007002				MOV	#FNC.4,-(SP)			
8187	027304	012746	005760				MOV	#WRD.12,-(SP)			
8188	027310	012746	006646				MOV	#PHR.5,-(SP)			
8189	027314	012746	005646				MOV	#WRD.3,-(SP)			
8190	027320	012746	005414				MOV	#FIV.FMT,-(SP)			
8191	027324	012746	000006				MOV	#6,-(SP)			
8192	027330	010600					MOV	SP,R0		; SP,*	
8193	027332	104414					TRAP	14			
8194	027334	012702	000001				MOV	#1,R2		; *.DODU.FLG	4812

```

8196           ;ML4
8197           ;
8198
8199 027340 062706 000016           ADD #16,SP           ;
8200 027344 032777 020000 162406 5$: BIT #20000,@AML.REG+60 ;
8201 027352 001430           BEQ 6$              ;
8202 027354 104455           TRAP 55             ;
8203 027356 000022           .WORD 22          ;
8204 027360 007444           .WORD ASYNC       ;
8205 027362 000000           .WORD 0           ;
8206 027364 012746 006040           MOV #WRD.19,-(SP) ; 4819
8207 027370 012746 007002           MOV #FNC.4,-(SP) ;
8208 027374 012746 005760           MOV #WRD.12,-(SP) ;
8209 027400 012746 006646           MOV #PHR.5,-(SP) ;
8210 027404 012746 005654           MOV #WRD.4,-(SP) ;
8211 027410 012746 005414           MOV #FIV.FMT,-(SP) ;
8212 027414 012746 000006           MOV #6,-(SP)     ;
8213 027420 010600           MOV SP,R0        ; SP,*
8214 027422 104414           TRAP 14          ;
8215 027424 012702 000001           MOV #1,R2        ; *,DODU.FLG 4820
8216 027430 062706 000016           ADD #16,SP       ; 4817
8217 027434 012700 000050           MOV #50,R0       ; *,$$TMP2 4823
8218 027440 001410           BEQ 10$          ;
8219 027442 016701 152450           MOV L$DLY,R1    ; *,$$TMP1
8220 027446 001403           BEQ 9$          ;
8221 027450 005016           CLR (SP)        ; $$TMP
8222 027452 005301           DEC R1          ; $$TMP1
8223 027454 001375           BNE 8$          ;
8224 027456 005300           DEC R0          ; $$TMP2
8225 027460 000767           BR 7$           ;
8226 027462 132777 000200 162260 10$: BITB #200,@AML.REG+50 ; 4825
8227 027470 001106           BNE 15$         ;
8228 027472 132777 000001 162200           BITB #1,@AML.REG ; 4829
8229 027500 001452           BEQ 13$         ;
8230 027502 152777 000040 162230           BISB #40,@AML.REG+40 ; 4831
8231 027510 016701 162454           MOV ML.DUT,R1   ;
8232 027514 042701 177770           BIC #177770,R1 ;
8233 027520 142777 000007 162212           BICB #7,@AML.REG+40 ;
8234 027526 150177 162206           BISB R1,@AML.REG+40 ;
8235 027532 132777 000001 162140           BITB #1,@AML.REG ; 4834
8236 027540 001405           BEQ 11$        ;
8237 027542 104455           TRAP 55        ;
8238 027544 000023           .WORD 23      ;
8239 027546 007444           .WORD ASYNC   ;
8240 027550 000000           .WORD 0       ;
8241 027552 000404           BR 12$       ;
8242 027554 104455           TRAP 55      ; 11$:
8243 027556 000024           .WORD 24     ;
8244 027560 007500           .WORD SYNC   ;
8245 027562 000000           .WORD 0      ;
8246 027564 012746 006040           MOV #WRD.19,-(SP) ; 4837
8247 027570 012746 007002           MOV #FNC.4,-(SP) ;
8248 027574 012746 005750           MOV #WRD.11,-(SP) ;
8249 027600 012746 006560           MOV #PHR.2,-(SP) ;
8250 027604 012746 005630           MOV #WRD.1,-(SP) ;

```

22-Oct-1980 10:47:44 TOPS
22-Oct-1980 10:45:32 PA:<

8252								22-Oct-1980 10:47:44	TOPS
8253					:ML4			22-Oct-1980 10:45:32	PA:<
8254					:				
8255	027610	012746	005414			MOV	#FIV.FMT,-(SP)		
8256	027614	012746	000006			MOV	#6,-(SP)		
8257	027620	010600				MOV	SP,R0	: SP,*	
8258	027622	104414				TRAP	14		
8259	027624	000424				BR	14\$:	4829
8260	027626	104455		13\$:		TRAP	55	:	4841
8261	027630	000025				.WORD	25		
8262	027632	007444				.WORD	ASync		
8263	027634	000000				.WORD	0		
8264	027636	012746	006040			MOV	#WRD.19,-(SP)	:	4842
8265	027642	012746	007002			MOV	#FNC.4,-(SP)		
8266	027646	012746	005750			MOV	#WRD.11,-(SP)		
8267	027652	012746	006542			MOV	#PHR.1,-(SP)		
8268	027656	012746	005634			MOV	#WRD.2,-(SP)		
8269	027662	012746	005414			MOV	#FIV.FMT,-(SP)		
8270	027666	012746	000006			MOV	#6,-(SP)		
8271	027672	010600				MOV	SP,R0	: SP,*	
8272	027674	104414				TRAP	14		
8273	027676	012702	000001	14\$:		MOV	#1,R2	: *,DODU.FLG	4845
8274	027702	062706	000016			ADD	#16,SP	:	4827
8275	027706	132777	000001	161764	15\$:	BITB	#1,@ML.REG	:	4848
8276	027714	001455				BEQ	18\$		
8277	027716	152777	000040	162014		BISB	#40,@ML.REG+40	:	4850
8278	027724	016701	162240			MOV	ML.DUT,R1		
8279	027730	042701	177770			BIC	#177770,R1		
8280	027734	142777	000007	161776		BICB	#7,@ML.REG+40		
8281	027742	150177	161772			BISB	R1,@ML.REG+40		
8282	027746	132777	000001	161724		BITB	#1,@ML.REG	:	4853
8283	027754	001405				BEQ	16\$		
8284	027756	104455				TRAP	55		
8285	027760	000026				.WORD	26		
8286	027762	007444				.WORD	ASync		
8287	027764	000000				.WORD	0		
8288	027766	000404				BR	17\$		
8289	027770	104455		16\$:		TRAP	55		
8290	027772	000027				.WORD	27		
8291	027774	007500				.WORD	SynC		
8292	027776	000000				.WORD	0		
8293	030000	012746	006040	17\$:		MOV	#WRD.19,-(SP)	:	4855
8294	030004	012746	007002			MOV	#FNC.4,-(SP)		
8295	030010	012746	005750			MOV	#WRD.11,-(SP)		
8296	030014	012746	006560			MOV	#PHR.2,-(SP)		
8297	030020	012746	005630			MOV	#WRD.1,-(SP)		
8298	030024	012746	005414			MOV	#FIV.FMT,-(SP)		
8299	030030	012746	000006			MOV	#6,-(SP)		
8300	030034	010600				MOV	SP,R0	: SP,*	
8301	030036	104414				TRAP	14		
8302	030040	012702	000001			MOV	#1,R2	: *,DODU.FLG	4856
8303	030044	062706	000016			ADD	#16,SP	:	4850
8304	030050	104467		18\$:		TRAP	67	:	4857
8305	030052	006000				ROR	R0		
8306	030054	103002				BHIS	19\$		

```
8308 ;ML4
8309 ;
8310 ;
8311 030056 000167 176670 JMP 1$
8312 030062 005302 19$: DEC R2 ; DODU.FLG 4861
8313 030064 001004 BNE 20$
8314 030066 016700 162074 MOV ML.LUN,R0 ; 4864
8315 030072 104451 TRAP 51
8316 030074 104444 TRAP 44
8317 030076 005726 20$: TST (SP)+ ; 4743
8318 030100 000207 RTS PC
8319
8320 ; Routine Size: 303 words
8321 ; Maximum stack depth per invocation: 18 words
8326
8327
8331
8335 030102 T11::
8336 030102 004767 176636 1$: JSR PC,$T11 ; 4866
8337 030106 104466 TRAP 66
8338 030110 006000 ROR R0
8339 030112 103773 BLO 1$
8340 030114 000207 RTS PC
8341
8342 ; Routine Size: 6 words
8343 ; Maximum stack depth per invocation: 0 words
8348
8349
8350 ; 4869 !<BLF/PAGE>
```

22-Oct-1980 10:47:44 TOPS-20 Bliss-16 V2(206)
 22-Oct-1980 10:45:32 PA:<NEALE>BL2ML4.BLI.2 (55)

```

8352 ;ML4
8353 ;
8354 ;
8355 : 4870 !
8356 : 4871 BGNTST;
8357 : 4872
8358 : 4873 !++
8359 : 4874 ! TEST NUMBER: TST 12
8360 : 4875
8361 : 4876 ! TEST NAME: WRITE FUNCTION TEST
8362 : 4877
8363 : 4878 ! TEST DESCRIPTION:
8364 : 4879 ! TEST IF THE DRIVE CAN PERFORM A WRITE FUNCTION WITHOUT
8365 : 4880 ! HANGING THE DRIVE.
8366 : 4881
8367 : 4882 ! A WRITE FUNCTION IS WRITTEN INTO MLCS1. THEN GO AND ERROR BITS ARE
8368 : 4883 ! CHECKED FOR CORRECT STATUS. THIS UNIT IS DROPPED ON DETECTED ERRORS.
8369 : 4884 !--
8370 : 4885
8371 : 4886 local
8372 : 4887 DODU_FLG; !DROP UNIT FLAG
8373 : 4888
8374 : 4889 BGNSUB;
8375 : 4890 CLR_MBUS;
8376 : 4891 DODU_FLG = ZERO;
8377 : 4892 FIRST_BLK_XFER (); !SET UP A FIRST BLOCK XFERR
8378 : 4893 MLCS1 = write; !DO A WRITE FUNCTION
8379 : 4894
8380 : 4895 if .GO IS_NOT_SET !SEE IF THE GO IS SET
8381 : 4896 then
8382 : 4897 begin !ERROR IF NOT SET
8383 : 4898 ERRDF (24, ASYNC, 0);
8384 : 4899 PRINTB (FIV_FMT, WRD_1, PHR_1, WRD_12, FNC_5, WRD_19);
8385 : 4900
8386 : 4901 if .DRY IS_NOT_SET !SEE IF DRY SET WITH GO CLEAR
8387 : 4902 then
8388 : 4903 begin
8389 : 4904 ERRDF (25, ASYNC, 0);
8390 : 4905 PRINTB (FIV_FMT, WRD_2, PHR_1, WRD_43, WRD_1, PHR_6);
8391 : 4906 end;
8392 : 4907
8393 : 4908 DODU_FLG = ONE; !SET DODU_FLG
8394 : 4909 end
8395 : 4910 else !GO IS SET DURING FUNCTION
8396 : 4911
8397 : 4912 if .DRY IS_SET !SEE IF DRY CLEAR WITH GO SET
8398 : 4913 then
8399 : 4914 begin !ERROR IF SET
8400 : 4915 ERRDF (26, ASYNC, 0);
8401 : 4916 PRINTB (FIV_FMT, WRD_2, PHR_2, WRD_43, WRD_1, PHR_5);
8402 : 4917 DODU_FLG = ONE;
8403 : 4918 end;
8404 : 4919
8405 : 4920 if .ILF IS_SET !DID FUNCTION CAUSE ILF
8406 : 4921 then
  
```



```

8408 ;ML4
8409 ;
8410 ;
8411 ; 4922 begin !ERROR IF YES
8412 ; 4923 ERRDF (27, ASYNC, 0);
8413 ; 4924 PRINTB (FIV_FMT, WRD_3, PHR_5, WRD_12, FNC_5, WRD_19);
8414 ; 4925 DODU_FLG = ONE;
8415 ; 4926 end;
8416 ; 4927
8417 ; 4928 if .OPI IS_SET !DID FUNCTION CAUSE OPI
8418 ; 4929 then
8419 ; 4930 begin !ERROR IF YES
8420 ; 4931 ERRDF (28, ASYNC, 0);
8421 ; 4932 PRINTB (FIV_FMT, WRD_4, PHR_5, WRD_12, FNC_5, WRD_19);
8422 ; 4933 DODU_FLG = ONE;
8423 ; 4934 end;
8424 ; 4935
8425 ; 4936 DELAY (FRTY_US); !WAIT FOR XFERR TO COMPLETE
8426 ; 4937
8427 ; 4938 if .DRY IS_NOT_SET !SEE IF DRY CLEARED AFTER XFERR
8428 ; 4939 then
8429 ; 4940 begin
8430 ; 4941
8431 ; 4942 if .GO IS_SET !TST GO CLR IF DRY NOT SET
8432 ; 4943 then
8433 ; 4944 begin
8434 ; 4945 CLR_MBUS; !CLEAR GO IF STILL SET
8435 ; 4946
8436 ; 4947 if .GO IS_SET then ERRDF (29, ASYNC, 0) else ERRDF (30, SYNC, 0);
8437 ; 4948
8438 ; 4949 !TST GO TO DETERMINE FAILING MOD
8439 ; 4950 PRINTB (FIV_FMT, WRD_1, PHR_2, WRD_11, FNC_5, WRD_19);
8440 ; 4951 end
8441 ; 4952 else !DRY NOT SET AND GO CLEARED
8442 ; 4953 begin !REPORT ERROR
8443 ; 4954 ERRDF (31, ASYNC, 0);
8444 ; 4955 PRINTB (FIV_FMT, WRD_2, PHR_1, WRD_11, FNC_5, WRD_19);
8445 ; 4956 end;
8446 ; 4957
8447 ; 4958 DODU_FLG = ONE; !SET DODU_FLG
8448 ; 4959 end;
8449 ; 4960
8450 ; 4961 if .GO IS_SET !SEE IF GO CLEARED AFTER XFERR.
8451 ; 4962 then
8452 ; 4963 begin
8453 ; 4964 CLR_MBUS; !CLEAR GO IF STILL SET
8454 ; 4965
8455 ; 4966 if .GO IS_SET then ERRDF (32, ASYNC, 0) else ERRDF (33, SYNC, 0); !TST GO TO DETERMINE FAILING MOD
8456 ; 4967
8457 ; 4968 PRINTB (FIV_FMT, WRD_1, PHR_2, WRD_11, FNC_5, WRD_19);
8458 ; 4969 DODU_FLG = ONE;
8459 ; 4970 end;
8460 ; 4971
8461 ; 4972 ENDSUB;
8462 ; 4973
  
```

22-Oct-1980 10:47:44
 22-Oct-1980 10:45:32

TOPS-20 Bliss-16 V2(206)
 PA:<NEALE>BL2ML4.BLI.2 (55)

8464 :ML4

8465 :

8466

8467 : 4974 if .TRE IS_SET

8468 : 4975 then

8469 : 4976 begin

8470 : 4977 ERRDF (115, INTER, 0);

8471 : 4978 PRINTB (SIX_FMT, WRD_61, WRD_20, PHR_5, WRD_12, FNC_5, WRD_19);

8472 : 4979 DODU_FLG = ONE;

8473 : 4980 end;

8474 : 4981

8475 : 4982 if .DODU_FLG IS_SET

8476 : 4983 then

8477 : 4984 begin

8478 : 4985 DODU (.ML_LUN);

8479 : 4986 DOCLN;

8480 : 4987 end;

8481 : 4988

8482 : 4989 ENDTST;

8486

8490 030116 004167 153664 \$T12: JSR R1,\$SAVE2 ; 4868

8491 030122 005746 TST -(SP) ;

8492 030124 104402 1\$: TRAP 2 ; 4887

8493 030126 152777 000040 161604 BISB #40,@ML.REG+40 ; 4889

8494 030134 016701 162030 MOV ML.DUT,R1 ;

8495 030140 042701 177770 BIC #177770,R1 ;

8496 030144 142777 000007 161566 BICB #7,@ML.REG+40 ;

8497 030152 150177 161562 BISB R1,@ML.REG+40 ;

8498 030156 005002 CLR R2 ; DODU.FLG 4891

8499 030160 004767 162334 JSR PC,FIRST.BLK.XFER ; 4892

8500 030164 012777 000061 161506 MOV #61,@ML.REG ; 4893

8501 030172 132777 000001 161500 BITB #1,@ML.REG ; 4895

8502 030200 001057 BNE 2\$;

8503 030202 104455 TRAP 55 ; 4898

8504 030204 000030 .WORD 30 ;

8505 030206 007444 .WORD ASYNC ;

8506 030210 000000 .WORD 0 ;

8507 030212 012746 006040 MOV #WRD.19,-(SP) ; 4899

8508 030216 012746 007020 MOV #FNC.5,-(SP) ;

8509 030222 012746 005760 MOV #WRD.12,-(SP) ;

8510 030226 012746 006542 MOV #PHR.1,-(SP) ;

8511 030232 012746 005630 MOV #WRD.1,-(SP) ;

8512 030236 012746 005414 MOV #FIV.FMT,-(SP) ;

8513 030242 012746 000006 MOV #6,-(SP) ;

8514 030246 010600 MOV SP,R0 ; SP,*

8515 030250 104414 TRAP 14 ;

8516 030252 132777 000200 161470 BITB #200,@ML.REG+50 ; 4901

8517 030260 001056 BNE 3\$;

22-Oct-1980 10:47:44

22-Oct-1980 10:45:32

TOPS-20 Bliss-16 V2(206)

PA:<NEALE>BL2ML4.BLI.2 (55)

!SEE IF XFERR CAUSED A TRANSFER ERROR

!REPORT ERROR IF SET AND CONTINUE TESTING

!DROP THIS UNIT IF DODU_FLG SET

22-Oct-1980 10:47:44 TOPS
22-Oct-1980 10:45:32 PA:<

```

8519                                     ;ML4
8520                                     ;
8521
8522 030262 104455                      TRAP      55
8523 030264 000031                      .WORD    31
8524 030266 007444                      .WORD    ASYNC
8525 030270 000000                      .WORD    0
8526 030272 012746 006660              MOV      #PHR.6,-(SP)
8527 030276 012746 005630              MOV      #WRD.1,-(SP)
8528 030302 012746 006306              MOV      #WRD.43,-(SP)
8529 030306 012746 006542              MOV      #PHR.1,-(SP)
8530 030312 012746 005634              MOV      #WRD.2,-(SP)
8531 030316 012746 005414              MOV      #FIV.FMT,-(SP)
8532 030322 012746 000006              MOV      #6,-(SP)
8533 030326 010600                      MOV      SP,R0
8534 030330 104414                      TRAP     14
8535 030332 062706 000016              ADD      #16,SP
8536 030336 000427                      BR       3$
8537 030340 105777 161404              2$:     TSTB   @ML.REG+50
8538 030344 100030                      BPL     4$
8539 030346 104455                      TRAP     55
8540 030350 000032                      .WORD    32
8541 030352 007444                      .WORD    ASYNC
8542 030354 000000                      .WORD    0
8543 030356 012746 006646              MOV      #PHR.5,-(SP)
8544 030362 012746 005630              MOV      #WRD.1,-(SP)
8545 030366 012746 006306              MOV      #WRD.43,-(SP)
8546 030372 012746 006560              MOV      #PHR.2,-(SP)
8547 030376 012746 005634              MOV      #WRD.2,-(SP)
8548 030402 012746 005414              MOV      #FIV.FMT,-(SP)
8549 030406 012746 000006              MOV      #6,-(SP)
8550 030412 010600                      MOV      SP,R0
8551 030414 104414                      TRAP     14
8552 030416 012702 000001              3$:     MOV      #1,R2
8553 030422 062706 000016              ADD      #16,SP
8554 030426 132777 000001 161324 4$:  BITB   #1,@ML.REG+60
8555 030434 001430                      BEQ     5$
8556 030436 104455                      TRAP     55
8557 030440 000033                      .WORD    33
8558 030442 007444                      .WORD    ASYNC
8559 030444 000000                      .WORD    0
8560 030446 012746 006040              MOV      #WRD.19,-(SP)
8561 030452 012746 007020              MOV      #FNC.5,-(SP)
8562 030456 012746 005760              MOV      #WRD.12,-(SP)
8563 030462 012746 006646              MOV      #PHR.5,-(SP)
8564 030466 012746 005646              MOV      #WRD.3,-(SP)
8565 030472 012746 005414              MOV      #FIV.FMT,-(SP)
8566 030476 012746 000006              MOV      #6,-(SP)
8567 030502 010600                      MOV      SP,R0
8568 030504 104414                      TRAP     14
8569 030506 012702 000001              5$:     MOV      #1,R2
8570 030512 062706 000016              ADD      #16,SP
8571 030516 032777 020000 161234 5$:  BIT     #20000,@ML.REG+60
8572 030524 001430                      BEQ     6$
8573 030526 104455                      TRAP     55

```

4904

4905

4903

4908

4912

4915

4916

4917

4914

4920

4923

4924

4925

4922

4928

4931

```

8575      ;ML4
8576      :
8577
8578 030530 000034      .WORD 34
8579 030532 007444      .WORD ASYNC
8580 030534 000000      .WORD 0
8581 030536 012746 006040      MOV #WRD.19,-(SP)      ;
8582 030542 012746 007020      MOV #FNC.5,-(SP)
8583 030546 012746 005760      MOV #WRD.12,-(SP)
8584 030552 012746 006646      MOV #PHR.5,-(SP)
8585 030556 012746 005654      MOV #WRD.4,-(SP)
8586 030562 012746 005414      MOV #FIV.FMT,-(SP)
8587 030566 012746 000006      MOV #6,-(SP)
8588 030572 010600      MOV SP,R0      ; SP,*
8589 030574 104414      TRAP 14
8590 030576 012702 000001      MOV #1,R2      ; *,DODU.FLG
8591 030602 062706 000016      ADD #16,SP      ;
8592 030606 012700 000050      MOV #50,R0      ; *,$$TMP2
8593 030612 001410      6$: BEQ 10$
8594 030614 016701 151276      7$: MOV LSDLY,R1      ; *,$$TMP1
8595 030620 001403      BEQ 9$
8596 030622 005016      8$: CLR (SP)      ; $$TMP
8597 030624 005301      DEC R1      ; $$TMP1
8598 030626 001375      BNE 8$
8599 030630 005300      9$: DEC R0      ; $$TMP2
8600 030632 000767      BR 7$
8601 030634 132777 000200 161106 10$: BITB #200,@ML.REG+50      ;
8602 030642 001106      BNE 15$
8603 030644 132777 000001 161026      BITB #1,@ML.REG      ;
8604 030652 001452      BEQ 13$
8605 030654 152777 000040 161056      BISB #40,@ML.REG+40      ;
8606 030662 016701 161302      MOV ML.DUT,R1
8607 030666 042701 177770      BIC #177770,R1
8608 030672 142777 000007 161040      BICB #7,@ML.REG+40
8609 030700 150177 161034      BISB R1,@ML.REG+40
8610 030704 132777 000001 160766      BITB #1,@ML.REG      ;
8611 030712 001405      BEQ 11$
8612 030714 104455      TRAP 55
8613 030716 000035      .WORD 35
8614 030720 007444      .WORD ASYNC
8615 030722 000000      .WORD 0
8616 030724 000404      BR 12$
8617 030726 104455      11$: TRAP 55
8618 030730 000036      .WORD 36
8619 030732 007500      .WORD SYNC
8620 030734 000000      .WORD 0
8621 030736 012746 006040      12$: MOV #WRD.19,-(SP)      ;
8622 030742 012746 007020      MOV #FNC.5,-(SP)
8623 030746 012746 005750      MOV #WRD.11,-(SP)
8624 030752 012746 006560      MOV #PHR.2,-(SP)
8625 030756 012746 005630      MOV #WRD.1,-(SP)
8626 030762 012746 005414      MOV #FIV.FMT,-(SP)
8627 030766 012746 000006      MOV #6,-(SP)
8628 030772 010600      MOV SP,R0      ; SP,*
8629 030774 104414      TRAP 14

```

CZMLAAO ML-11 LOGIC TEST
HARDWARD TEST SECTION

MACRO M1113 23-OCT-80 09:11 PAGE 207

N 15

SEQ 0195

22-Oct-1980 10:47:44 TOPS
22-Oct-1980 10:45:32 PA:<

```

8631          ;ML4
8632          ;
8633
8634 030776 000424      BR      14$           ;
8635 031000 104455      13$:  TRAP    55           ;
8636 031002 000037      .WORD   37           ;
8637 031004 007444      .WORD  ASYNC          ;
8638 031006 000000      .WORD    0           ;
8639 031010 012746 006040  MOV   #WRD.19,-(SP)   ;
8640 031014 012746 007020  MOV   #FNC.5,-(SP)   ;
8641 031020 012746 005750  MOV   #WRD.11,-(SP)  ;
8642 031024 012746 006542  MOV   #PHR.1,-(SP)  ;
8643 031030 012746 005634  MOV   #WRD.2,-(SP)  ;
8644 031034 012746 005414  MOV   #FIV.FMT,-(SP)
8645 031040 012746 000006  MOV   #6,-(SP)
8646 031044 010600      MOV   SP,R0          ; SP,*
8647 031046 104414      TRAP   14
8648 031050 012702 000001  14$:  MOV   #1,R2          ; *,DODU.FLG
8649 031054 062706 000016  ADD   #16,SP         ;
8650 031060 132777 000001 160612 15$:  BITB   #1,@ML.REG    ;
8651 031066 001455      BEQ   18$
8652 031070 152777 000040 160642  BISB   #40,@ML.REG+40 ;
8653 031076 016701 161066  MOV   ML.DUT,R1
8654 031102 042701 177770  BIC   #177770,R1
8655 031106 142777 000007 160624  BICB   #7,@ML.REG+40 ;
8656 031114 150177 160620  BITSB  R1,@ML.REG+40 ;
8657 031120 132777 000001 160552  BITB   #1,@ML.REG    ;
8658 031126 001405      BEQ   16$
8659 031130 104455      TRAP   55
8660 031132 000040      .WORD   40
8661 031134 007444      .WORD  ASYNC
8662 031136 000000      .WORD    0
8663 031140 000404      BR     17$
8664 031142 104455      16$:  TRAP   55
8665 031144 000041      .WORD   41
8666 031146 007500      .WORD  SYNC
8667 031150 000000      .WORD    0
8668 031152 012746 006040  17$:  MOV   #WRD.19,-(SP)   ;
8669 031156 012746 007020  MOV   #FNC.5,-(SP)   ;
8670 031162 012746 005750  MOV   #WRD.11,-(SP)  ;
8671 031166 012746 006560  MOV   #PHR.2,-(SP)  ;
8672 031172 012746 005630  MOV   #WRD.1,-(SP)  ;
8673 031176 012746 005414  MOV   #FIV.FMT,-(SP)
8674 031202 012746 000006  MOV   #6,-(SP)
8675 031206 010600      MOV   SP,R0          ; SP,*
8676 031210 104414      TRAP   14
8677 031212 012702 000001  MOV   #1,R2          ; *,DODU.FLG
8678 031216 062706 000016  ADD   #16,SP         ;
8679 031222 104467      18$:  TRAP   67           ;
8680 031224 006000      ROR   R0
8681 031226 103002      BHIS  19$
8682 031230 000167 176670  JMP   1$
8683 031234 032777 040000 160436 19$:  BIT   #40000,@ML.REG ;
8684 031242 001432      BEQ   20$
8685 031244 104455      TRAP   55           ;

```

```

8687      ;ML4
8688      ;
8689
8690 031246 000163      .WORD 163
8691 031250 007622      .WORD INTER
8692 031252 000000      .WORD 0
8693 031254 012746 006040  MOV #WRD.19,-(SP)      ;
8694 031260 012746 007020  MCV #FNC.5,-(SP)      ;
8695 031264 012746 005760  MOV #WRD.12,-(SP)      ;
8696 031270 012746 006646  MOV #PHR.5,-(SP)      ;
8697 031274 012746 006046  MOV #WRD.20,-(SP)      ;
8698 031300 012746 006526  MOV #WRD.61,-(SP)      ;
8699 031304 012746 005432  MOV #SIX.FMT,-(SP)     ;
8700 031310 012746 000007  MOV #7,-(SP)           ;
8701 031314 010600      MOV SP,R0              ; SP,*
8702 031316 104414      TRAP 14                ;
8703 031320 012702 000001  MOV #1,R2              ; *,DODU.FLG
8704 031324 062706 000020  ADD #20,SP             ;
8705 031330 005302      20$: DEC R2            ; DODU.FLG
8706 031332 001004      BNE 21$                ;
8707 031334 016700 160626  MOV ML.LUN,R0         ;
8708 031340 104451      TRAP 51                ;
8709 031342 104444      TRAP 44                ;
8710 031344 005726      21$: TST (SP)+        ;
8711 031346 000207      RTS PC                 ;
8712
8713      ; Routine Size: 333 words
8714      ; Maximum stack depth per invocation: 18 words
8719
8720
8724
8728 031350      T12::
8729 031350 004767 176542  1$: JSR PC,$T12        ;
8730 031354 104466      TRAP 66                ;
8731 031356 006000      ROR R0                 ;
8732 031360 103773      BLO 1$                 ;
8733 031362 000207      RTS PC                 ;
8734
8735      ; Routine Size: 6 words
8736      ; Maximum stack depth per invocation: 0 words

```

8742 :ML4
8743 :
8744 :
8745 :
8746 :
8747 :
8748 :
8749 :
8750 :
8751 :
8752 :
8753 :
8754 :
8755 :
8756 :
8757 :
8758 :
8759 :
8760 :
8761 :
8762 :
8763 :
8764 :
8765 :
8766 :
8767 :
8768 :
8769 :
8770 :
8771 :
8772 :
8773 :
8774 :
8775 :
8776 :
8777 :
8778 :
8779 :
8780 :
8781 :
8782 :
8783 :
8784 :
8785 :
8786 :
8787 :
8788 :
8789 :
8790 :
8791 :
8792 :
8793 :
8794 :
8795 :
8796 :

4990
4991
4992
4993
4994
4995
4996
4997
4998
4999
5000
5001
5002
5003
5004
5005
5006
5007
5008
5009
5010
5011
5012
5013
5014
5015
5016
5017
5018
5019
5020
5021
5022
5023
5024
5025
5026
5027
5028
5029
5030
5031
5032
5033
5034
5035
5036
5037
5038
5039
5040
5041

BGNTST;

!++

TEST NUMBER: TST 13

TEST NAME: READ FUNCTION TEST

TEST DESCRIPTION:

TEST IF THE DRIVE CAN PERFORM
A READ FUNCTION WITHOUT
HANGING THE DRIVE.

A READ FUNCTION IS WRITTEN
INTO MLCS1

THEN GO AND ERROR BITS ARE
CHECKED FOR CORRECT STATES.

THIS UNIT IS DROPPED ON DETECTED
ERRORS.

IMPLICIT INPUTS: NONE

!--

local

DODU_FLG;

!DROP UNIT FLAG

BGNSUB;

CLR MBUS;

DODU_FLG = ZERO;

FIRST_BLK_XFER ();

MLCS1 = read;

!SET UP A FIRST BLK XFERR

!DO A READ FUNCTION

if .GO IS_NOT_SET

!SEE IF GO GOT SET

then

begin

!ERROR IF CLEAR

ERRDF (34, ASYNC, 0);

PRINTB (FIV_FMT, WRD_1, PHR_1, WRD_12, FNC_6, WRD_19);

if .DRY IS_NOT_SET

!TST DRY SET WITH GO CLEAR

then

begin

!ERROR IF NOT SET

ERRDF (35, ASYNC, 0);

PRINTB (FIV_FMT, WRD_2, PHR_1, WRD_43, WRD_1, PHR_6);

end;

DODU_FLG = ONE;

!SET DODU_FLG

```

8798 :ML4
8799 :
8800 :
8801 : 5042 end
8802 : 5043 else !GO BIT GOT SET
8803 : 5044
8804 : 5045 if .DRY IS_SET !SEE IF DRY IS CLEAR
8805 : 5046 then
8806 : 5047 begin !ERROR IF SET
8807 : 5048 ERRDF (36, ASYNC, 0);
8808 : 5049 PRINTB (FIV_FMT, WRD_2, PHR_2, WRD_43, WRD_1, PHR_5);
8809 : 5050 DODU_FLG = ONE;
8810 : 5051 end;
8811 : 5052
8812 : 5053 if .ILF IS_SET !DID FUNCTION CAUSE ILF
8813 : 5054 then
8814 : 5055 begin !ERROR IF YES
8815 : 5056 ERRDF (37, ASYNC, 0);
8816 : 5057 PRINTB (FIV_FMT, WRD_3, PHR_5, WRD_12, FNC_5, WRD_19);
8817 : 5058 DODU_FLG = ONE;
8818 : 5059 end;
8819 : 5060
8820 : 5061 if .OPI IS_SET !DID FUNCTION CAUSE OPI
8821 : 5062 then
8822 : 5063 begin !ERROR IF YES
8823 : 5064 ERRDF (38, ASYNC, 0);
8824 : 5065 PRINTB (FIV_FMT, WRD_4, PHR_5, WRD_12, FNC_5, WRD_19);
8825 : 5066 DODU_FLG = ONE;
8826 : 5067 end;
8827 : 5068
8828 : 5069 DELAY (FRTY_US); !WAIT FOR XFERR TO COMPLETE
8829 : 5070
8830 : 5071 if .DRY IS_NOT_SET !IS DRY SET AFTER XFERR
8831 : 5072 then
8832 : 5073 begin
8833 : 5074
8834 : 5075 if .GO IS_SET !TEST GO CLEAR WITH DRY NOT SET
8835 : 5076 then
8836 : 5077 begin
8837 : 5078 CLR_MBUS; !CLEAR GO
8838 : 5079
8839 : 5080 if .GO IS_SET then ERRDF (39, ASYNC, 0) else ERRDF (40, SYNC, 0);
8840 : 5081
8841 : 5082 !TST GO TO DETERMINE FAILING MOD
8842 : 5083 PRINTB (FIV_FMT, WRD_1, PHR_2, WRD_11, FNC_5, WRD_19);
8843 : 5084 end
8844 : 5085 else !GO BIT CLEAR WITH DRY NOT SET
8845 : 5086 begin !REPORT ERROR
8846 : 5087 ERRDF (41, ASYNC, 0);
8847 : 5088 PRINTB (FIV_FMT, WRD_2, PHR_1, WRD_11, FNC_5, WRD_19);
8848 : 5089 end;
8849 : 5090
8850 : 5091 DODU_FLG = ONE; !SET DODU_FLG
8851 : 5092 end;
8852 : 5093

```

22-Oct-1980 10:47:44
22-Oct-1980 10:45:32

TOPS-20 Bliss-16 V2(206)
PA:<NEALE>BL2ML4.BLI.2 (55)

8854 :ML4

22-Oct-1980 10:47:44 TOPS-20 Bliss-16 V2(206)
 22-Oct-1980 10:45:32 PA:<NEALE>BL2ML4.BLI.2 (55)

```

8855 :
8856 :
8857 : 5094 if .GO IS_SET          !SEE IF GO CLEAR AFTER XFERR
8858 : 5095 then
8859 : 5096   begin
8860 : 5097   CLR_MBUS;             !CLEAR GO IF STILL SET
8861 : 5098
8862 : 5099   if .GO IS_SET then ERRDF (42, ASYNC, 0) else ERRDF (43, SYNC, 0); !TST GO TO DETERMINE FAILING MOD
8863 : 5100
8864 : 5101   PRINTB (FIV_FMT, WRD_1, PHR_2, WRD_11, FNC_5, WRD_19);
8865 : 5102   DODU_FLG = ONE;
8866 : 5103   end;
8867 : 5104
8868 : 5105 ENDSUB;
8869 : 5106
8870 : 5107 if .TRE IS_SET          !SEE IF XFERR CAUSED A TRANSFER ERROR
8871 : 5108 then
8872 : 5109   begin                !REPORT ERROR IF SET AND CONTINUE TESTING
8873 : 5110   ERRDF (116, INTER, 0);
8874 : 5111   PRINTB (SIX_FMT, WRD_61, WRD_20, PHR_5, WRD_12, FNC_6, WRD_19);
8875 : 5112   DODU_FLG = ONE;
8876 : 5113   end;
8877 : 5114
8878 : 5115 if .DODU_FLG IS_SET      !DROP THIS UNIT IF DODU_FLG SET
8879 : 5116 then
8880 : 5117   begin
8881 : 5118   DODU (.ML_LUN);
8882 : 5119   DOCLN;
8883 : 5120   end;
8884 : 5121
8885 : 5122 ENDTST;
  
```

```

8893 031364 004167 152416 $T13: JSR R1,$SAVE2 ; 4989
8894 031370 005746 TST -(SP) ;
8895 031372 104402 1$: TRAP 2 ; 5020
8896 031374 152777 000040 160336 BISB #40,@ML.REG+40 ; 5022
8897 031402 016701 160562 MOV ML.DUT,R1 ;
8898 031406 042701 177770 BIC #177770,R1 ;
8899 031412 142777 000007 160320 BICB #7,@ML.REG+40 ;
8900 031420 150177 160314 BISB R1,@ML.REG+40 ;
8901 031424 005002 CLR R2 ; DODU.FLG 5024
8902 031426 004767 161066 JSR PC,FIRST.BLK.XFER ; 5025
8903 031432 012777 000071 160240 MOV #71,@ML.REG ; 5026
8904 031440 132777 000001 160232 BITB #1,@ML.REG ; 5028
8905 031446 001057 BNE 2$ ;
8906 031450 104455 TRAP 55 ; 5031
8907 031452 000042 .WORD 42 ;
8908 :ML4
8909 :
8910 :
8911 031454 007444 .WORD ASYNC
8912 031456 000000 .WORD 0
8913 031460 012746 006040 MOV #WRD.19,-(SP) ; 5032
8914 031464 012746 007030 MOV #FNC.6,-(SP)
8915 031470 012746 005760 MOV #WRD.12,-(SP)
8916 031474 012746 006542 MOV #PHR.1,-(SP)
  
```

22-Oct-1980 10:47:44 TOPS
 22-Oct-1980 10:45:32 PA:<

CZMLAAO ML-11 LOGIC TEST
HARDWARD TEST SECTION

SEQ 0200

Line	Address	Hex	Value	Label	Op	Operand	Comments	Seq
8917	031500	012746	005630		MOV	#WRD.1,-(SP)		
8918	031504	012746	005414		MOV	#FIV.FMT,-(SP)		
8919	031510	012746	000006		MOV	#6,-(SP)		
8920	031514	010600			MOV	SP,R0	; SP,*	
8921	031516	104414			TRAP	14		
8922	031520	132777	000200	160222	BITB	#200,@ML.REG+50		5034
8923	031526	001056			BNE	3\$		
8924	031530	104455			TRAP	55		5037
8925	031532	000043			.WORD	43		
8926	031534	007444			.WORD	ASYNC		
8927	031536	000000			.WORD	0		
8928	031540	012746	006660		MOV	#PHR.6,-(SP)		5038
8929	031544	012746	005630		MOV	#WRD.1,-(SP)		
8930	031550	012746	006306		MOV	#WRD.43,-(SP)		
8931	031554	012746	006542		MOV	#PHR.1,-(SP)		
8932	031560	012746	005634		MOV	#WRD.2,-(SP)		
8933	031564	012746	005414		MOV	#FIV.FMT,-(SP)		
8934	031570	012746	000006		MOV	#6,-(SP)		
8935	031574	010600			MOV	SP,R0	; SP,*	
8936	031576	104414			TRAP	14		
8937	031600	062706	000016		ADD	#16,SP		5036
8938	031604	000427			BR	3\$		5041
8939	031606	105777	160136	2\$:	TSTB	@ML.REG+50		5045
8940	031612	100030			BPL	4\$		
8941	031614	104455			TRAP	55		5048
8942	031616	000044			.WORD	44		
8943	031620	007444			.WORD	ASYNC		
8944	031622	000000			.WORD	0		
8945	031624	012746	006646		MOV	#PHR.5,-(SP)		5049
8946	031630	012746	005630		MOV	#WRD.1,-(SP)		
8947	031634	012746	006306		MOV	#WRD.43,-(SP)		
8948	031640	012746	006560		MOV	#PHR.2,-(SP)		
8949	031644	012746	005634		MOV	#WRD.2,-(SP)		
8950	031650	012746	005414		MOV	#FIV.FMT,-(SP)		
8951	031654	012746	000006		MOV	#6,-(SP)		
8952	031660	010600			MOV	SP,R0	; SP,*	
8953	031662	104414			TRAP	14		
8954	031664	012702	000001	3\$:	MOV	#1,R2	; *,DODU.FLG	5050
8955	031670	062706	000016		ADD	#16,SP		5047
8956	031674	132777	000001	160056 4\$:	BITB	#1,@ML.REG+60		5053
8957	031702	001430			BEQ	5\$		
8958	031704	104455			TRAP	55		5056
8959	031706	000045			.WORD	45		
8960	031710	007444			.WORD	ASYNC		
8961	031712	000000			.WORD	0		
8962	031714	012746	006040		MOV	#WRD.19,-(SP)		5057
8963				:ML4				22-Oct-1980 10:47:44 TOPS
8964				:				22-Oct-1980 10:45:32 PA:<
8965								
8966	031720	012746	007020		MOV	#FNC.5,-(SP)		
8967	031724	012746	005760		MOV	#WRD.12,-(SP)		
8968	031730	012746	006646		MOV	#PHR.5,-(SP)		
8969	031734	012746	005646		MOV	#WRD.3,-(SP)		
8970	031740	012746	005414		MOV	#FIV.FMT,-(SP)		
8971	031744	012746	000006		MOV	#6,-(SP)		
8972	031750	010600			MOV	SP,R0	; SP,*	
8973	031752	104414			TRAP	14		

CZMLAAO ML-11 LOGIC TEST
HARDWARD TEST SECTION

MACRO M1113 23-OCT-80 09:11 PAGE 211-2

SEQ 0201
5058
5055
5061
5064
5065

```

8974 031754 012702 000001          MOV    #1,R2                ; *,DODU.FLG
8975 031760 062706 000016          ADD    #16,SP              ;
8976 031764 032777 020000 157766 5$:  BIT    #20000,@ML.REG+60  ;
8977 031772 001430          BEQ    6$                  ;
8978 031774 104455          TRAP   55                  ;
8979 031776 000046          .WORD 46                  ;
8980 032000 007444          .WORD ASYNC               ;
8981 032002 000000          .WORD 0                   ;
8982 032004 012746 006040          MOV    #WRD.19,-(SP)      ;
8983 032010 012746 007020          MOV    #FNC.5,-(SP)      ;
8984 032014 012746 005760          MOV    #WRD.12,-(SP)     ;
8985 032020 012746 006646          MOV    #PHR.5,-(SP)      ;
8986 032024 012746 005654          MOV    #WRD.4,-(SP)      ;
8987 032030 012746 005414          MOV    #FIV.FMT,-(SP)   ;
8988 032034 012746 000006          MOV    #6,-(SP)          ;
8989 032040 010600          MOV    SP,R0              ; SP,*
8990 032042 104414          TRAP   14                  ;
8991 032044 012702 000001          MOV    #1,R2              ; *,DODU.FLG
8992 032050 062706 000016          ADD    #16,SP              ;
8993 032054 012700 000050          6$:  MOV    #50,R0           ; *,$$TMP2
8994 032060 001410          7$:  BEQ    10$              ;
8995 032062 016701 150030          MOV    L$DLY,R1           ; *,$$TMP1
8996 032066 001403          BEQ    9$                  ;
8997 032070 005016          8$:  CLR    (SP)              ; $$TMP
8998 032072 005301          DEC    R1                  ; $$TMP1
8999 032074 001375          BNE    8$                  ;
9000 032076 005300          9$:  DEC    R0                ; $$TMP2
9001 032100 000767          BR     7$                  ;
9002 032102 132777 000200 157640 10$:  BITB   #200,@ML.REG+50   ;
9003 032110 001106          BNE    15$                  ;
9004 032112 132777 000001 157560          BITB   #1,@ML.REG        ;
9005 032120 001452          BEQ    13$                  ;
9006 032122 152777 000040 157610          BISB   #40,@ML.REG+40    ;
9007 032130 016701 160034          MOV    ML.DUT,R1         ;
9008 032134 042701 177770          BIC    #177770,R1        ;
9009 032140 142777 000007 157572          BICB   #7,@ML.REG+40    ;
9010 032146 150177 157566          BISB   R1,@ML.REG+40    ;
9011 032152 132777 000001 157520          BITB   #1,@ML.REG        ;
9012 032160 001405          BEQ    11$                  ;
9013 032162 104455          TRAP   55                  ;
9014 032164 000047          .WORD 47                  ;
9015 032166 007444          .WORD ASYNC               ;
9016 032170 000000          .WORD 0                   ;
9017 032172 000404          BR     12$                 ;
9018                                ;ML4
9019                                ;
9020                                ;
9021 032174 104455          11$:  TRAP   55                  ;
9022 032176 000050          .WORD 50                  ;
9023 032200 007500          .WORD SYNC                ;
9024 032202 000000          .WORD 0                   ;
9025 032204 012746 006040          12$:  MOV    #WRD.19,-(SP)     ;
9026 032210 012746 007020          MOV    #FNC.5,-(SP)      ;
9027 032214 012746 005750          MOV    #WRD.11,-(SP)     ;
9028 032220 012746 006560          MOV    #PHR.2,-(SP)      ;
9029 032224 012746 005630          MOV    #WRD.1,-(SP)      ;
9030 032230 012746 005414          MOV    #FIV.FMT,-(SP)   ;

```

5066
5063
5069
5071
5075
5077
5080

22-Oct-1980 10:47:44 TOPS
22-Oct-1980 10:45:32 PA:<

9031	032234	012746	000006		MOV #6,-(SP)				
9032	032240	010600			MOV SP,R0	; SP,*			
9033	032242	104414			TRAP 14*				
9034	032244	000424			BR 14\$				5075
9035	032246	104455		13\$:	TRAP 55	:			5087
9036	032250	000051			.WORD 51				
9037	032252	007444			.WORD ASYNC				
9038	032254	000000			.WORD 0				
9039	032256	012746	006040		MOV #WRD.19,-(SP)	:			5088
9040	032262	012746	007020		MOV #FNC.5,-(SP)				
9041	032266	012746	005750		MOV #WRD.11,-(SP)				
9042	032272	012746	006542		MOV #PHR.1,-(SP)				
9043	032276	012746	005634		MOV #WRD.2,-(SP)				
9044	032302	012746	005414		MOV #FIV.FMT,-(SP)				
9045	032306	012746	000006		MOV #6,-(SP)				
9046	032312	010600			MOV SP,R0	; SP,*			
9047	032314	104414			TRAP 14				
9048	032316	012702	000001	14\$:	MOV #1,R2	; *,DODU.FLG			5091
9049	032322	062706	000016		ADD #16,SP	:			5073
9050	032326	132777	000001	157344	15\$:	BITB #1,@ML.REG	:		5094
9051	032334	001455			BEQ 18\$:			
9052	032336	152777	000040	157374	BISB #40,@ML.REG+40	:			5096
9053	032344	016701	157620		MOV ML.DUT,R1				
9054	032350	042701	177770		BIC #177770,R1				
9055	032354	142777	000007	157356	BICB #7,@ML.REG+40				
9056	032362	150177	157352		BISB R1,@ML.REG+40				
9057	032366	132777	000001	157304	BITB #1,@ML.REG	:			5099
9058	032374	001405			BEQ 16\$				
9059	032376	104455			TRAP 55				
9060	032400	000052			.WORD 52				
9061	032402	007444			.WORD ASYNC				
9062	032404	000000			.WORD 0				
9063	032406	000404			BR 17\$				
9064	032410	104455		16\$:	TRAP 55				
9065	032412	000053			.WORD 53				
9066	032414	007500			.WORD SYNC				
9067	032416	000000			.WORD 0				
9068	032420	012746	006040	17\$:	MOV #WRD.19,-(SP)	:			5101
9069	032424	012746	007020		MOV #FNC.5,-(SP)				
9070	032430	012746	005750		MOV #WRD.11,-(SP)				
9071	032434	012746	006560		MOV #PHR.2,-(SP)				
9072	032440	012746	005630		MOV #WRD.1,-(SP)				
9073				:ML4				22-Oct-1980 10:47:44	TOPS
9074				:				22-Oct-1980 10:45:32	PA:<
9075				:					
9076	032444	012746	005414		MOV #FIV.FMT,-(SP)				
9077	032450	012746	000006		MOV #6,-(SP)				
9078	032454	010600			MOV SP,R0	; SP,*			
9079	032456	104414			TRAP 14				
9080	032460	012702	000001		MOV #1,R2	; *,DODU.FLG			5102
9081	032464	062706	000016		ADD #16,SP	:			5096
9082	032470	104467		18\$:	TRAP 67	:			5103
9083	032472	006000			ROR R0				
9084	032474	103002			BHIS 19\$				
9085	032476	000167	176670		JMP 1\$				
9086	032502	032777	040000	157170	19\$:	BIT #40000,@ML.REG	:		5107
9087	032510	001432			BEQ 20\$				

9088	032512	104455		TRAP	55				5110
9089	032514	000164		.WORD	164				
9090	032516	007622		.WORD	INTER				
9091	032520	000000		.WORD	0				
9092	032522	012746	006040	MOV	#WRD.19,-(SP)				5111
9093	032526	012746	007030	MOV	#FNC.6,-(SP)				
9094	032532	012746	005760	MOV	#WRD.12,-(SP)				
9095	032536	012746	006646	MOV	#PHR.5,-(SP)				
9096	032542	012746	006046	MOV	#WRD.20,-(SP)				
9097	032546	012746	006526	MOV	#WRD.61,-(SP)				
9098	032552	012746	005432	MOV	#SIX.FMT,-(SP)				
9099	032556	012746	000007	MOV	#7,-(SP)				
9100	032562	010600		MOV	SP,R0			; SP,*	
9101	032564	104414		TRAP	14				
9102	032566	012702	000001	MOV	#1,R2			; *,DODU.FLG	5112
9103	032572	062706	000020	ADD	#20,SP			:	5109
9104	032576	005302		20\$: DEC	R2			; DODU.FLG	5115
9105	032600	001004		BNE	21\$				
9106	032602	016700	157360	MOV	ML.LUN,R0				5118
9107	032606	104451		TRAP	51				
9108	032610	104444		TRAP	44				
9109	032612	005726		21\$: TST	(SP)+				4989
9110	032614	000207		RTS	PC				
9111									
9112									
9113									
9118									
9119									
9123									
9127									
9128									
9129									
9130	032616			T13::					
9131	032616	004767	176542	1\$: JSR	PC,\$T13				5120
9132	032622	104466		TRAP	66				
9133	032624	006000		ROR	R0				
9134	032626	103773		BLO	1\$				
9135	032630	000207		RTS	PC				

: Routine Size: 333 words
 ; Maximum stack depth per invocation: 18 words

:ML4
 :

22-Oct-1980 10:47:44 TOPS
 22-Oct-1980 10:45:32 PA:<

22-Oct-1980 10:47:44 TOPS-20 Bliss-16 V2(206)
22-Oct-1980 10:45:32 PA:<NEALE>BL2ML4.BLI.2 (56)

```

9141 :ML4
9142 :
9143 :
9144 : 5124 !
9145 : 5125 BGNTST;
9146 : 5126
9147 : 5127 !++
9148 : 5128 TEST NUMBER: TST 14
9149 : 5129
9150 : 5130 TEST NAME: CLEAR FUNCTION TEST
9151 : 5131
9152 : 5132 TEST DESCRIPTION:
9153 : 5133 TEST IF THE DRIVE CAN PERFORM A CLEAR FUNCTION WITHOUT HANGING THE DRIVE.
9154 : 5134
9155 : 5135 A CLEAR FUNCTION IS WRITTEN INTO MLCS1.
9156 : 5136
9157 : 5137 THEN GO AND ERROR BITS ARE CHECKED FOR CORRECT STATUS.
9158 : 5138 THIS DRIVE IS DROPPED ON DETECTED ERRORS.
9159 : 5139
9160 : 5140 !--
9161 : 5141
9162 : 5142 Local
9163 : 5143 DODU_FLG; !DROP UNIT FLAG
9164 : 5144
9165 : 5145 BGNSUB;
9166 : 5146 CLR MBUS;
9167 : 5147 DODU_FLG = ZERO;
9168 : 5148 MLER = ONES; !SET BITS IN ERROR REGISTER
9169 : 5149 MLCS1 = DRV CLR; !DO A CLEAR FUNCTION
9170 : 5150 DELAY (ONE_US); !DELAY
9171 : 5151
9172 : 5152 if .GO IS_SET !SEE IF GO CLEARED AFTER FUNCTION
9173 : 5153 then
9174 : 5154 begin !ERROR IF SET
9175 : 5155 ERRDF (44, ASYNC, 0);
9176 : 5156 PRINTB (FIV_FMT, WRD_1, PHR_5, WRD_11, FNC_3, FNC_7, WRD_19);
9177 : 5157
9178 : 5158 if .DRY IS_SET !TST DRY CLEAR WITH GO SET
9179 : 5159 then
9180 : 5160 begin !ERROR IF SET
9181 : 5161 ERRDF (45, ASYNC, 0);
9182 : 5162 PRINTB (FIV_FMT, WRD_2, PHR_5, WRD_43, WRD_1, PHR_5);
9183 : 5163 end;
9184 : 5164
9185 : 5165 DODU_FLG = ONE; !SET DODU_FLG
9186 : 5166 end
9187 : 5167 else !GO CLEARED AFTER FUNCTION
9188 : 5168
9189 : 5169 if .DRY IS_NOT_SET !TST DRY SET WITH GO CLEAR
9190 : 5170 then
9191 : 5171 begin !ERROR IF NOT SET
9192 : 5172 ERRDF (46, ASYNC, 0);
9193 : 5173 PRINTB (FIV_FMT, WRD_2, PHR_1, WRD_43, WRD_1, PHR_6);
9194 : 5174 DODU_FLG = ONE;
9195 : 5175 end;

```

22-Oct-1980 10:47:44 TOPS-20 Bliss-16 V2(206)
22-Oct-1980 10:45:32 PA:<NEALE>BL2ML4.BLI.2 (56)

```

9197 :ML4
9198 :
9199 :
9200 :          5176
9201 :          5177 if .ILF IS_SET          !DID FUNCTION CAUSE ILF
9202 :          5178 then
9203 :              5179 begin                !ERROR IF YES
9204 :              5180 ERRDF (47, ASYNC, 0);
9205 :              5181 PRINTB (FIV_FMT, WRD_3, PHR_5, WRD_12, FNC_3, FNC_7, WRD_19);
9206 :              5182 DODU_FLG = ONE;
9207 :              5183 end;
9208 :          5184
9209 :          5185 if .OPI IS_SET          !DID FUNCTION CAUSE OPI
9210 :          5186 then
9211 :              5187 begin                !ERROR IF YES
9212 :              5188 ERRDF (48, ASYNC, 0);
9213 :              5189 PRINTB (FIV_FMT, WRD_4, PHR_5, WRD_12, FNC_3, FNC_7, WRD_19);
9214 :              5190 DODU_FLG = ONE;
9215 :              5191 end;
9216 :          5192
9217 :          5193 if .MLER neq ZERO        !TEST ERROR REGISTER FOR CLEAR
9218 :          5194 then
9219 :              5195 begin                !ERROR IF NOT CLEAR
9220 :              5196 ERRDF (49, ASYNC, 0);
9221 :              5197 PRINTB (SIX_FMT, FNC_3, FNC_7, WRD_19, WRD_14, WRD_13, REG_3);
9222 :              5198 end;
9223 :          5199
9224 :          5200 ENDSUB;
9225 :          5201
9226 :          5202 if .DODU_FLG IS_SET      !DROP THIS UNIT IF DODU_FLG SET
9227 :          5203 then
9228 :              5204 begin
9229 :              5205 DODU (.ML_LUN);
9230 :              5206 DOCLN;
9231 :              5207 end;
9232 :          5208
9233 :          5209 ENDTST;

```

```

9241 032632 004167 151150          $T14: JSR      R1,$SAVE2          ;          5122
9242 032636 005746                TST      -(SP)              ;
9243 032640 104402                1$:    TRAP     2              ;          5143
9244 032642 152777 000040 157070  BISB     #40,@ML.REG+40      ;          5145
9245 032650 016701 157314                MOV     ML.DUT,R1
9246 032654 042701 177770                BIC     #177770,R1
9247 032660 142777 000007 157052  BICB     #7,@ML.REG+40
9248 032666 150177 157046                BISB     R1,@ML.REG+40
9249 032672 005002                CLR     R2                  ; DODU.FLG          5147
9250 032674 012777 177777 157056  MOV     #-1,@ML.REG+60      ;          5148

```



```

9308          :ML4
9309          :
9310
9311 033142 012746 005630      MOV      #WRD.1,-(SP)
9312 033146 012746 006306      MOV      #WRD.43,-(SP)
9313 033152 012746 006542      MOV      #PHR.1,-(SP)
9314 033156 012746 005634      MOV      #WRD.2,-(SP)
9315 033162 012746 005414      MOV      #FIV.FMT,-(SP)
9316 033166 012746 000006      MOV      #6,-(SP)
9317 033172 010600              MOV      SP,R0
9318 033174 104414              TRAP     14
9319 033176 012702 000001      MOV      #1,R2
9320 033202 062706 000016      ADD      #16,SP
9321 033206 132777 000001 156544 8$: BITB     #1,@ML.REG+60
9322 033214 001432              BEQ      9$
9323 033216 104455              TRAP     55
9324 033220 000057              .WORD   57
9325 033222 007444              .WORD   ASYNC
9326 033224 000000              .WORD   0
9327 033226 012746 006040      MOV      #WRD.19,-(SP)
9328 033232 012746 007036      MOV      #FNC.7,-(SP)
9329 033236 012746 006774      MOV      #FNC.3,-(SP)
9330 033242 012746 005760      MOV      #WRD.12,-(SP)
9331 033246 012746 006646      MOV      #PHR.5,-(SP)
9332 033252 012746 005646      MOV      #WRD.3,-(SP)
9333 033256 012746 005414      MOV      #FIV.FMT,-(SP)
9334 033262 012746 000007      MOV      #7,-(SP)
9335 033266 010600              MOV      SP,R0
9336 033270 104414              TRAP     14
9337 033272 012702 000001      MOV      #1,R2
9338 033276 062706 000020      ADD      #20,SP
9339 033302 032777 020000 156450 9$: BIT      #20000,@ML.REG+60
9340 033310 001432              BEQ      10$
9341 033312 104455              TRAP     55
9342 033314 000060              .WORD   60
9343 033316 007444              .WORD   ASYNC
9344 033320 000000              .WORD   0
9345 033322 012746 006040      MOV      #WRD.19,-(SP)
9346 033326 012746 007036      MOV      #FNC.7,-(SP)
9347 033332 012746 006774      MOV      #FNC.3,-(SP)
9348 033336 012746 005760      MOV      #WRD.12,-(SP)
9349 033342 012746 006646      MOV      #PHR.5,-(SP)
9350 033346 012746 005654      MOV      #WRD.4,-(SP)
9351 033352 012746 005414      MOV      #FIV.FMT,-(SP)
9352 033356 012746 000007      MOV      #7,-(SP)
9353 033362 010600              MOV      SP,R0
9354 033364 104414              TRAP     14
9355 033366 012702 000001      MOV      #1,R2
9356 033372 062706 000020      ADD      #20,SP
9357 033376 005777 156356 10$: TST     @ML.REG+60
9358 033402 001430              BEQ      11$
9359 033404 104455              TRAP     55
9360 033406 000061              .WORD   61
9361 033410 007444              .WORD   ASYNC
9362 033412 000000              .WORD   0

```

5174
5171
5177
5180
5181
5182
5179
5185
5188
5189
5190
5187
5193
5196

22-Oct-1980 10:47:44 TOPS
22-Oct-1980 10:45:32 PA:<

```

9364      ;ML4
9365      ;
9366
9367 033414 012746 007320      MOV      #REG.3,-(SP)
9368 033420 012746 005770      MOV      #WRD.13,-(SP)
9369 033424 012746 005774      MOV      #WRD.14,-(SP)
9370 033430 012746 006040      MOV      #WRD.19,-(SP)
9371 033434 012746 007036      MOV      #FNC.7,-(SP)
9372 033440 012746 006774      MOV      #FNC.3,-(SP)
9373 033444 012746 005432      MOV      #SIX.FMT,-(SP)
9374 033450 012746 000007      MOV      #7,-(SP)
9375 033454 010600      MOV      SP,R0
9376 033456 104414      TRAP     14
9377 033460 062706 000020      ADD      #20,SP
9378 033464 104467      11$:    TRAP     67
9379 033466 006000      ROR      R0
9380 033470 103002      BHIS     12$
9381 033472 000167 177142      JMP      1$
9382 033476 005302      12$:    DEC      R2
9383 033500 001004      BNE      13$
9384 033502 016700 156460      MOV      ML.LUN,R0
9385 033506 104451      TRAP     51
9386 033510 104444      TRAP     44
9387 033512 005726      13$:    TST      (SP)+
9388 033514 000207      RTS      PC

```

; Routine Size: 218 words
; Maximum stack depth per invocation: 19 words

```

9389
9390
9391
9396
9397
9401
9405 033516      T14::
9406 033516 004767 177110      1$:    JSR      PC,$T14
9407 033522 104466      TRAP     66
9408 033524 006000      ROR      R0
9409 033526 103773      BLD      1$
9410 033530 000207      RTS      PC

```

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

```

9411
9412
9413
9418 ;ML4
9419 ;
9420
9421 ;      5210 !<BLF/PAGE>

```

22-Oct-1980 10:47:44 TOPS-20 Bliss-16 V2(206)
22-Oct-1980 10:45:32 PA:<NEALE>BL2ML4.BLI.2 (56)

22-Oct-1980 10:47:44
22-Oct-1980 10:45:32

TOPS-20 Bliss-16 V2(206)
PA:<NEALE>BL2ML4.BLI.2 (57)

```

9423 :ML4
9424 :
9425 :
9426 :      5211 !
9427 :      5212 BGNTST;
9428 :      5213
9429 :      5214 !++
9430 :      5215 TEST NUMBER: TST 15
9431 :      5216
9432 :      5217 TEST NAME: COMPOSIT ERROR TEST
9433 :      5218
9434 :      5219 TEST DESCRIPTION:
9435 :      5220 TEST TO SEE IF SETTING EACH
9436 :      5221 BIT IN THE ERROR REGISTER
9437 :      5222 CAUSES A COMPOSIT ERROR BY:
9438 :      5223
9439 :      5224 WRITING A SHIFTING ONE THROUGH
9440 :      5225 THE ERROR REGISTER (SKIPPING THE
9441 :      5226 READ ONLY BITS) AND TESTING THE
9442 :      5227 COMPOSIT ERROR BIT IN MLDS
9443 :      5228 FOR BEING SET AFTER EACH
9444 :      5229 WRITE.
9445 :      5230
9446 :      5231 !IMPLICIT INPUTS: NONE
9447 :      5232 --
9448 :      5233
9449 :      5234 local
9450 :      5235 DODU_FLG,
9451 :      5236 DAT_PAT,
9452 :      5237 SKIP_MASK;
9453 :      5238
9454 :      5239 CLR MBUS;
9455 :      5240 DODU_FLG = ZERO;
9456 :      5241 SKIP_MASK = %o'163157';
9457 :      5242 DAT_PAT = ONE;
9458 :      5243 DODU_FLG = ZERO;
9459 :      5244
9460 :      5245 incr COUNT from 0 to 15 do
9461 :      5246 begin
9462 :      5247
9463 :      5248 if (.DAT_PAT and .SKIP_MASK) neq ZERO
9464 :      5249 then
9465 :      5250 begin
9466 :      5251 BGNSUB;
9467 :      5252 MLER = .DAT_PAT;
9468 :      5253
9469 :      5254 if .COMP_ERR IS_NOT_SET
9470 :      5255 then
9471 :      5256 begin
9472 :      5257 ERRDF (50, ASYNC, 0); !ERROR IF NO COMP ERROR
9473 :      5258 PRINTB (FOR_FMT, FNC_8, PHR_1, WRD_12, FNC_8);
9474 :      5259 PRINTB (FMT_4, .DAT_PAT);
9475 :      5260 DODU_FLG = ONE;
9476 :      5261 end;
9477 :      5262

```

!DROP UNIT FLAG
!DATA PATTERN
!POINTS TO MLER READ ONLY BITS

!LOAD SKIP MASK
!DATA PATTERN SET BIT 0 IN MLER

!WRITE AND SHIFT DATA PAT TO MLER 16 TIMES

!SKIP IF DAT_PAT FALLS ON READ ONLY BIT

!WRITE DATA_PAT TO MLER

!SEE IF DAT_PAT CAUSED A COMP ERROR

22-Oct-1980 10:47:44 TOPS-20 Bliss-16 V2(206)
22-Oct-1980 10:45:32 PA:<NEALE>BL2ML4.BLI.2 (57)

```

9479 ;ML4
9480 ;
9481
9482 :      5263      ENDSUB;
9483 :      5264      end;
9484 :      5265
9485 :      5266      DAT_PAT = .DAT_PAT^ONE;
9486 :      5267      end;
9487 :      5268
9488 :      5269      if .DODU_FLG IS_SET
9489 :      5270      then
9490 :      5271      begin
9491 :      5272      DODU (.ML_LUN);
9492 :      5273      DOCLN;
9493 :      5274      end;
9494 :      5275
9495 :      5276      ENDTST;
9499
9503 033532 004167 150302 $T15: JSR R1,$SAVE4 ; 5209
9504 033536 152777 000040 156174 BISB #40,@ML.REG+40 ; 5237
9505 033544 016704 156420 MOV ML.DUT,R4
9506 033550 042704 177770 BIC #177770,R4
9507 033554 142777 000007 156156 BICB #7,@ML.REG+40
9508 033562 150477 156152 BISB R4,@ML.REG+40
9509 033566 005001 CLR R1 ; DODU_FLG 5240
9510 033570 012704 163157 MOV #-14621,R4 ; *,SKIP.MASK 5241
9511 033574 012702 000001 MOV #1,R2 ; *,DAT.PAT 5242
9512 033600 005003 CLR R3 ; COUNT 5245
9513 033602 030204 1$: BIT R2,R4 ; DAT.PAT,SKIP.MASK 5248
9514 033604 001447 BEQ 4$
9515 033606 104402 2$: TRAP 2 ; 5250
9516 033610 010277 156144 MOV R2,@ML.REG+60 ; DAT.PAT,* 5252
9517 033614 032777 040000 156126 BIT #40000,@ML.REG+50 ; 5254
9518 033622 001035 BNE 3$
9519 033624 104455 TRAP 55 ; 5257
9520 033626 000062 .WORD 62
9521 033630 007444 .WORD ASYNC
9522 033632 000000 .WORD 0
9523 033634 012746 007046 MOV #FNC.8,-(SP) ; 5258
9524 033640 012746 005760 MOV #WRD.12,-(SP)
9525 033644 012746 006542 MOV #PHR.1,-(SP)
9526 033650 012746 007046 MOV #FNC.8,-(SP)
9527 033654 012746 005400 MOV #FOR.FMT,-(SP)
9528 033660 012746 000005 MOV #5,-(SP)
9529 033664 010600 MOV SP,R0 ; SP,*
9530 033666 104414 TRAP 14
9531 033670 010216 MOV R2,(SP) ; DAT.PAT,* 5259
9532 033672 012746 004336 MOV #FMT.4,-(SP)

```

```
9534 :ML4
9535 :
9536 :
9537 033676 012746 000002      MOV    #2,-(SP)
9538 033702 010600      MOV    SP,R0          ; SP,*
9539 033704 104414      TRAP   14
9540 033706 012701 000001      MOV    #1,R1          ; *,DODU.FLG
9541 033712 062706 000020      ADD    #20,SP
9542 033716 104467      3$:   TRAP   67
9543 033720 006000      ROR    R0
9544 033722 103731      BLO    2$
9545 033724 006302      4$:   ASL    R2          ; DAT.PAT
9546 033726 005203      INC    R3              ; COUNT
9547 033730 020327 000017      CMP    R3,#17         ; COUNT,*
9548 033734 003722      BLE    1$
9549 033736 005301      DEC    R1              ; DODU.FLG
9550 033740 001004      BNE    5$
9551 033742 016700 156220      MOV    ML.LUN,R0
9552 033746 104451      TRAP   51
9553 033750 104444      TRAP   44
9554 033752 000207      5$:   RTS    PC
9555 :
9556 : Routine Size: 73 words
9557 : Maximum stack depth per invocation: 13 words
9562 :
9563 :
9567 :
9571 033754      T15::
9572 033754 004767 177552      1$:   JSR    PC,$T15
9573 033760 104466      TRAP   66
9574 033762 006000      ROR    R0
9575 033764 103773      BLO    1$
9576 033766 000207      RTS    PC
9577 :
9578 : Routine Size: 6 words
9579 : Maximum stack depth per invocation: 0 words
9584 :
9585 :
9586 :      5277 !<BLF/PAGE>
```

22-Oct-1980 10:47:44
22-Oct-1980 10:45:32

TOPS-20 Bliss-16 V2(206)
PA:<NEALE>BL2ML4.BLI.2 (58)

```

9588 ;ML4
9589 :
9590 :
9591 : 5278 !
9592 : 5279 !
9593 : 5230 BGNTST;
9594 : 5281 !
9595 : 5282 !++
9596 : 5283 TEST NUMBER: TST 16
9597 : 5284 !
9598 : 5285 TEST NAME: ATA BIT TEST
9599 : 5286 !
9600 : 5287 TEST DESCRIPTION:
9601 : 5288 TEST THE ATA BIT FOR SETTING
9602 : 5289 BY:
9603 : 5290 1. SETTING A BIT IN THE ERROR
9604 : 5291 REGISTER.
9605 : 5292 TEST THE ATA BIT FOR CLEARING
9606 : 5293 AFTER BEING SET BY:
9607 : 5294 1. WRITING A FUNCTION TO MLCS1.
9608 : 5295 2. WRITING A ONE INTO THIS
9609 : 5296 UNITS ATA BIT
9610 : 5297 TEST THE ATA BIT FOR NOT CLEARING
9611 : 5298 AFTER BEING SET BY:
9612 : 5299 1. WRITING A ONE INTO THE
9613 : 5300 ATA BIT OF THE OTHER
9614 : 5301 UNITS.
9615 : 5302 IMPLICIT INPUTS: NONE
9616 : 5303
9617 : 5304
9618 : 5305
9619 : 5306
9620 : 5307
9621 : 5308
9622 : 5309
9623 : 5310
9624 : 5311 --
9625 : 5312
9626 : 5313 Local
9627 : 5314 ATA_SAVE : bitvector [8], !STORES ALL 8 ATA BITS ON READS AND WRITES
9628 : 5315 DAT_PAT; !DATA PATTERN
9629 : 5316
9630 : 5317 CLR MBUS;
9631 : 5318 MLER = ONE; !SET THE ATA BIT
9632 : 5319 MLER = ZERO;
9633 : 5320 ATA_SAVE = .MLAS; !READ THE ATTN REGISTER
9634 : 5321
9635 : 5322 if .ATA_SAVE [.ML_DUT] IS_NOT_SET !SEE IF THIS DRIVES ATA BIT IS SET
9636 : 5323 then
9637 : 5324 begin
9638 : 5325 ERRDF (51, ASYNC, 0); !ERROR AND EXIT_TST IF NOT SET
9639 : 5326 PRINTB (FOR_FMT, WRD_15, PHR_1, WRD_11, FNC_8);
9640 : 5327 EXIT_TST;
9641 : 5328 end;
9642 : 5329

```

```
9644 :ML4
9645 :
9646 :
9647 : 5330 if .ATTN IS_NOT_SET
9648 : 5331 then
9649 : 5332 begin
9650 : 5333 ERRDF (52, ASYNC, 0);
9651 : 5334 PRINTB (FIV_FMT, REG_2, WRD_16, PHR_1, WRD_11, FNC_8);
9652 : 5335 EXIT_TST;
9653 : 5336 end;
9654 : 5337
9655 : 5338 MLCS1 = NOOP;
9656 : 5339
9657 : 5340 if .ATTN IS_SET
9658 : 5341 then
9659 : 5342 begin
9660 : 5343 ERRDF (53, ASYNC, 0);
9661 : 5344 PRINTB (FOR_FMT, WRD_15, PHR_2, WRD_11, FNC_2, WRD_19);
9662 : 5345 EXIT_TST;
9663 : 5346 end;
9664 : 5347
9665 : 5348 ATA_SAVE = .MLAS;
9666 : 5349
9667 : 5350 if .ATA_SAVE [.ML_DUT] IS_SET
9668 : 5351 then
9669 : 5352 begin
9670 : 5353 ERRDF (58, ASYNC, 0);
9671 : 5354 PRINTB (FIV_FMT, WRD_15, PHR_2, WRD_11, FNC_2, WRD_19);
9672 : 5355 end;
9673 : 5356
9674 : 5357 BGNSUB;
9675 : 5358 MLER = ONE;
9676 : 5359 MLER = ZERO;
9677 : 5360 ATA_SAVE = ZEROES;
9678 : 5361 ATA_SAVE [.ML_DUT] = ONE;
9679 : 5362 MLAS = .ATA_SAVE;
9680 : 5363
9681 : 5364 if .ATTN IS_SET
9682 : 5365 then
9683 : 5366 begin
9684 : 5367 ERRDF (54, ASYNC, 0);
9685 : 5368 PRINTB (FIV_FMT, WRD_15, PHR_2, WRD_11, WRD_17, REG_5);
9686 : 5369 end;
9687 : 5370
9688 : 5371 ENDSUB;
9689 : 5372 BGNSUB;
9690 : 5373 MLER = ONE;
9691 : 5374 MLER = ZERO;
9692 : 5375 DAT_PAT = ONE;
9693 : 5376
9694 : 5377 incr ATA_SEL from 0 to 7 do
9695 : 5378 begin
9696 : 5379
9697 : 5380 if .ATA_SEL neq .ML_DUT
9698 : 5381 then
```

22-Oct-1980 10:47:44 TOPS-20 Bliss-16 V2(206)
22-Oct-1980 10:45:32 PA:<NEALE>BL2ML4.BLI.2 (58)

!SEE IF THE ATTN BIT IS SET

!ERROR AND EXIT_TST IF NOT SET

!TRY TO CLEAR THE ATA BIT WITH NOOP FUNC

!SEE IF ATA GOT CLEARED

!ERROR AND EXIT_TST IF SET

!READ THE ATTENTION REGISTER

!SEE IF THE ATA REG GOT CLEARED BY NO-OP

!SET THE ATA BIT

!CLEAR ATA_SAVE

!SET ATA_SAVE FOR THIS DRIVE

!TRY TO CLEAR THE ATA BY WRITING TO IT.

!SEE IF THE ATA GOT CLEARED

!ERROR IF NOT CLEARED

!SET THE ATA BIT

!DATA PATTERN OF ONE IN FIELD OF ZEROES

!REPEAT LOOP 8 TIMES

!SKIP IF ATA_SEL EQLS THIS DRIVE NO.

22-Oct-.980 10:47:44 TOPS-20 Bliss-16 V2(206)
22-Oct-1980 10:45:32 PA:<NEALE>BL2ML4.BLI.2 (58)

```

9700 ;ML4
9701 ;
9702
9703 :      5382      begin
9704 :      5383      MLAS = .DAT_PAT;           !WRITE DAT PAT TO ATA REGISTER
9705 :      5384      ATA_SAVE = .MLAS;       !READ ATA REG BACK
9706 :      5385
9707 :      5386      if .ATA_SAVE [.ML_DUT] IS_NOT_SET   !SEE IF THIS DRIVE ATA IS CLEARED
9708 :      5387      then
9709 :      5388          begin
9710 :      5389          ERRDF (55, ASYNC, 0);           !ERROR AND EXIT LOOP IF CLEARED
9711 :      5390          PRINTB (SIX_FMT, WRD_15, PHR_6, WRD_11, WRD_17, REG_5, PHR_7);
9712 :      5391          PRINTB (FMT_7, .DAT_PAT);
9713 :      5392          exitloop;
9714 :      5393          end;
9715 :      5394
9716 :      5395      end;
9717 :      5396
9718 :      5397      DAT_PAT = .DAT_PAT^ONE;       !SHIFT DAT_PAT AND REPEAT
9719 :      5398      end;
9720 :      5399
9721 :      5400      ENDSUB;
9722 :      5401      ENDTST;

```

```

9730 033770 004167 150044      $T16:  JSR      R1,$SAVE4           ;           5276
9731 033774 152777 000040 155736  BISB    #40,@ML.REG+40      ;           5315
9732 034002 016700 156162      MOV     ML.DUT,R0
9733 034006 010004      MOV     R0,R4
9734 034010 042704 177770      BIC    #177770,R4
9735 034014 142777 000007 155716  BICB    #7,@ML.REG+40
9736 034022 150477 155712      BISB    R4,@ML.REG+40
9737 034026 012777 000001 155724  MOV     #1,@ML.REG+60      ;           5318
9738 034034 005077 155720      CLR    @ML.REG+60        ;           5319
9739 034040 017746 155724      MOV     @ML.REG+70,-(SP)  ; * ,ATA.SAVE  5320
9740 034044 010001      MOV     R0,R1           ;           5322
9741 034046 006201      ASR    R1
9742 034050 006201      ASR    R1
9743 034052 006201      ASR    R1
9744 034054 010602      MOV     SP,R2           ; ATA.SAVE,*
9745 034056 060201      ADD    R2,R1
9746 034060 010146      MOV     R1,-(SP)
9747 034062 010046      MOV     R0,-(SP)
9748 034064 042716 177770      BIC    #177770,(SP)
9749 034070 012746 000001      MOV     #1,-(SP)
9750 034074 005046      CLR    -(SP)
9751 034076 004767 147000      JSR    PC,BL$GT2
9752 034102 062706 000010      ADD    #10,SP
9753 034106 005700      TST    R0

```


22-Oct-1980 10:47:44 TOPS
 22-Oct-1980 10:45:32 PA:<

Address	OpCode	Operand 1	Operand 2	Operand 3	Operand 4	Comment	Line No.
9755							
9756							
9757							
9758	034110	001026					
9759	034112	104455					5325
9760	034114	000063					
9761	034116	007444					
9762	034120	000000					
9763	034122	012746	007046				5326
9764	034126	012746	005750				
9765	034132	012746	006542				
9766	034136	012746	006006				
9767	034142	012746	005400				
9768	034146	012746	000005				
9769	034152	010600					
9770	034154	104414					
9771	034156	104463					
9772	034160	062706	000014				5322
9773	034164	000467					5324
9774	034166	032777	100000	155554	1\$:		5330
9775	034174	001026					
9776	034176	104455					5333
9777	034200	000064					
9778	034202	007444					
9779	034204	000000					
9780	034206	012746	007046				5334
9781	034212	012746	005750				
9782	034216	012746	006542				
9783	034222	012746	006014				
9784	034226	012746	007312				
9785	034232	012746	005414				
9786	034236	012746	000006				
9787	034242	010600					
9788	034244	104414					
9789	034246	104463					
9790	034250	000433					5330
9791	034252	012777	000001	155420	2\$:		5338
9792	034260	005777	155464				5340
9793	034264	100031					
9794	034266	104455					5343
9795	034270	000065					
9796	034272	007444					
9797	034274	000000					
9798	034276	012746	006040				5344
9799	034302	012746	006766				
9800	034306	012746	005750				
9801	034312	012746	006560				
9802	034316	012746	006006				
9803	034322	012746	005400				
9804	034326	012746	000006				
9805	034332	010600					
9806	034334	104414					
9807	034336	104463					
9808	034340	062706	000016		3\$:		5340
9809	034344	000167	000544		4\$:		5342

Address	Hex	Hex	Hex	Label	Instruction	Comments	Line No.
9811				;ML4			
9812				:			
9813							
9814	034350	017716	155414	5\$:	MOV @ML.REG+70,(SP)	; *,ATA.SAVE	5348
9815	034354	016701	155610		MOV ML.DUT,R1	:	5350
9816	034350	006201			ASR R1	:	
9817	034362	006201			ASR R1	:	
9818	034364	006201			ASR R1	:	
9819	034366	010602			MOV SP,R2	; ATA.SAVE,*	
9820	034370	060201			ADD R2,R1	:	
9821	034372	010146			MOV R1,-(SP)	:	
9822	034374	016746	155570		MOV ML.DUT,-(SP)	:	
9823	034400	042716	177770		BIC #177770,(SP)	:	
9824	034404	012746	000001		MOV #1,-(SP)	:	
9825	034410	005046			CLR -(SP)	:	
9826	034412	004767	146464		JSR PC,BL\$GT2	:	
9827	034416	062706	000010		ADD #10,SP	:	
9828	034422	005300			DEC R0	:	
9829	034424	001026			BNE 6\$:	
9830	034426	104455			TRAP 55	:	5353
9831	034430	000072			.WORD 72	:	
9832	034432	007444			.WORD ASYNC	:	
9833	034434	000000			.WORD 0	:	
9834	034436	012746	006040		MOV #WRD.19,-(SP)	:	5354
9835	034442	012746	006766		MOV #FNC.2,-(SP)	:	
9836	034446	012746	005750		MOV #WRD.11,-(SP)	:	
9837	034452	012746	006560		MOV #PHR.2,-(SP)	:	
9838	034456	012746	006006		MOV #WRD.15,-(SP)	:	
9839	034462	012746	005414		MOV #FIV.FMT,-(SP)	:	
9840	034466	012746	000006		MOV #6,-(SP)	:	
9841	034472	010600			MOV SP,R0	; SP,*	
9842	034474	104414			TRAP 14	:	
9843	034476	062706	000016		ADD #16,SP	:	5352
9844	034502	104402			TRAP 2	:	5355
9845	034504	012777	000001 155246	6\$:	MOV #1,@ML.REG+60	:	5358
9846	034512	005077	155242		CLR @ML.REG+60	:	5359
9847	034516	005016			CLR (SP)	; ATA.SAVE	5360
9848	034520	016701	155444		MOV ML.DUT,R1	:	5361
9849	034524	006201			ASR R1	:	
9850	034526	006201			ASR R1	:	
9851	034530	006201			ASR R1	:	
9852	034532	010602			MOV SP,R2	; ATA.SAVE,*	
9853	034534	060201			ADD R2,R1	:	
9854	034536	010146			MOV R1,-(SP)	:	
9855	034540	016746	155424		MOV ML.DUT,-(SP)	:	
9856	034544	042716	177770		BIC #177770,(SP)	:	
9857	034550	012746	000001		MOV #1,-(SP)	:	
9858	034554	011646			MOV (SP),-(SP)	:	
9859	034556	004767	146556		JSR PC,BL\$PU2	:	
9860	034562	016677	000010 155200		MOV 10(SP),@ML.REG+70	; ATA.SAVE,*	5362
9861	034570	005777	155154		TST @ML.REG+50	:	5364
9862	034574	100026			BPL 7\$:	
9863	034576	104455			TRAP 55	:	5367
9864	034600	000066			.WORD 66	:	
9865	034602	007444			.WORD ASYNC	:	

22-Oct-1980 10:47:44 TOPS
 22-Oct-1980 10:45:32 PA:<

Address	OpCode	Operand 1	Operand 2	Label	Comment	Seq
9867				:ML4		
9868				:		
9869						
9870	034604	000000			.WORD 0	
9871	034606	012746	007334		MOV #REG.5,-(SP)	5368
9872	034612	012746	006022		MOV #WRD.17,-(SP)	
9873	034616	012746	005750		MOV #WRD.11,-(SP)	
9874	034622	012746	006560		MOV #PHR.2,-(SP)	
9875	034626	012746	006006		MOV #WRD.15,-(SP)	
9876	034632	012746	005414		MOV #FIV.FMT,-(SP)	
9877	034636	012746	000006		MOV #6,-(SP)	
9878	034642	010600			MOV SP,R0	: SP,*
9879	034644	104414			TRAP 14	
9880	034646	062706	000016		ADD #16,SP	5366
9881	034652	062706	000010	7\$:	ADD #10,SP	5355
9882	034656	104467			TRAP 67	5369
9883	034660	006000			ROR R0	
9884	034662	103707			BLO 6\$	
9885	034664	104402		8\$:	TRAP 2	5371
9886	034666	012777	000001	155064	MOV #1,@ML.REG+60	5373
9887	034674	005077	155060		CLR @ML.REG+60	5374
9888	034700	012704	000001		MOV #1,R4	: *,DAT.PAT
9889	034704	005003			CLR R3	: ATA.SEL
9890	034706	016702	155256	9\$:	MOV ML.DUT,R2	: ATA.SEL,*
9891	034712	020302			CMP R3,R2	
9892	034714	001467			BEQ 10\$	
9893	034716	010477	155046		MOV R4,@ML.REG+70	: DAT.PAT,*
9894	034722	017716	155042		MOV @ML.REG+70,(SP)	: *,ATA.SAVE
9895	034726	010201			MOV R2,R1	
9896	034730	006201			ASR R1	
9897	034732	006201			ASR R1	
9898	034734	006201			ASR R1	
9899	034736	010600			MOV SP,R0	: ATA.SAVE,*
9900	034740	060001			ADD R0,R1	
9901	034742	010146			MOV R1,-(SP)	
9902	034744	010246			MOV R2,-(SP)	
9903	034746	042716	177770		BIC #177770,(SP)	
9904	034752	012746	000001		MOV #1,-(SP)	
9905	034756	005046			CLR -(SP)	
9906	034760	004767	146116		JSR PC,BL\$GT2	
9907	034764	062706	000010		ADD #10,SP	
9908	034770	005700			TST R0	
9909	034772	001040			BNE 10\$	
9910	034774	104455			TRAP 55	5389
9911	034776	000067			.WORD 67	
9912	035000	007444			.WORD ASYNC	
9913	035002	000000			.WORD 0	
9914	035004	012746	006672		MOV #PHR.7,-(SP)	5390
9915	035010	012746	007334		MOV #REG.5,-(SP)	
9916	035014	012746	006022		MOV #WRD.17,-(SP)	
9917	035020	012746	005750		MOV #WRD.11,-(SP)	
9918	035024	012746	006660		MOV #PHR.6,-(SP)	
9919	035030	012746	006006		MOV #WRD.15,-(SP)	
9920	035034	012746	005432		MOV #SIX.FMT,-(SP)	
9921	035040	012746	000007		MOV #7,-(SP)	

22-Oct-1980 10:47:44 TOPS
22-Oct-1980 10:45:32 PA:<

```
9923      ;ML4
9924      ;
9925
9926 035044 010600      MOV      SP,R0      ; SP,*
9927 035046 104414      TRAP     14
9928 035050 010416      MOV      R4,(SP)    ; DAT.PAT,*
9929 035052 012746 004520  MOV      #FMT.7,-(SP)
9930 035056 012746 000002  MOV      #2,-(SP)
9931 035062 010600      MOV      SP,R0      ; SP,*
9932 035064 104414      TRAP     14
9933 035066 062706 000024  ADD      #24,SP
9934 035072 000405      BR       11$
9935 035074 006304      10$:    ASL      R4      ; DAT.PAT
9936 035076 005203      INC      R3      ; ATA.SEL
9937 035100 020327 000007  CMP      R3,#7     ; ATA.SEL,*
9938 035104 003700      BLE     9$
9939 035106 104467      11$:    TRAP     67
9940 035110 006000      ROR      R0
9941 035112 103664      BLO     8$
9942 035114 005726      12$:    TST      (SP)+
9943 035116 000207      RTS      PC
9944
9945      ; Routine Size: 300 words
9946      ; Maximum stack depth per invocation: 17 words
9951
9952
9956
9960 035120      T16::
9961 035120 004767 176644  1$:    JSR      PC,$T16
9962 035124 104466      TRAP     66
9963 035126 006000      ROR      R0
9964 035130 103773      BLO     1$
9965 035132 000207      RTS      PC
9966
9967      ; Routine Size: 6 words
9968      ; Maximum stack depth per invocation: 0 words
9973
9974
9975 ;          5402 !<BLF/PAGE>
```

22-Oct-1980 10:47:44
22-Oct-1980 10:45:32

TOPS-20 Bliss-16 V2(206)
PA:<NEALE>BL2ML4.BLI.2 (59)

9977 :ML4
9978 :
9979 :
9980 :
9981 :
9982 :
9983 :
9984 :
9985 :
9986 :
9987 :
9988 :
9989 :
9990 :
9991 :
9992 :
9993 :
9994 :
9995 :
9996 :
9997 :
9998 :
9999 :
10000 :
10001 :
10002 :
10003 :
10004 :
10005 :
10006 :
10007 :
10008 :
10009 :
10010 :
10011 :
10012 :
10013 :
10014 :
10015 :
10016 :
10017 :
10018 :
10019 :
10020 :
10021 :
10022 :
10023 :
10024 :
10025 :
10026 :
10027 :
10028 :
10029 :
10030 :
10031 :
10032 :ML4
10033 :

5403
5404
5405
5406
5407
5408
5409
5410
5411
5412
5413
5414
5415
5416
5417
5418
5419
5420
5421
5422
5423
5424
5425
5426
5427
5428
5429
5430
5431
5432
5433
5434
5435
5436
5437
5438
5439
5440
5441
5442
5443
5444
5445
5446
5447
5448
5449
5450
5451
5452
5453
5454

! BGNTST;

!++

TEST NUMBER: TST 17

TEST NAME: SEARCH FUNCTION TEST

TEST DESCRIPTION:

TEST THE SEARCH FUNCTION BY:

1. DOING A SEARCH FUNCTION AT ARRAY ZERO AND TEST GO, ERROR BITS AND ATTN FOR SETTING/NOT SETTING.
2. DOING SEARCH FUNCTIONS AT ALL PRESENT ARRAYS' AND TEST ATTN SET
3. DOING SEARCH FUNCTIONS AT ALL NOT PRESENT ARRAYS' AND TEST ATTN CLEARED.

IMPLICIT INPUTS: NONE

!--

CLR MBUS;

MLDA = ZEROES;

MLCS1 = SEARCH;

!DO A SEARCH FUNCTION

if .GO IS_SET

!SEE IF GO IS SET

then

begin

ERRDF (56, ASYNC, 0);

!ERROR IF NOT SET

PRINTB (FIV_FMT, WRD_1, PHR_2, WRD_11, FNC_10, WRD_19);

end;

if .ILF IS_SET

!SEE IF ILF IS SET

then

begin

ERRDF (57, ASYNC, 0);

!ERROR IF SET

PRINTB (FIV_FMT, WRD_3, PHR_5, WRD_11, FNC_11, FNC_10, WRD_19);

end;

if .ATTN IS_NOT_SET

!SEE IF ATTN IS SET

then

begin

22-Oct-1980 10:47:44

22-Oct-1980 10:45:32

TOPS-20 Bliss-16 V2(206)

PA:<NEALE>BL2ML4.BLI.2 (59)

```

10034 :
10035 : 5455 ERRDF (59, ASYNC, 0); !ERROR IF NOT SET
10036 : 5456 PRINTB (FIV_FMT, WRD_16, PHR_1, WRD_12, FNC_10, WRD_19);
10037 : 5457 end
10038 : 5458 else
10039 : 5459 begin
10040 : 5460
10041 : 5461 incr ARR_SEL from 0 to .LST_ARR by .ARR_INC do !DO SEARCH AT ALL PRESENT ARRAYS
10042 : 5462 begin
10043 : 5463 BGNSUB;
10044 : 5464 CLR MBUS;
10045 : 5465 MLDA = .ARR_SEL; !LOAD DSA REG WITH ARR_SEL
10046 : 5466 MLCS1 = SEARCH; !DO A SEARCH FUNCTION
10047 : 5467
10048 : 5468 if .OPI IS_SET !READ ATTN
10049 : 5469 then
10050 : 5470 begin
10051 : 5471 ERRDF (60, ASYNC, 0); !ERROR IF NOT SET
10052 : 5472 PRINTB (FIV_FMT, WRD_4, PHR_5, WRD_12, FNC_10, WRD_19);
10053 : 5473 PRINTB (FMT_9, .ARR_SEL);
10054 : 5474 end;
10055 : 5475
10056 : 5476 ENDSUB;
10057 : 5477 end;
10058 : 5478
10059 : 5479 if .OP_NUM_ARR lss %o'00C017' !SEE IF LSS 17 ARRAYS ARE PRESENT
10060 : 5480 then
10061 : 5481
10062 : 5482 incr ARR_SEL from .LST_ARR + .ARR_INC to .ARR_16 by .ARR_INC do
10063 : 5483 !DO A SEARCH AT ALL NOT PRESENT
10064 : 5484 !ARRAYS IF LSS 17
10065 : 5485 begin
10066 : 5486 BGNSUB;
10067 : 5487 CLR MBUS;
10068 : 5488 MLDA = .ARR_SEL; !LOAD DSA REG WITH ARR_SEL
10069 : 5489 MLCS1 = SEARCH; !DO A SEARCH FUNCTION
10070 : 5490
10071 : 5491 if .OPI IS_NOT_SET !SEE IF OPI IS SET
10072 : 5492 then
10073 : 5493 begin
10074 : 5494 ERRDF (61, ASYNC, 0); !ERROR IF NOT SET
10075 : 5495 PRINTB (FIV_FMT, WRD_4, PHR_1, WRD_12, FNC_10, WRD_19);
10076 : 5496 PRINTB (FMT_9, .ARR_SEL);
10077 : 5497 end;
10078 : 5498 ENDSUB;
10079 : 5499 end;
10080 : 5500
10081 : 5501 end;
10082 : 5502
10083 : 5503 ENDTST;

```

10087										22-Oct-1980 10:47:44	TOPS
10088										22-Oct-1980 10:45:32	PA:<
10089											
10090											
10094	035134	004167	146662		\$T17:	JSR	R1,\$SAVE3				5401
10095	035140	152777	000040	154572		BISB	#40,@ML.REG+40				5405
10096	035146	016703	155016			MOV	ML.DUT,R3				

ADDRESS	TEST SECTION	DATA	MACRO	INSTR	OPERANDS	SEQ
10097	035152	042703	177770	BIC	#177770,R3	
10098	035156	142777	000007	BICB	#7,@ML.REG+40	
10099	035164	150377	154550	BISB	R3,@ML.REG+40	
10100	035170	005077	154534	CLR	@ML.REG+30	5435
10101	035174	012777	000031	MOV	#31,@ML.REG	5436
10102	035202	132777	000001	BITB	#1,@ML.REG	5438
10103	035210	001426		BEQ	1\$	
10104	035212	104455		TRAP	55	5441
10105	035214	000070		.WORD	70	
10106	035216	007444		.WORD	ASYN	
10107	035220	000000		.WORD	0	
10108	035222	012746	006040	MOV	#WRD.19,-(SP)	5442
10109	035226	012746	007074	MOV	#FNC.10,-(SP)	
10110	035232	012746	005750	MOV	#WRD.11,-(SP)	
10111	035236	012746	006560	MOV	#PHR.2,-(SP)	
10112	035242	012746	005630	MOV	#WRD.1,-(SP)	
10113	035246	012746	005414	MOV	#FIV.FMT,-(SP)	
10114	035252	012746	000006	MOV	#6,-(SP)	
10115	035256	010600		MOV	SP,R0	: SP,*
10116	035260	104414		TRAP	14	
10117	035262	062706	000016	ADD	#16,SP	5440
10118	035266	132777	000001	BITB	#1,@ML.REG+60	5445
10119	035274	001430		BEQ	2\$	
10120	035276	104455		TRAP	55	5448
10121	035300	000071		.WORD	71	
10122	035302	007444		.WORD	ASYN	
10123	035304	000000		.WORD	0	
10124	035306	012746	006040	MOV	#WRD.19,-(SP)	5449
10125	035312	012746	007074	MOV	#FNC.10,-(SP)	
10126	035316	012746	007104	MOV	#FNC.11,-(SP)	
10127	035322	012746	005750	MOV	#WRD.11,-(SP)	
10128	035326	012746	006646	MOV	#PHR.5,-(SP)	
10129	035332	012746	005646	MOV	#WRD.3,-(SP)	
10130	035336	012746	005414	MOV	#FIV.FMT,-(SP)	
10131	035342	012746	000007	MOV	#7,-(SP)	
10132	035346	010600		MOV	SP,R0	: SP,*
10133	035350	104414		TRAP	14	
10134	035352	062706	000020	ADD	#20,SP	5447
10135	035356	032777	100000	BIT	#100000,@ML.REG+50	5452
10136	035364	001027		BNE	3\$	
10137	035366	104455		TRAP	55	5455
10138	035370	000073		.WORD	73	
10139	035372	007444		.WORD	ASYN	
10140	035374	000000		.WORD	0	
10141						22-Oct-1980 10:47:44 TOPS
10142						22-Oct-1980 10:45:32 PA:<
10143						
10144	035376	012746	006040	MOV	#WRD.19,-(SP)	5456
10145	035402	012746	007074	MOV	#FNC.10,-(SP)	
10146	035406	012746	005760	MOV	#WRD.12,-(SP)	
10147	035412	012746	006542	MOV	#PHR.1,-(SP)	
10148	035416	012746	006014	MOV	#WRD.16,-(SP)	
10149	035422	012746	005414	MOV	#FIV.FMT,-(SP)	
10150	035426	012746	000006	MOV	#6,-(SP)	
10151	035432	010600		MOV	SP,R0	: SP,*
10152	035434	104414		TRAP	14	
10153	035436	062706	000016	ADD	#16,SP	5454

SEQ 0221

:ML4

Address	OpCode	Operand 1	Operand 2	Label	Comment	Seq
10154	035442	000207			RTS PC	5452
10155	035444	016702	152670	3\$:	MOV LST.ARR,P2	5461
10156	035450	016703	152650		MOV ARR.INC,R3	
10157	035454	005001			CLR R1	: ARR.SEL
10158	035456	000467			BR 6\$	
10159	035460	104402		4\$:	TRAP 2	5462
10160	035462	152777	000040 154250		BISB #40,@ML.REG+40	5463
10161	035470	016700	154474		MOV ML.DUT,R0	
10162	035474	042700	177770		BIC #177770,R0	
10163	035500	142777	000007 154232		BICB #7,@ML.REG+40	
10164	035506	150077	154226		BISB R0,@ML.REG+40	
10165	035512	010177	154212		MOV R1,@ML.REG+30	: ARR.SEL,*
10166	035516	012777	000031 154154		MOV #31,@ML.REG	5465
10167	035524	032777	020000 154226		BIT #20000,@ML.REG+60	5466
10168	035532	001435			BEQ 5\$	5468
10169	035534	104455			TRAP 55	5471
10170	035536	000074			.WORD 74	
10171	035540	007444			.WORD ASYNC	
10172	035542	000000			.WORD 0	
10173	035544	012746	006040		MOV #WRD.19,-(SP)	5472
10174	035550	012746	007074		MOV #FNC.10,-(SP)	
10175	035554	012746	005760		MOV #WRD.12,-(SP)	
10176	035560	012746	006646		MOV #PHR.5,-(SP)	
10177	035564	012746	005654		MOV #WRD.4,-(SP)	
10178	035570	012746	005414		MOV #FIV.FMT,-(SP)	
10179	035574	012746	000006		MOV #6,-(SP)	
10180	035600	010600			MOV SP,R0	: SP,*
10181	035602	104414			TRAP 14	
10182	035604	010116			MOV R1,(SP)	: ARR.SEL,*
10183	035606	012746	004602		MOV #FMT.9,-(SP)	5473
10184	035612	012746	000002		MOV #2,-(SP)	
10185	035616	010600			MOV SP,R0	: SP,*
10186	035620	104414			TRAP 14	
10187	035622	062706	000022		ADD #22,SP	5470
10188	035626	104467		5\$:	TRAP 67	5474
10189	035630	006000			ROR R0	
10190	035632	103712			BLO 4\$	
10191	035634	060301			ADD R3,R1	: *,ARR.SEL
10192	035636	020102		6\$:	CMP R1,R2	: ARR.SEL,*
10193	035640	003707			BLE 4\$	
10194	035642	026727	152454 000017		CMP OP.NUM.ARR,#17	5479
10195	035650	002102			BGE 10\$	
10196				:ML4		22-Oct-1980 10:47:44 TOPS
10197				:		22-Oct-1980 10:45:32 PA:<
10198						
10199	035652	016701	152462		MOV LST.ARR,R1	5482
10200	035656	066701	152442		ADD ARR.INC,R1	
10201	035662	016703	152450		MOV ARR.16,R3	
10202	035666	016702	152432		MOV ARR.INC,R2	
10203	035672	000467			BR 9\$	
10204	035674	104402		7\$:	TRAP 2	5484
10205	035676	152777	000040 154034		BISB #40,@ML.REG+40	5485
10206	035704	016700	154260		MOV ML.DUT,R0	
10207	035710	042700	177770		BIC #177770,R0	
10208	035714	142777	000007 154016		BICB #7,@ML.REG+40	
10209	035722	150077	154012		BISB R0,@ML.REG+40	
10210	035726	010177	153776		MOV R1,@ML.REG+30	: ARR.SEL,*

10211	035732	012777	000031	153740	MOV	#31,@ML.REG	:	5488
10212	035740	032777	020000	154012	BIT	#20000,@ML.REG+60	:	5490
10213	035746	001035			BNE	8\$:	
10214	035750	104455			TRAP	55	:	5493
10215	035752	000075			.WORD	75	:	
10216	035754	007444			.WORD	ASYN	:	
10217	035756	000000			.WORD	0	:	
10218	035760	012746	006040		MOV	#WRD.19,-(SP)	:	5494
10219	035764	012746	007074		MOV	#FNC.10,-(SP)	:	
10220	035770	012746	005760		MOV	#WRD.12,-(SP)	:	
10221	035774	012746	006542		MOV	#PHR.1,-(SP)	:	
10222	036000	012746	005654		MOV	#WRD.4,-(SP)	:	
10223	036004	012746	005414		MOV	#FIV.FMT,-(SP)	:	
10224	036010	012746	000006		MOV	#6,-(SP)	:	
10225	036014	010600			MOV	SP,R0	: SP,*	
10226	036016	104414			TRAP	14	:	
10227	036020	010116			MOV	R1,(SP)	: ARR.SEL,*	5495
10228	036022	012746	004602		MOV	#FMT.9,-(SP)	:	
10229	036026	012746	000002		MOV	#2,-(SP)	:	
10230	036032	010600			MOV	SP,R0	: SP,*	
10231	036034	104414			TRAP	14	:	
10232	036036	062706	000022		ADD	#22,SP	:	5492
10233	036042	104467			8\$: TRAP	67	:	5496
10234	036044	006000			ROR	R0	:	
10235	036046	103712			BLO	7\$:	
10236	036050	060201			ADD	R2,R1	: *,ARR.SEL	5482
10237	036052	020103			9\$: CMP	R1,R3	: ARR.SEL,*	
10238	036054	003707			BLE	7\$:	
10239	036056	000207			10\$: RTS	PC	:	5401
10240								
10241								
10242								
10247								
10248								
10255								
10259	036060				T17::			
10260	036060	004767	177050		1\$: JSR	PC,\$T17	:	5501
10261	036064	104466			TRAP	66	:	
10262	036066	006000			ROR	R0	:	
10263	036070	103773			BLO	1\$:	
10264	036072	000207			RTS	PC	:	

: Routine Size: 234 words
 : Maximum stack depth per invocation: 13 words

22-Oct-1980 10:47:44
 22-Oct-1980 10:45:32

TOPS-20 Bliss-16 V2(206)
 PA:<NEALE>BL2ML4.BLI.2 (60)

```

10270 :ML4
10271 ;
10272 :
10273 : 5505 !
10274 : 5506 BGNTST;
10275 : 5507
10276 : 5508 !++
10277 : 5509 TEST NUMBER: TST 18
10278 : 5510
10279 : 5511 TEST NAME: READ IN PRESET TEST
10280 : 5512
10281 : 5513 TEST DESCRIPTION:
10282 : 5514 TEST THE READ IN PRESET FUNCTION BY:
10283 : 5515
10284 : 5516 1. PERFORMING A READ-IN-PRESET FUNCTION AND TESTING GO, ERROR BITS
10285 : 5517 AND VV FOR SET / NOT SET.
10286 : 5518
10287 : 5519 !--
10288 : 5520
10289 : 5521 CLR MBUS;
10290 : 5522 MLC51 = RD_IN_PRE; !DO A READ IN PRESET FUNCTION
10291 : 5523
10292 : 5524 if .GO IS_SET !SEE IF GO IS NOT SET
10293 : 5525 then
10294 : 5526 begin
10295 : 5527 ERRDF (62, ASYNC, 0); !ERROR IF SET
10296 : 5528 PRINTB (FIV_FMT, WRD_1, PHR_2, WRD_11, FNC_11, WRD_19);
10297 : 5529 end;
10298 : 5530
10299 : 5531 if .ILF IS_SET !SEE IF ILF IS NOT SET
10300 : 5532 then
10301 : 5533 begin
10302 : 5534 ERRDF (63, ASYNC, 0); !ERROR IF SET
10303 : 5535 PRINTB (FIV_FMT, WRD_3, PHR_5, WRD_11, FNC_11, WRD_19);
10304 : 5536 end;
10305 : 5537
10306 : 5538 if .OPI IS_SET !SEE IF OPI IS NOT SET
10307 : 5539 then
10308 : 5540 begin
10309 : 5541 ERRDF (64, ASYNC, 0); !ERROR IF SET
10310 : 5542 PRINTB (FIV_FMT, WRD_4, PHR_5, WRD_11, FNC_11, WRD_19);
10311 : 5543 end;
10312 : 5544
10313 : 5545 if .VV IS_NOT_SET !SEE IF VV IS SET
10314 : 5546 then
10315 : 5547 begin
10316 : 5548 ERRDF (65, ASYNC, 0); !ERROR IF NOT SET
10317 : 5549 PRINTB (FIV_FMT, WRD_18, PHR_1, WRD_11, FNC_11, WRD_19);
10318 : 5550 end;
10319 : 5551
10320 : 5552 ENDTST;
10324 :

```

```

10326                ;ML4
10327                ;
10328
10332 036074 152777 000040 153636 $T18:  BISB  #40,@ML.REG+40  ;
10333 036102 016700 154062             MOV  ML.DUT,RO
10334 036106 042700 177770             BIC  #177770,RO
10335 036112 142777 000007 153620  BISB  #7,@ML.REG+40
10336 036120 150077 153614             MOV  RO,@ML.REG+40
10337 036124 012777 000021 153546  MOV  #21,@ML.REG  ;
10338 036132 132777 000001 153540  BITB  #1,@ML.REG  ;
10339 036140 001426                 BEQ  1$
10340 036142 104455                 TRAP 55  ;
10341 036144 000076                 .WORD 76
10342 036146 007444                 .WORD ASYNC
10343 036150 000000                 .WORD 0
10344 036152 012746 006040             MOV  #WRD.19,-(SP)  ;
10345 036156 012746 007104             MOV  #FNC.11,-(SP)
10346 036162 012746 005750             MOV  #WRD.11,-(SP)
10347 036166 012746 006560             MOV  #PHR.2,-(SP)
10348 036172 012746 005630             MOV  #WRD.1,-(SP)
10349 036176 012746 005414             MOV  #FIV.FMT,-(SP)
10350 036202 012746 000006             MOV  #6,-(SP)
10351 036206 010600                 MOV  SP,RO  ; SP,*
10352 036210 104414                 TRAP 14
10353 036212 062706 000016             ADD  #16,SP  ;
10354 036216 132777 000001 153534 1$:  BITB  #1,@ML.REG+60  ;
10355 036224 001426                 BEQ  2$
10356 036226 104455                 TRAP 55  ;
10357 036230 000077                 .WORD 77
10358 036232 007444                 .WORD ASYNC
10359 036234 000000                 .WORD 0
10360 036236 012746 006040             MOV  #WRD.19,-(SP)  ;
10361 036242 012746 007104             MOV  #FNC.11,-(SP)
10362 036246 012746 005750             MOV  #WRD.11,-(SP)
10363 036252 012746 006646             MOV  #PHR.5,-(SP)
10364 036256 012746 005646             MOV  #WRD.3,-(SP)
10365 036262 012746 005414             MOV  #FIV.FMT,-(SP)
10366 036266 012746 000006             MOV  #6,-(SP)
10367 036272 010600                 MOV  SP,RO  ; SP,*
10368 036274 104414                 TRAP 14
10369 036276 062706 000016             ADD  #16,SP  ;
10370 036302 032777 020000 153450 2$:  BIT  #20000,@ML.REG+60  ;
10371 036310 001426                 BEQ  3$
10372 036312 104455                 TRAP 55  ;
10373 036314 000100                 .WORD 100
10374 036316 007444                 .WORD ASYNC
10375 036320 000000                 .WORD 0
10376 036322 012746 006040             MOV  #WRD.19,-(SP)  ;
10377 036326 012746 007104             MOV  #FNC.11,-(SP)
10378 036332 012746 005750             MOV  #WRD.11,-(SP)
10379 036336 012746 006646             MOV  #PHR.5,-(SP)
    
```

22-Oct-1980 10:47:44 TOPS
22-Oct-1980 10:45:32 PA:<

```
10381          ;ML4
10382          ;
10383
10384 036342 012746 005654      MOV      #WRD.4,-(SP)
10385 036346 012746 005414      MOV      #FIV.FMT,-(SP)
10386 036352 012746 000006      MOV      #6,-(SP)
10387 036356 010600              MOV      SP,R0          ; SP,*
10388 036360 104414              TRAP     14
10389 036362 062706 000016      ADD      #16,SP          ;
10390 036366 132777 000100 153354 3$: BITB     #100,@ML.REG+50 ;
10391 036374 001026              BNE     4$              ;
10392 036376 104455              TRAP     55              ;
10393 036400 000101              .WORD   101              ;
10394 036402 007444              .WORD   ASYNC              ;
10395 036404 000000              .WORD   0                  ;
10396 036406 012746 006040      MOV      #WRD.19,-(SP) ;
10397 036412 012746 007104      MOV      #FNC.11,-(SP) ;
10398 036416 012746 005750      MCV     #WRD.11,-(SP) ;
10399 036422 012746 006542      MOV      #PHR.1,-(SP) ;
10400 036426 012746 006034      MOV      #WRD.18,-(SP) ;
10401 036432 012746 005414      MOV      #FIV.FMT,-(SP) ;
10402 036436 012746 000006      MOV      #6,-(SP) ;
10403 036442 010600              MOV      SP,R0          ; SP,*
10404 036444 104414              TRAP     14
10405 036446 062706 000016      ADD      #16,SP          ;
10406 036452 000207 4$:      RTS     PC              ;
10407
10408          ; Routine Size: 120 words
10409          ; Maximum stack depth per invocation: 7 words
10414
10415
10419
10423 036454          T18::
10424 036454 004767 177414 1$:      JSR     PC,$T18          ;
10425 036460 104466              TRAP     66              ;
10426 036462 006000              ROR     R0
10427 036464 103773              BLO     1$
10428 036466 000207              RTS     PC
10429
10430          ; Routine Size: 6 words
10431          ; Maximum stack depth per invocation: 0 words
10439
10440
10441 ;          5553 !<BLF/PAGE>
```

22-Oct-1980 10:47:44
22-Oct-1980 10:45:32

TOPS-20 Bliss-16 V2(206)
PA:<NEALE>BL2ML4.BLI.2 (61)

```

10443 :ML4
10444 :
10445 :
10446 : 5554 !
10447 : 5555 !
10448 : 5556 BGNTST;
10449 : 5557 !
10450 : 5558 !++
10451 : 5559 ! TEST NUMBER: TST 19
10452 : 5560 !
10453 : 5561 ! TEST NAME: ILLEGAL FUNCTION TEST
10454 : 5562 !
10455 : 5563 ! TEST DESCRIPTION:
10456 : 5564 !
10457 : 5565 ! TEST THE DETECTION OF ILLEGAL
10458 : 5566 ! FUNCTIONS WRITTEN TO MLCS1
10459 : 5567 ! BY:
10460 : 5568 !
10461 : 5569 ! WRITING ALL POSSIBLE ILLEGAL
10462 : 5570 ! FUNCTIONS TO MLCS1. THEN
10463 : 5571 ! TEST GO AND ERROR BITS CLEARED.
10464 : 5572 !
10465 : 5573 ! IMPLICIT INPUTS: NONE
10466 : 5574 !
10467 : 5575 !
10468 : 5576 !--
10469 : 5577 !
10470 : 5578 local
10471 : 5579 ! BAD_BITS, !STORES A COUNT TO GENERATE BAD FUNCTIONS
10472 : 5580 ! BAD_FUNC; !STORES GENERATED BAD FUNCTION
10473 : 5581 !
10474 : 5582 BAD_BITS = ZEROES; !CLEAR BAD BITS
10475 : 5583 !
10476 : 5584 incr CNT_1 from 0 to 2 do !REPEAT LOOP 3 TIMES
10477 : 5585 ! begin
10478 : 5586 ! BAD_BITS = .BAD_BITS + %0'2'; !ADD 2 TO BAD_BITS
10479 : 5587 !
10480 : 5588 ! incr CNT_2 from %0'1' to %0'71' by %0'10' do !REPEAT LOOP GENERATING 'GOOD' FUNCTIONS
10481 : 5589 ! begin
10482 : 5590 ! BGNSUB;
10483 : 5591 ! CLR_MBUS;
10484 : 5592 ! BAD_FUNC = .CNT_2 + .BAD_BITS; !ADD BAD BITS TO CNT_2 GENERATING BAD FUNCTIONS
10485 : 5593 ! FIRST_BLK_XFER (?); !SET UP A FIRST BLOCK XFERR
10486 : 5594 ! MLCS1 = .BAD_FUNC; !LOAD MLCS1 WITH TWO BAD FUNCTIONS
10487 : 5595 !
10488 : 5596 ! if .ILF IS_SET !SEE IF ILF IS SET
10489 : 5597 ! then
10490 : 5598 ! begin
10491 : 5599 !
10492 : 5600 ! if .GO IS_SET !IF ILF IS SET THEN SEE IF GO IS SET
10493 : 5601 ! then
10494 : 5602 ! begin
10495 : 5603 ! ERRDF (66, ASYNC, 0); !ERROR IF GO SET WITH BAD FUNCTION
10496 : 5604 ! PRINTB (FIV_FMT, WRD_1, PHR_2, WRD_11, FNC_12, WRD_19);
10497 : 5605 ! PRINTB (FMT_12, .BAD_FUNC);

```

22-Oct-1980 10:47:44 TOPS-20 Bliss-16 V2(206)
22-Oct-1980 10:45:32 PA:<NEALE>BL2ML4.BLI.2 (61)

```

10499 :ML4
10500 :
10501 :
10502 : 5606
10503 : 5607
10504 : 5608
10505 : 5609
10506 : 5610
10507 : 5611
10508 : 5612
10509 : 5613
10510 : 5614
10511 : 5615
10512 : 5616
10513 : 5617
10514 : 5618
10515 : 5619
10516 : 5620
10517 : 5621
10518 : 5622
10519 : 5623
10520 : 5624
10521 : 5625
10522 : 5626
10523 : 5627
10524 : 5628
10525 : 5629
10529 :

```

```

end;
end
else
begin
ERRDF (67, ASYNC, 0); !ERROR BAD FUNCTION DON'T CAUSE ILF
PRINTB (FIV_FMT, WRD_3, PHR_1, WRD_11, FNC_12, WRD_19);
PRINTB (FMT_12, .BAD_FUNC);
end;

if .OPI IS_SET !SEE IF OPI IS SET
then
begin
ERRDF (68, ASYNC, 0); !ERROR IF SET
PRINTB (FIV_FMT, WRD_4, PHR_5, WRD_12, FNC_12, WRD_19);
PRINTB (FMT_12, .BAD_FUNC);
end;

ENDSUB;
end;

end;
ENDTST;

```

10533	036470	004167	145364	\$T19:	JSR	R1,\$SAVE5	:	5552
10534	036474	005004			CLR	R4	: BAD.BITS	5582
10535	036476	005001			CLR	R1	: CNT.1	5584
10536	036500	062704	000002	1\$:	ADD	#2,R4	: *,BAD.BITS	5586
10537	036504	012702	000001		MOV	#1,R2	: *,CNT.2	5588
10538	036510	010405		2\$:	MOV	R4,R5	: BAD.BITS,*	5592
10539	036512	060205			ADD	R2,R5	: CNT.2,*	
10540	036514	104402		3\$:	TRAP	2	:	5589
10541	036516	152777	000040	153214	BISB	#40,@ML.REG+40	:	5590
10542	036524	016700	153440		MOV	ML.DUT,R0		
10543	036530	042700	177770		BIC	#177770,R0		
10544	036534	142777	000007	153176	BICB	#7,@ML.REG+40		
10545	036542	150077	153172		BISB	R0,@ML.REG+40		
10546	036546	010503			MOV	R5,R3	: *,BAD.FUNC	5592
10547	036550	004767	153744		JSR	PC,FIRST.BLK.XFER	:	5593
10548	036554	010377	153120		MOV	R3,@ML.REG	: BAD.FUNC,*	5594
10549	036560	132777	000001	153172	BITB	#1,@ML.REG+60	:	5596
10550	036566	001440			BEQ	4\$		
10551	036570	132777	000001	153102	BITB	#1,@ML.REG	:	5600
10552	036576	001471			BEQ	6\$		

```

10554                                ;ML4
10555                                ;
10556                                ;
10557 036600 104455                TRAP    55                ;
10558 036602 000102                .WORD 102                ;
10559 036604 007444                .WORD ASYNC              ;
10560 036606 000000                .WORD 0                  ;
10561 036610 012746 006040        MOV    #WRD.19,-(SP)      ;
10562 036614 012746 007124        MOV    #FNC.12,-(SP)      ;
10563 036620 012746 005750        MOV    #WRD.11,-(SP)      ;
10564 036624 012746 006560        MOV    #PHR.2,-(SP)      ;
10565 036630 012746 005630        MOV    #WRD.1,-(SP)      ;
10566 036634 012746 005414        MOV    #FIV.FMT,-(SP)    ;
10567 036640 012746 000006        MOV    #6,-(SP)          ;
10568 036644 010600                MOV    SP,RO              ; SP,*
10569 036646 104414                TRAP   14                 ;
10570 036650 010316                MOV    R3,(SP)            ; BAD.FUNC,*
10571 036652 012746 004742        MOV    #FMT.12,-(SP)     ;
10572 036656 012746 000002        MOV    #2,-(SP)          ;
10573 036662 010600                MOV    SP,RO              ; SP,*
10574 036664 104414                TRAP   14                 ;
10575 036666 000433                BR     5$                 ;
10576 036670 104455                TRAP   55                 ;
10577 036672 000103                .WORD 103                ;
10578 036674 007444                .WORD ASYNC              ;
10579 036676 000000                .WORD 0                  ;
10580 036700 012746 006040        MOV    #WRD.19,-(SP)      ;
10581 036704 012746 007124        MOV    #FNC.12,-(SP)      ;
10582 036710 012746 005750        MOV    #WRD.11,-(SP)      ;
10583 036714 012746 006542        MOV    #PHR.1,-(SP)      ;
10584 036720 012746 005646        MOV    #WRD.3,-(SP)      ;
10585 036724 012746 005414        MOV    #FIV.FMT,-(SP)    ;
10586 036730 012746 000006        MOV    #6,-(SP)          ;
10587 036734 010600                MOV    SP,RO              ; SP,*
10588 036736 104414                TRAP   14                 ;
10589 036740 010316                MOV    R3,(SP)            ; BAD.FUNC,*
10590 036742 012746 004742        MOV    #FMT.12,-(SP)     ;
10591 036746 012746 000002        MOV    #2,-(SP)          ;
10592 036752 010600                MOV    SP,RO              ; SP,*
10593 036754 104414                TRAP   14                 ;
10594 036756 062706 000022        ADD    #22,SP              ;
10595 036762 032777 020000 152770 5$: BIT    #20000,@ML.REG+60 ;
10596 036770 001435                BEQ    7$                 ;
10597 036772 104455                TRAP   55                 ;
10598 036774 000104                .WORD 104                ;
10599 036776 007444                .WORD ASYNC              ;
10600 037000 000000                .WORD 0                  ;
10601 037002 012746 006040        MOV    #WRD.19,-(SP)      ;
10602 037006 012746 007124        MOV    #FNC.12,-(SP)      ;
10603 037012 012746 005760        MOV    #WRD.12,-(SP)      ;
10604 037016 012746 006646        MOV    #PHR.5,-(SP)      ;
10605 037022 012746 005654        MOV    #WRD.4,-(SP)      ;
10606 037026 012746 005414        MOV    #FIV.FMT,-(SP)    ;
10607 037032 012746 000006        MOV    #6,-(SP)          ;
10608 037036 010600                MOV    SP,RO              ; SP,*

```

22-Oct-1980 10:47:44 TOPS
22-Oct-1980 10:45:32 PA:<

5603

5604

5605

5602
5611

5612

5613

5610
5616

5619

5620

```
10610 ;ML4
10611 ;
10612 ;
10613 037040 104414 TRAP 14
10614 037042 010316 MOV R3,(SP) ; BAD.FUNC,* 5621
10615 037044 012746 004742 MOV #FMT.12,-(SP)
10616 037050 012746 000002 MOV #2,-(SP)
10617 037054 010600 MOV SP,R0 ; SP,*
10618 037056 104414 TRAP 14
10619 037060 062706 000022 ADD #22,SP ; 5618
10620 037064 104467 7$: TRAP 67 ; 5622
10621 037066 006000 ROR R0
10622 037070 103611 BLO 3$
10623 037072 062702 000010 ADD #10,R2 ; *,CNT.2 5588
10624 037076 020227 000071 CMP R2,#71 ; CNT.2,*
10625 037102 003602 BLE 2$
10626 037104 005201 INC R1 ; CNT.1 5584
10627 037106 020127 000002 CMP R1,#2 ; CNT.1,*
10628 037112 003002 BGT 8$
10629 037114 000167 177360 JMP 1$
10630 037120 000207 8$: RTS PC ; 5552
10631
10632 ; Routine Size: 141 words
10633 ; Maximum stack depth per invocation: 15 words
10638
10639
10643
10647 037122 T19::
10648 037122 004767 177342 1$: JSR PC,$T19 ; 5627
10649 037126 104466 TRAP 66
10650 037130 006000 ROR R0
10651 037132 103773 BLO 1$
10652 037134 000207 RTS PC
10653
10654 ; Routine Size: 6 words
10655 ; Maximum stack depth per invocation: 0 words
10660
10661
10662 ; 5630 !<BLF/PAGE>
```


10664 :ML4

22-Oct-1980 10:47:44
22-Oct-1980 10:45:32

TOPS-20 Bliss-16 V2(206)
PA:<NEALE>BL2ML4.BLI.2 (62)

10665 :
10666 :
10667 :
10668 :
10669 :
10670 :
10671 :
10672 :
10673 :
10674 :
10675 :
10676 :
10677 :
10678 :
10679 :
10680 :
10681 :
10682 :
10683 :
10684 :
10685 :
10686 :
10687 :
10688 :
10689 :
10690 :
10691 :
10692 :
10693 :
10694 :
10695 :
10696 :
10697 :
10698 :
10699 :
10700 :
10701 :
10702 :
10703 :
10704 :
10705 :
10706 :
10707 :
10708 :
10709 :
10710 :
10711 :
10712 :
10713 :
10714 :
10715 :
10716 :
10717 :
10718 :

5631
5632
5633
5634
5635
5636
5637
5638
5639
5640
5641
5642
5643
5644
5645
5646
5647
5648
5649
5650
5651
5652
5653
5654
5655
5656
5657
5658
5659
5660
5661
5662
5663
5664
5665
5666
5667
5668
5669
5670
5671
5672
5673
5674
5675
5676
5677
5678
5679
5680
5681
5682

BGNTST;

!++

TEST NUMBER: TST 20

TEST NAME: REGISTER MODIFICATION REFUSED TEST

TEST DESCRIPTION:

TEST THE DETECTION OF A
REGISTER MODIFICATION REFUSED
BY:

1. WRITTING TO MLCS1, MLDA
AND MLER WHILE THE DRIVE
IS BUSY AND TEST RMR
BIT SET.

ALSO SEE IF THE DRIVE ASSERTED
EXCEPTION BY TESTING THE TRE BIT SET.

IMPLICIT INPUTS: NONE

!--

incr CNT from 0 to 2 do

!REPEAT LOOP 3 TIMES

begin

BGNSUB;

CLR MBUS;

MLCS1 = write;

!DO A WRITE FUNCTION

case .CNT from 0 to 2 of

!WRITE TO SELECTED REGISTERS FORCING RMR

set

[0] :
MLCS1 = %o'000000';

[1] :
MLDA = ONES;

[2] :
MLER = ONES

tes;

DELAY (FRTY_US);

if .RMR IS_NOT_SET
then

!SEE IF RMR GOT SET

begin

22-Oct-1980 10:47:44
22-Oct-1980 10:45:32

TOPS-20 Bliss-16 V2(206)
PA:<NEALE>BL2ML4.BLI.2 (62)

10720 :ML4
10721 :
10722 :
10723 : 5683
10724 : 5684
10725 : 5685
10726 : 5686
10727 : 5687
10728 : 5688
10729 : 5689
10730 : 5690
10731 : 5691
10732 : 5692
10733 : 5693
10734 : 5694
10735 : 5695
10736 : 5696
10737 : 5697

```
ERRDF (69, ASYNC, 0);  
PRINTB (FOR_FMT, WRD_21, PHR_1, WRD_11, WRD_21);  
end;  
  
if .TRE IS_NOT_SET  
then  
begin  
ERRDF (117, SYNC, 0);  
PRINTB (FOR_FMT, WRD_20, PHR_1, WRD_11, WRD_21);  
end;  
  
ENDSUB;  
end;  
  
ENDTST;
```

!ERROR IF NOT SET
!SEE IF DRIVE ASSERTED EXCEPTION BY TESTING TRE

```
10741  
10745 037136 004167 144644 $T20: JSR R1,$SAVE2 ; 5629  
10746 037142 005746 TST -(SP) ;  
10747 037144 005002 CLR R2 ; CNT 5659  
10748 037146 104402 1$: TRAP 2 ; 5660  
10749 037150 152777 000040 152562 BISB #40,@ML.REG+40 ; 5661  
10750 037156 016701 153006 MOV ML,DUT,R1  
10751 037162 042701 177770 BIC #177770,R1  
10752 037166 142777 000007 152544 BICB #7,@ML.REG+40  
10753 037174 150177 152540 BISB R1,@ML.REG+40  
10754 037200 012777 000061 152472 MOV #61,@ML.REG ; 5663  
10755 037206 010201 MOV R2,R1 ; CNT,* 5665  
10756 037210 006301 ASL R1  
10757 037212 066107 037216 ADD 2$(R1),PC  
10758 037216 000006 2$: .WORD 3$-2$  
10759 037220 000014 .WORD 4$-2$  
10760 037222 000024 .WORD 5$-2$  
10761 037224 005077 152450 3$: CLR @ML.REG ; 5669  
10762 037230 000407 BR 6$ ; 5665  
10763 037232 012777 177777 152470 4$: MOV #-1,@ML.REG+30 ; 5672  
10764 037240 000403 BR 6$ ; 5665  
10765 037242 012777 177777 152510 5$: MOV #-1,@ML.REG+60 ; 5675  
10766 037250 012700 000050 6$: MOV #50,R0 ; *,$STMP2 5678  
10767 037254 001410 7$: BEQ 10$ ;  
10768 037256 016701 142634 MOV L$DLY,R1 ; *,$STMP1  
10769 037262 001403 BEQ 9$ ;  
10770 037264 005016 8$: CLR (SP) ; $STMP  
10771 037266 005301 DEC R1 ; $STMP1  
10772 037270 001375 BNE 8$ ;  
10773 037272 005300 9$: DEC R0 ; $STMP2  
10774 :ML4  
10775 :  
10776 :  
10777 037274 000767 BR 7$  
10778 037276 132777 000004 152454 10$: BITB #4,@ML.REG+60 ; 5680  
10779 037304 001024 BNE 11$ ;  
10780 037306 104455 TRAP 55 ; 5683  
10781 037310 000105 .WORD 105  
10782 037312 007444 .WORD ASYNC
```

22-Oct-1980 10:47:44 TOPS
22-Oct-1980 10:45:32 PA:<

10783	037314	000000			.WORD	0			
10784	037316	012746	006054		MOV	#WRD.21,-(SP)	:		5684
10785	037322	012746	005750		MOV	#WRD.11,-(SP)	:		
10786	037326	012746	006542		MOV	#PHR.1,-(SP)	:		
10787	037332	012746	006054		MOV	#WRD.21,-(SP)	:		
10788	037336	012746	005400		MOV	#FOR.FMT,-(SP)	:		
10789	037342	012746	000005		MOV	#5,-(SP)	:		
10790	037346	010600			MOV	SP,R0	:	SP,*	
10791	037350	104414			TRAP	14	:		
10792	037352	062706	000014		ADD	#14,SP	:		5682
10793	037356	032777	040000	152314	BIT	#40000,@ML.REG	:		5687
10794	037364	001024			BNE	12\$:		
10795	037366	104455			TRAP	55	:		5690
10796	037370	000165			.WORD	165	:		
10797	037372	007500			.WORD	SYNC	:		
10798	037374	000000			.WORD	0	:		
10799	037376	012746	006054		MOV	#WRD.21,-(SP)	:		5691
10800	037402	012746	005750		MOV	#WRD.11,-(SP)	:		
10801	037406	012746	006542		MOV	#PHR.1,-(SP)	:		
10802	037412	012746	006046		MOV	#WRD.20,-(SP)	:		
10803	037416	012746	005400		MOV	#FOR.FMT,-(SP)	:		
10804	037422	012746	000005		MOV	#5,-(SP)	:		
10805	037426	010600			MOV	SP,R0	:	SP,*	
10806	037430	104414			TRAP	14	:		
10807	037432	062706	000014		ADD	#14,SP	:		5689
10808	037436	104467		12\$:	TRAP	67	:		5692
10809	037440	006000			ROR	R0	:		
10810	037442	103641			BLO	1\$:		
10811	037444	005202			INC	R2	:	CNT	5659
10812	037446	020227	000002		CMP	R2,#2	:	CNT,*	
10813	037452	003635			BLE	1\$:		
10814	037454	005726			TST	(SP)+	:		5629
10815	037456	000207			RTS	PC	:		
10823									
10827	037460			T20::					
10828	037460	004767	177452	1\$:	JSR	PC,\$T20	:		5695
10829	037464	104466			TRAP	66	:		
10830	037466	006000			ROR	R0	:		
10831	037470	103773			BLO	1\$:		
10832	037472	000207			RTS	PC	:		
10833									
10834									
10835									

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

22-Oct-1980 10:47:44 TOPS-20 Bliss-16 V2(206)
22-Oct-1980 10:45:32 PA:<NEALE>BL2ML4.BLI.2 (63)

10841 :ML4
10842 :
10843 :
10844 :
10845 :
10846 :
10847 :
10848 :
10849 :
10850 :
10851 :
10852 :
10853 :
10854 :
10855 :
10856 :
10857 :
10858 :
10859 :
10860 :
10861 :
10862 :
10863 :
10864 :
10865 :
10866 :
10867 :
10868 :
10869 :
10870 :
10871 :
10872 :
10873 :
10874 :
10875 :
10876 :
10877 :
10878 :
10879 :
10880 :
10881 :
10882 :
10883 :
10884 :
10885 :
10886 :
10887 :
10888 :
10889 :
10890 :
10891 :
10892 :
10893 :
10894 :
10895 :

```
5699 :  
5700 :  
5701 : BGNTST;  
5702 :  
5703 : !++  
5704 : TEST NUMBER: TST 21  
5705 :  
5706 : TEST NAME : initial PROM TEST  
5707 :  
5708 : TEST DESCRIPTION:  
5709 :  
5710 : TEST THE MEMORY ARRAYS' PROM  
5711 : TIMING AND CONTROL LOGIC FOR  
5712 : INITIAL PROM READS BY:  
5713 :  
5714 : 1. READING 14 PROM LOCATIONS  
5715 : AND TESTING FOR:  
5716 :  
5717 : A. CHECK SUM ERRORS AT  
5718 : EACH ROW COLUMN ADRS  
5719 :  
5720 : B. SUM OF EACH PROM BIT SET  
5721 : <9,0> GTR 14.  
5722 :  
5723 : IMPLICIT INPUTS: NONE  
5724 :  
5725 :  
5726 : --  
5727 :  
5728 : local  
5729 : OFF_SET_CNT : vector [10, byte],  
5730 : ROW_ORED_COL : bitvector [16],  
5731 : R_C_SAV : bitvector [16],  
5732 : PROM_ADRS,  
5733 : CHK_SUM,  
5734 : CHK_SUM_ERR,  
5735 : BAD_NIB_CNT,  
5736 : CNT_14_BAD,  
5737 : DOD0_FLG,  
5738 : ERR_FLG;  
5739 :  
5740 : CLR MBUS;  
5741 : DOD0_FLG = ZERO;  
5742 : PROM_DIS = ONE;  
5743 : CHK_SUM_ERR = ZEROES;  
5744 :  
5745 : incr CNT from 0 to 9 do  
5746 : OFF_SET_CNT [.CNT] = ZEROES;  
5747 :  
5748 : incr ADRS_CNT from 0 to 14 do  
5749 : begin  
5750 : ROW_ORED_COL = ZEROES;
```

```
!COUNTS EACH NIBBLE OFFSET  
!SAVES ROW DATA ORED WITH COL DATA  
!TEMP LOCATION FOR ROW COL DATA  
!PROM ADDRESS  
!CHECK SUM DATA  
!CHECK SUM ERROR  
!COUNTS BAD NIBBLES  
!COUNTS BAD NIBBLE POSITION EQL 14  
!DROP UNIT FLAG  
!ERROR FLAG  
  
!SET PROM DISABLE MODE  
  
!CLEAR OFFSET COUNTS  
  
!READ PROM DATA FROM 15 ARRAY WORDS  
!CLEAR ROW ORED COL SAVE LOCATION
```

22-Oct-1980 10:47:44 TOPS-20 Bliss-16 V2(206)
22-Oct-1980 10:45:32 PA:<NEALE>BL2ML4.BLI.2 (63)

```

10897 ;ML4
10898 ;
10899 ;
10900 : 5751 PROM_ADRS = .ADRS_CNT; !GET COPY OF ADRS_CNT
10901 : 5752
10902 : 5753 incr TWICE from 0 to 1 do !READ ROW AND COL DATA FOR THIS ARRAY WORD
10903 : 5754 begin
10904 : 5755 BAD_NIB_CNT = ZEROES;
10905 : 5756 ERR_FLG = ZERO;
10906 : 5757 MLPA = .PROM_ADRS; !LOADING MLPA INITIATES PROM READ
10907 : 5758 DELAY (ONE_US);
10908 : 5759 R_C_SAV = .MLPD; !GET THE ROW OR COL DATA
10909 : 5760
10910 : 5761 incr BIT_CNT from 0 to 9 do !COUNT NUMBER BITS SET IN <9:0>
10911 : 5762
10912 : 5763 if .R_C_SAV [.BIT_CNT] IS_SET then BAD_NIB_CNT = .BAD_NIB_CNT + 1;
10913 : 5764
10914 : 5765 CHK_SUM = .R_C_SAV<10, 3>; !GET THE CHECK SUM BITS
10915 : 5766
10916 : 5767 if .R_C_SAV [13] IS_SET then CHK_SUM = .CHK_SUM + 1; !ADD IN BIT 13
10917 : 5768
10918 : 5769 if .R_C_SAV [14] IS_SET then CHK_SUM = .CHK_SUM + 1; !ADD IN BIT 14
10919 : 5770
10920 : 5771 if .R_C_SAV [15] IS_SET then CHK_SUM = .CHK_SUM + 1; !ADD IN BIT 15
10921 : 5772
10922 : 5773 if .R_C_SAV [15] IS_SET !IS BIT 15 SET
10923 : 5774 then
10924 : 5775 begin
10925 : 5776
10926 : 5777 if .BAD_NIB_CNT lss .CHK_SUM then ERR_FLG = ONE; !SET ERROR FLG IF CHECK SUM ERROR
10927 : 5778
10928 : 5779 end
10929 : 5780 else
10930 : 5781 begin
10931 : 5782
10932 : 5783 if .BAD_NIB_CNT neq .CHK_SUM then ERR_FLG = ONE; !SET ERROR FLG IF CHECK SUM ERROR
10933 : 5784
10934 : 5785 end;
10935 : 5786
10936 : 5787 if .ERR_FLG IS_SET !WAS THERE A CHECK SUM ERROR
10937 : 5788 then
10938 : 5789 begin !REPORT INTERMEDIATE ERROR IF YES
10939 : 5790 ERRDF (70, INTER, 0);
10940 : 5791 PRINTB (SIX_FMT, FNC_21, WRD_10, WRD_12, WRD_45, WRD_35, FNC_6);
10941 : 5792 PRINTB (FMT_10, .CHK_SUM, .R_C_SAV);
10942 : 5793 DODU_FLG = ONE;
10943 : 5794 end;
10944 : 5795
10945 : 5796 ROW_ORED_COL = (.ROW_ORED_COL) or (.R_C_SAV); !OR ROW AND COLUMN DATA
10946 : 5797 PROM_ADRS = (.PROM_ADRS) or (%o'2000'); !GET COLUMN DATA
10947 : 5798 end;
10948 : 5799
10949 : 5800 incr index from 0 to 9 do !GET TOTAL OFF SET COUNTS FROM THE 15 ARRAY WORDS
10950 : 5801
10951 : 5802 if .ROW_ORED_COL [.index] IS_SET then OFF_SET_CNT [.index] = .OFF_SET_CNT [.index] + 1;

```

```

10953 :ML4
10954 :
10955 :
10956 : 5803
10957 : 5804     end;
10958 : 5805
10959 : 5806     CNT_14_BAD = ZEROES;
10960 : 5807
10961 : 5808     incr CNT from 0 to 9 do                !COUNT HOW MANY OFFSET COUNTS GEQ 14
10962 : 5809
10963 : 5810         if .OFF_SET_CNT [.CNT] geq 14 then CNT_14_BAD = .CNT_14_BAD + 1;
10964 : 5811
10965 : 5812     if .CNT_14_BAD neq ZERO                !WAS ANY OFF_SET COUNTS GTR ZERO
10966 : 5813     then
10967 : 5814         begin
10968 : 5815         ERRDF (71, ARR_DAT, 0);                !ERROR IF YES
10969 : 5816
10970 : 5817         if .CNT_14_BAD eql 10                !WERE ALL 10 NIBBLE OFFSETS GEQ 14
10971 : 5818         then
10972 : 5819         PRINTB (SIX_FMT, WRD_55, WRD_54, WRD_46, WRD_51, WRD_42, WRD_40) !ERROR IF YES
10973 : 5820         else
10974 : 5821         begin
10975 : 5822         PRINTB (FOR_FMT, WRD_46, WRD_47, WRD_42, WRD_40); !PRINT MESSAGE
10976 : 5823
10977 : 5824         incr CNT from 0 to 9 do                !FIND OFFSETS GEQ 14
10978 : 5825
10979 : 5826         if .OFF_SET_CNT [.CNT] geq 14 then PRINTB (FMT_13, .CNT, (.OFF_SET_CNT [.CNT]));
10980 : 5827
10981 : 5828
10982 : 5829         end;
10983 : 5830
10984 : 5831         DODU_FLG = ONE;
10985 : 5832         end;
10986 : 5833
10987 : 5834     if .DODU_FLG IS_SET                !DROP THIS UNIT IF DODU_FLG SET
10988 : 5835     then
10989 : 5836         begin
10990 : 5837         DODU (.ML_LUN);
10991 : 5838         DOCLN;
10992 : 5839         end;
10993 : 5840
10994 : 5841     ENDTST;
  
```

11002	037474	004167	144360	\$T21:	JSR	R1,\$SAVE5	:	5697
11003	037500	162706	000030		SUB	#30,SP	:	
11004	037504	152777	000040	152226	BISB	#40,@ML.REG+40	:	5738
11005	037512	016705	152452		MOV	ML.DUT,R5	:	
11006	037516	042705	177770		BIC	#177770,R5	:	
11007								
11008								
11009								
11010	037522	142777	000007	152210	BICB	#7,@ML.REG+40	:	
11011	037530	150577	152204		BISB	R5,@ML.REG+40	:	
11012	037534	005066	000002		CLR	2(SP)	:	DODU.FLG
11013	037540	152777	000040	152252	BISB	#40,@ML.REG+120	:	5741
11014	037546	005002			CLR	R2	:	5742
11015	037550	012701	000016	1\$:	MOV	#16,R1	:	5743
							:	5746

11016	037554	060601			ADD	SP,R1		: OFF.SET.CNT,*	
11017	037556	060201			ADD	R2,R1		: CNT,*	
11018	037560	105011			CLRB	(R1)			
11019	037562	005202			INC	R2		: CNT	5745
11020	037564	020227	000011		CMP	R2,#11		: CNT,*	
11021	037570	003767			BLE	1\$			
11022	037572	005005			CLR	R5		: ADRS.CNT	5748
11023	037574	005066	000010	2\$:	CLR	10(SP)		: ROW.ORED.COL	5750
11024	037600	010566	000006		MOV	R5,6(SP)		: ADRS.CNT,PROM.ADRS	5751
11025	037604	005004			CLR	R4		: TWICE	5753
11026	037606	005066	000004	3\$:	CLR	4(SP)		: BAD.NIB.CNT	5755
11027	037612	005016			CLR	(SP)		: ERR.FLG	5756
11028	037614	016677	000006	152156	MOV	6(SP),@ML.REG+100		: PROM.ADRS,*	5757
11029	037622	012701	000001		MOV	#1,R1		: *,\$STMP2	5758
11030	037626	001411			BEQ	7\$			
11031	037630	016702	142262		MOV	L\$DLY,R2		: *,\$STMP1	
11032	037634	001404			BEQ	6\$			
11033	037636	005066	000014	5\$:	CLR	14(SP)		: \$STMP	
11034	037642	005302			DEC	R2		: \$STMP1	
11035	037644	001374			BNE	5\$			
11036	037646	005301		6\$:	DEC	R1		: \$STMP2	
11037	037650	000766			BR	4\$			
11038	037652	017766	152252	000012	MOV	@ML.REG+230,12(SP)		: *,R.C.SAV	5759
11039	037660	005002			CLR	R2		: BIT.CNT	5761
11040	037662	010201		8\$:	MOV	R2,R1		: BIT.CNT,*	5763
11041	037664	006201			ASR	R1			
11042	037666	006201			ASR	R1			
11043	037670	006201			ASR	R1			
11044	037672	012700	000012		MOV	#12,R0			
11045	037676	060600			ADD	SP,R0		: R.C.SAV,*	
11046	037700	060001			ADD	R0,R1			
11047	037702	010146			MOV	R1,-(SP)			
11048	037704	010246			MOV	R2,-(SP)		: BIT.CNT,*	
11049	037706	042716	177770		BIC	#177770,(SP)			
11050	037712	012746	000001		MOV	#1,-(SP)			
11051	037716	005046			CLR	-(SP)			
11052	037720	004767	143156		JSR	PC,BL\$GT2			
11053	037724	062706	000010		ADD	#10,SP			
11054	037730	005300			DEC	R0			
11055	037732	001002			BNE	9\$			
11056	037734	005266	000004		INC	4(SP)		: BAD.NIB.CNT	
11057	037740	005202		9\$:	INC	R2		: BIT.CNT	5761
11058	037742	020227	000011		CMP	R2,#11		: BIT.CNT,*	
11059	037746	003745			BLE	8\$			
11060	037750	016603	000012		MOV	12(SP),R3		: R.C.SAV,CHK.SUM	5765
11061	037754	006203			ASR	R3		: CHK.SUM	
11062				:ML4					
11063				:					
11064									
11065	037756	006203			ASR	R3		: CHK.SUM	
11066	037760	000303			SWAB	R3		: CHK.SUM	
11067	037762	042703	177770		BIC	#177770,R3		: *,CHK.SUM	
11068	037766	132766	000040	000013	BITB	#40,13(SP)		: *,R.C.SAV+1	5767
11069	037774	001401			BEQ	10\$			
11070	037776	005203			INC	R3		: CHK.SUM	
11071	040000	132766	000100	000013	BITB	#100,13(SP)		: *,R.C.SAV+1	5769
11072	040006	001401		10\$:	BEQ	11\$			

22-Oct-1980 10:47:44
22-Oct-1980 10:45:32

TOPS
PA:<

11073	040010	005203			INC	R3		: CHK.SUM	
11074	040012	005002			CLR	R2		:	5771
11075	040014	105766	000013		TSTB	13(SP)		: R.C.SAV+1	
11076	040020	100002			BPL	12\$			
11077	040022	005202			INC	R2			
11078	040024	005203			INC	R3		: CHK.SUM	
11079	040026	006002			ROR	R2		:	5773
11080	040030	103004			BCC	13\$			
11081	040032	026603	000004		CMP	4(SP),R3		: BAD.NIB.CNT,CHK.SUM	5777
11082	040036	002006			BGE	15\$			
11083	040040	000403			BR	14\$			
11084	040042	026603	000004		CMP	4(SP),R3		: BAD.NIB.CNT,CHK.SUM	5783
11085	040046	001402			BEQ	15\$			
11086	040050	012716	000001		MOV	#1,(SP)		: *,ERR.FLG	
11087	040054	021627	000001		CMP	(SP),#1		: ERR.FLG,*	5787
11088	040060	001044			BNE	16\$			
11089	040062	104455			TRAP	55		:	5790
11090	040064	000106			.WORD	106			
11091	040066	007622			.WORD	INTER			
11092	040070	000000			.WORD	0			
11093	040072	012746	007030		MOV	#FNC.6,-(SP)		:	5791
11094	040076	012746	006220		MOV	#WRD.35,-(SP)			
11095	040102	012746	006324		MOV	#WRD.45,-(SP)			
11096	040106	012746	005760		MOV	#WRD.12,-(SP)			
11097	040112	012746	005740		MOV	#WRD.10,-(SP)			
11098	040116	012746	007242		MOV	#FNC.21,-(SP)			
11099	040122	012746	005432		MOV	#SIX.FMT,-(SP)			
11100	040126	012746	000007		MOV	#7,-(SP)			
11101	040132	010600			MOV	SP,R0		: SP,*	
11102	040134	104414			TRAP	14			
11103	040136	016616	000032		MOV	32(SP),(SP)		: R.C.SAV,*	5792
11104	040142	010346			MOV	R3,-(SP)		: CHK.SUM,*	
11105	040144	012746	004634		MOV	#FMT.10,-(SP)			
11106	040150	012746	000003		MOV	#3,-(SP)			
11107	040154	010600			MOV	SP,R0		: SP,*	
11108	040156	104414			TRAP	14			
11109	040160	012766	000001	000030	MOV	#1,30(SP)		: *,DODU.FLG	5793
11110	040166	062706	000026		ADD	#26,SP			5789
11111	040172	056666	000012	000010	16\$: BIS	12(SP),10(SP)		: R.C.SAV,ROW.ORED.COL	5796
11112	040200	052766	002000	000006	BIS	#2000,6(SP)		: *,PROM.ADRS	5797
11113	040206	005204			INC	R4		: TWICE	5753
11114	040210	020427	000001		CMP	R4,#1		: TWICE,*	
11115	040214	003002			BGT	17\$			
11116	040216	000167	177364		JMP	3\$			
11117					:ML4				22-Oct-1980 10:47:44
11118					:				22-Oct-1980 10:45:32
11119									TOPS
11120	040222	005002			17\$: CLR	R2		: INDEX	5800
11121	040224	010201			18\$: MOV	R2,R1		: INDEX,*	5802
11122	040226	006201			ASR	R1			
11123	040230	006201			ASR	R1			
11124	040232	006201			ASR	R1			
11125	040234	012700	000010		MOV	#10,R0			
11126	040240	060600			ADD	SP,R0		: ROW.ORED.COL,*	
11127	040242	060001			ADD	R0,R1			
11128	040244	010146			MOV	R1,-(SP)			
11129	040246	010246			MOV	R2,-(SP)		: INDEX,*	

11130	040250	042716	177770	BIC	#177770,(SP)			
11131	040254	012746	000001	MOV	#1,-(SP)			
11132	040260	005046		CLR	-(SP)			
11133	040262	004767	142614	JSR	PC,BL\$GT2			
11134	040266	062706	000010	ADD	#10,SP			
11135	040272	005300		DEC	R0			
11136	040274	001005		BNE	19\$			
11137	040276	012701	000016	MOV	#16,R1			
11138	040302	060601		ADD	SP,R1		; OFF.SET.CNT,*	
11139	040304	060201		ADD	R2,R1		; INDEX,*	
11140	040306	105211		INCB	(R1)			
11141	040310	005202		19\$: INC	R2		; INDEX	5800
11142	040312	020227	000011	CMP	R2,#11		; INDEX,*	
11143	040316	003742		BLE	18\$			
11144	040320	005205		INC	R5		; ADRS.CNT	5748
11145	040322	020527	000016	CMP	R5,#16		; ADRS.CNT,*	
11146	040326	003002		BGT	20\$			
11147	040330	000167	177240	JMP	2\$			
11148	040334	005000		20\$: CLR	R0		; CNT.14.BAD	5806
11149	040336	005001		CLR	R1		; CNT	5808
11150	040340	012702	000016	21\$: MOV	#16,R2		; CNT	5810
11151	040344	060602		ADD	SP,R2		; OFF.SET.CNT,*	
11152	040346	060102		ADD	R1,R2		; CNT,*	
11153	040350	121227	000016	CMPB	(R2),#16			
11154	040354	103401		BLO	22\$			
11155	040356	005200		INC	R0		; CNT.14.BAD	
11156	040360	005201		22\$: INC	R1		; CNT	5808
11157	040362	020127	000011	CMP	R1,#11		; CNT,*	
11158	040366	003764		BLE	21\$			
11159	040370	005700		TST	R0		; CNT.14.BAD	5812
11160	040372	001505		BEQ	27\$			
11161	040374	104455		TRAP	55			5815
11162	040376	000107		.WORD	107			
11163	040400	007534		.WORD	ARR.DAT			
11164	040402	000000		.WORD	0			
11165	040404	020027	000012	CMP	R0,#12		; CNT.14.BAD,*	5817
11166	040410	001024		BNE	23\$			
11167	040412	012746	006260	MOV	#WRD.40,-(SP)			5819
11168	040416	012746	006300	MOV	#WRD.42,-(SP)			
11169	040422	012746	006406	MOV	#WRD.51,-(SP)			
11170	040426	012746	006336	MOV	#WRD.46,-(SP)			
11171	040432	012746	006436	MOV	#WRD.54,-(SP)			
11172				:ML4				
11173				:				
11174								
11175	040436	012746	006446	MOV	#WRD.55,-(SP)			
11176	040442	012746	005432	MOV	#SIX.FMT,-(SP)			
11177	040446	012746	000007	MOV	#7,-(SP)			
11178	040452	010600		MOV	SP,R0		; SP,*	
11179	040454	104414		TRAP	14			
11180	040456	022626		CMP	(SP)+,(SP)+			
11181	040460	000445		BR	26\$			5817
11182	040462	012746	006260	23\$: MOV	#WRD.40,-(SP)			5822
11183	040466	012746	006300	MOV	#WRD.42,-(SP)			
11184	040472	012746	006350	MOV	#WRD.47,-(SP)			
11185	040476	012746	006336	MOV	#WRD.46,-(SP)			
11186	040502	012746	005400	MOV	#FOR.FMT,-(SP)			

22-Oct-1980 10:47:44 TOPS
22-Oct-1980 10:45:32 PA:<

11187	040506	012746	000005			MOV	#5,-(SP)			
11188	040512	010600				MOV	SP,R0		; SP,*	
11189	040514	104414				TRAP	14			
11190	040516	005002				CLR	R2		; CNT	5824
11191	040520	012701	000032	24\$:		MOV	#32,R1			5826
11192	040524	060601				ADD	SP,R1		; OFF.SET.CNT,*	
11193	040526	060201				ADD	R2,R1		; CNT,*	
11194	040530	121127	000016			CMPB	(R1),#16			
11195	040534	103413				BLO	25\$			
11196	040536	005046				CLR	-(SP)			
11197	040540	111116				MOVB	(R1),(SP)			
11198	040542	010246				MOV	R2,-(SP)		; CNT,*	
11199	040544	012746	004772			MOV	#FMT.13,-(SP)			
11200	040550	012746	000003			MOV	#3,-(SP)			
11201	040554	010600				MOV	SP,R0		; SP,*	
11202	040556	104414				TRAP	14			
11203	040560	062706	000010			ADD	#10,SP			
11204	040564	005202		25\$:		INC	R2		; CNT	5824
11205	040566	020227	000011			CMP	R2,#11		; CNT,*	
11206	040572	003752				BLE	24\$			
11207	040574	012766	000001	000016	26\$:	MOV	#1,16(SP)		; *,DODU.FLG	5831
11208	040602	062706	000014			ADD	#14,SP			5814
11209	040606	026627	000002	000001	27\$:	CMP	2(SP),#1		; DODU.FLG,*	5834
11210	040614	001004				BNE	28\$			
11211	040616	016700	151344			MOV	ML.LUN,R0			5837
11212	040622	104451				TRAP	51			
11213	040624	104444				TRAP	44			
11214	040626	062706	000030	28\$:		ADD	#30,SP			5697
11215	040632	000207				RTS	PC			
11216										
11217										
11218										
11223										
11224										
11231										
11235	040634				T21::					
11236	040634	004767	176634		1\$:	JSR	PC,\$T21			5839
11237	040640	104466				TRAP	66			
11238	040642	006000				ROR	R0			
11239	040644	103773				BLO	1\$			
11240	040646	000207				RTS	PC			

; Routine Size: 304 words
; Maximum stack depth per invocation: 29 words

22-Oct-1980 10:47:44 TOPS-20 Bliss-16 V2(206)
 22-Oct-1980 10:45:32 PA:<NEALE>BL2ML4.BLI.2 (64)

```

11246 :ML4
11247 :
11248 :
11249 : 5843 :
11250 : 5844 :
11251 : 5845 : BGNTST;
11252 : 5846 :
11253 : 5847 : !++
11254 : 5848 : TEST NUMBER: TST 22
11255 : 5849 :
11256 : 5850 : TEST NAME: PROM OR FUNCTION TEST
11257 : 5851 :
11258 : 5852 : TEST DESCRIPTION:
11259 : 5853 :
11260 : 5854 : TEST THE HARDWARE ORING OF
11261 : 5855 : THE MEMORY ARRAYS' PROM
11262 : 5856 : ROW COLUMN DATA BY:
11263 : 5857 :
11264 : 5858 : 1. READING AND STORING 128
11265 : 5859 : HARDWARE ORED ROW COLUMN
11266 : 5860 : DATA.
11267 : 5861 :
11268 : 5862 : 2. THEN IN PROM DISABLE MODE
11269 : 5863 : AND VIA SOFTWARE CONTROL,
11270 : 5864 : READ AND OR PROM ROW
11271 : 5865 : COLUMN DATA AND COMPARE
11272 : 5866 : AGAINST THE RESPECTIVE
11273 : 5867 : STORED HARDWARE ORED DATA.
11274 : 5868 :
11275 : 5869 : IMPLICIT INPUTS: NONE
11276 : 5870 :
11277 : 5871 :
11278 : 5872 : --
11279 : 5873 :
11280 : 5874 : local
11281 : 5875 : R_BITS,
11282 : 5876 : C_BITS,
11283 : 5877 : SW_ORED,
11284 : 5878 : HW_SAVE,
11285 : 5879 : DODU_FLG;
11286 : 5880 :
11287 : 5881 : CLR_MBUS;
11288 : 5882 : DODU_FLG = ZERO;
11289 : 5883 : DAT_DM = ONE;
11290 : 5884 : FIRST_BLK_XFER ();
11291 : 5885 : ML_FUNC = write;
11292 : 5886 :
11293 : 5887 : incr PROM_ADRS from 0 to 127 do
11294 : 5888 : begin
11295 : 5889 : DELAY (ONE US);
11296 : 5890 : HW OR TBL [.PROM_ADRS] = .MLPD;
11297 : 5891 : DAT_C[K = ONE;
11298 : 5892 : end;
11299 : 5893 :
11300 : 5894 : CLR_MBUS;
  
```

```

!PROM ROW DATA
!PROM COL DATA
!SOFTWARE CALCULATED PROM ORED DATA
!SOFTWARE PROM ORED DATA
!DROP UNIT FLAG

!SET UP A FIRST BLOCK XFER
!DO A WRITE FUNCTION

!READ AND STORE 128 HARDWARE PROM ORED DATA

!READ HARDWARE PROM ORED DATA
!CLOCK NEXT ONE OUT
  
```

22-Oct-1980 10:47:44 TOPS-20 Bliss-16 V2(206)
22-Oct-1980 10:45:32 PA:<NEALE>BL2ML4.BLI.2 (64)

```

11302 :ML4
11303 :
11304 :
11305 :      5895 PROM_DIS = ONE;
11306 :      5896
11307 :      5897 incr PROM_ADRS from 0 to 127 do
11308 :      5898 begin
11309 :      5899 MLPA = .PROM_ADRS;
11310 :      5900 DELAY (ONE US);
11311 :      5901 R_BITS = .MLPD;
11312 :      5902 M[PA = .PROM_ADRS or %o'2000';
11313 :      5903 DELAY (ONE US);
11314 :      5904 C_BITS = .MLPD;
11315 :      5905 SW_ORED = .R_BITS or .C_BITS;
11316 :      5906 HW_SAVE = .HW_OR_TBL [.PROM_ADRS];
11317 :      5907
11318 :      5908 if (.HW_SAVE<0, 9>) neq (.SW_ORED<0, 9>)
11319 :      5909 then
11320 :      5910 begin
11321 :      5911 ERRDF (76, ARR DAT, 0);
11322 :      5912 PRINTB (FOR_FMT, WRD 35, WRD 36, WRD 19, PHR 4);
11323 :      5913 PRINTB (FMT_2, .SW_ORED<0, 9>, .HW_SAVE<0, 9>, (.SW_ORED<0, 9> xor .HW_SAVE<0, 9>));
11324 :      5914 DODU_FLG = ONE;
11325 :      5915 end;
11326 :      5916
11327 :      5917 end;
11328 :      5918
11329 :      5919 if .DODU_FLG IS_SET
11330 :      5920 then
11331 :      5921 begin
11332 :      5922 DODU (.ML_LUN);
11333 :      5923 DOCLN;
11334 :      5924 end;
11335 :      5925
11336 :      5926 ENDTST;

```

```

!SET PROM DISABLE MODE
!CALCULATE 128 SW ORED DATA & COMPARE TO HW TABLE
!LOADING MLPA INITIATES A PROM READ
!SAVE ROW DATA
!ENABLE COLUMN DATA ADRS
!SAVE COL DATA
!CALCULATE SOFTWARE ORED
!GET RESPECTIVE HARDWARE ORED
!COMPARE SW & HW ORED
!IF NEQ THEN ERROR
!DROP THIS UNIT IF DODU FLG IS_SET

```

```

11344 040650 004167 143204      $T22: JSR    R1,$$SAVES      ;      5841
11345 040654 162706 000010      SUB    #10,SP      ;
11346 040660 152777 000040 151052  BISB  #40,@ML.REG+40 ;      5879
11347 040666 016705 151276      MOV    ML,DUT,R5
11348 040672 042705 177770      BIC   #177770,R5
11349 040676 142777 000007 151034  BICB  #7,@ML.REG+40
11350 040704 150577 151030      BISB  R5,@ML.REG+40
11351 040710 005066 000004      CLR   4(SP)      ; DODU.FLG      5882
11352 040714 152777 000010 151076  BISB  #10,@ML.REG+120 ;      5883
11353 040722 004767 151572      JSR   PC,FIRST.BLK.XFER ;      5884
11354 040726 142777 000077 150744  BICB  #77,@ML.REG  ;      5885
11355 040734 152777 000061 150736  BISB  #61,@ML.REG
11356 :ML4
11357 :
11358 :
11359 040742 005002      CLR   R2      ; PROM.ADRS      5887
11360 040744 012701 000001 1$:  MOV   #1,R1      ; *,$$TMP2      5889
11361 040750 001411 2$:  BEQ   5$
11362 040752 016703 141140      MOV   L$DLY,R3 ; *,$$TMP1
11363 040756 001404      BEQ   4$
11364 040760 005066 000006 3$:  CLR   6(SP)      ; $$TMP

```

22-Oct-1980 10:47:44 TOPS
22-Oct-1980 10:45:32 PA:<

11365	040764	005303			DEC	R3		: \$STMP1	
11366	040766	001374			BNE	3\$			
11367	040770	005301			4\$: DEC	R1		: \$STMP2	
11368	040772	000766			BR	2\$			
11369	040774	010203			5\$: MOV	R2,R3		: PROM.ADRS,*	5890
11370	040776	006303			ASL	R3			
11371	041000	017763	151124	007722	MOV	@ML.REG+230,HW.OR.TBL(R3)			
11372	041006	152777	000020	151004	BISB	#20,@ML.REG+120			5891
11373	041014	005202			INC	R2		: PROM.ADRS	5887
11374	041016	020227	000177		CMP	R2,#177		: PROM.ADRS,*	
11375	041022	003750			BLE	1\$			
11376	041024	152777	000040	150706	BISB	#40,@ML.REG+40			5892
11377	041032	016705	151132		MOV	ML.DUT,R5			
11378	041036	042705	177770		BIC	#177770,R5			
11379	041042	142777	000007	150670	BICB	#7,@ML.REG+40			
11380	041050	150577	150664		BISB	R5,@ML.REG+40			
11381	041054	152777	000040	150736	BISB	#40,@ML.REG+120			5895
11382	041062	005001			CLR	R1		: PROM.ADRS	5897
11383	041064	010177	150710		6\$: MOV	R1,@ML.REG+100		: PROM.ADRS,*	5899
11384	041070	012702	000001		MOV	#1,R2		: *,\$STMP2	5900
11385	041074	001411			7\$: BEQ	10\$			
11386	041076	016703	141014		MOV	L\$DLY,R3		: *,\$STMP1	
11387	041102	001404			BEQ	9\$			
11388	041104	005066	000006		8\$: CLR	6(SP)		: \$STMP	
11389	041110	005303			DEC	R3		: \$STMP1	
11390	041112	001374			BNE	8\$			
11391	041114	005302			9\$: DEC	R2		: \$STMP2	
11392	041116	000766			BR	7\$			
11393	041120	017766	151004	000002	10\$: MOV	@ML.REG+230,2(SP)		: *,R.BITS	5901
11394	041126	010103			MOV	R1,R3		: PROM.ADRS,*	5902
11395	041130	052703	002000		BIS	#2000,R3			
11396	041134	010377	150640		MOV	R3,@ML.REG+100			
11397	041140	012702	000001		MOV	#1,R2		: *,\$STMP2	5903
11398	041144	001411			11\$: BEQ	14\$			
11399	041146	016703	140744		MOV	L\$DLY,R3		: *,\$STMP1	
11400	041152	001404			BEQ	13\$			
11401	041154	005066	000006		12\$: CLR	6(SP)		: \$STMP	
11402	041160	005303			DEC	R3		: \$STMP1	
11403	041162	001374			BNE	12\$			
11404	041164	005302			13\$: DEC	R2		: \$STMP2	
11405	041166	000766			BR	11\$			
11406	041170	017716	150734		14\$: MOV	@ML.REG+230,(SP)		: *,C.BITS	5904
11407	041174	016605	000002		MOV	2(SP),R5		: R.BITS,SW.ORED	5905
11408	041200	051605			BIS	(SP),R5		: C.BITS,SW.ORED	
11409	041202	010103			MOV	R1,R3		: PROM.ADRS,*	5906
11410	041204	006303			ASL	R3			
11411					:ML4				
11412					:				
11413									
11414	041206	016304	007722		MOV	HW.OR.TBL(R3),R4		: *,HW.SAVE	
11415	041212	010502			MOV	R5,R2		: SW.ORED,*	5908
11416	041214	042702	177000		BIC	#177000,R2			
11417	041220	010403			MOV	R4,R3		: HW.SAVE,*	
11418	041222	042703	177000		BIC	#177000,R3			
11419	041226	020302			CMP	R3,R2			
11420	041230	001450			BEQ	15\$			
11421	041232	104455			TRAP	55			5911

22-Oct-1980 10:47:44
22-Oct-1980 10:45:32

TOPS
PA:<

11422	041234	000114			.WORD	114			
11423	041236	007534			.WORD	ARR.DAT			
11424	041240	000000			.WORD	0			
11425	041242	012746	006630		MOV	#PHR.4, -(SP)	:		5912
11426	041246	012746	006040		MOV	#WRD.19, -(SP)			
11427	041252	012746	006226		MOV	#WRD.36, -(SP)			
11428	041256	012746	006220		MOV	#WRD.35, -(SP)			
11429	041262	012746	005400		MOV	#FOR.FMT, -(SP)			
11430	041266	012746	000005		MOV	#5, -(SP)			
11431	041272	010600			MOV	SP, R0	:	SP, *	
11432	041274	104414			TRAP	14			
11433	041276	010403			MOV	R4, R3	:	HW.SAVE, *	5913
11434	041300	010516			MOV	R5, (SP)	:	SW.ORED, *	
11435	041302	040502			BIC	R5, R2			
11436	041304	040216			BIC	R2, (SP)			
11437	041306	050216			BIS	R2, (SP)			
11438	041310	010446			MOV	R4, -(SP)	:	HW.SAVE, *	
11439	041312	042716	177000		BIC	#177000, (SP)			
11440	041316	010546			MOV	R5, -(SP)	:	SW.ORED, *	
11441	041320	042716	177000		BIC	#177000, (SP)			
11442	041324	012746	004224		MOV	#FMT.2, -(SP)			
11443	041330	012746	000004		MOV	#4, -(SP)			
11444	041334	010600			MOV	SP, R0	:	SP, *	
11445	041336	104414			TRAP	14			
11446	041340	012766	000001	000030	MOV	#1, 30(SP)	:	*, DODU.FLG	5914
11447	041346	062706	000024		ADD	#24, SP	:		5910
11448	041352	005201		15\$:	INC	R1	:	PROM.ADRS	5897
11449	041354	020127	000177		CMP	R1, #177	:	PROM.ADRS, *	
11450	041360	003641			BLE	6\$			
11451	041362	026627	000004	000001	CMP	4(SP), #1	:	DODU.FLG, *	5919
11452	041370	001004			BNE	16\$			
11453	041372	016700	150570		MOV	ML.LUN, R0	:		5922
11454	041376	104451			TRAP	51			
11455	041400	104444			TRAP	44			
11456	041402	062706	000010	16\$:	ADD	#10, SP	:		5841
11457	041406	000207			RTS	PC			
11458									
11459									
11460									
11471									
11475	041410				T22::				
11476	041410	004767	177234	1\$:	JSR	PC, \$T22	:		5924
11477	041414	104466			TRAP	66			
11478	041416	006000			ROR	R0			
11479	041420	103773			BLO	1\$			
11480	041422	000207			RTS	PC			

; Routine Size: 176 words
; Maximum stack depth per invocation: 20 words

22-Oct-1980 10:47:44
 22-Oct-1980 10:45:32

TOPS-20 Bliss-16 V2(206)
 PA:<NEALE>BL2ML4.BLI.2 (65)

11486 :ML4
 11487 :
 11488 :
 11489 :
 11490 :
 11491 :
 11492 :
 11493 :
 11494 :
 11495 :
 11496 :
 11497 :
 11498 :
 11499 :
 11500 :
 11501 :
 11502 :
 11503 :
 11504 :
 11505 :
 11506 :
 11507 :
 11508 :
 11509 :
 11510 :
 11511 :
 11512 :
 11513 :
 11514 :
 11515 :
 11516 :
 11517 :
 11518 :
 11519 :
 11520 :
 11521 :
 11522 :
 11523 :
 11524 :
 11525 :
 11526 :
 11527 :
 11528 :
 11529 :
 11530 :
 11531 :
 11532 :
 11533 :
 11534 :
 11535 :
 11536 :
 11537 :
 11538 :
 11539 :
 11540 :

5928
 5929
 5930
 5931
 5932
 5933
 5934
 5935
 5936
 5937
 5938
 5939
 5940
 5941
 5942
 5943
 5944
 5945
 5946
 5947
 5948
 5949
 5950
 5951
 5952
 5953
 5954
 5955
 5956
 5957
 5958
 5959
 5960
 5961
 5962
 5963
 5964
 5965
 5966
 5967
 5968
 5969
 5970
 5971
 5972
 5973
 5974
 5975
 5976
 5977
 5978
 5979

BGNTST;

!++

TEST NUMBER: TST 23

TEST NAME: UV ADRS ERROR TEST

TEST DESCRIPTION:

TEST THE DETECTION OF UV ADRS
 ERRORS BY:

1. GENERATING PROM DATA PATTERN
 FROM 0 TO %0'177777' AND
 DETERMINE WHETHER RESPECTIVE
 PATTERN IS GOOD/OR BAD
 PROM DATA.
2. VIA DAT DM AND PROM R/W
 MODES PRESENT GENERATED
 PROM DATA TO THE UV ADRS
 ERR PROM.
3. TEST ERROR CONDITIONS FOR
 CORRECT RESPONCE TO GOOD/
 OR BAD PROM DATA.

IMPLICIT INPUTS: NONE

!--

local

DODU_FLG,
 HIGH_CNT,
 PROM_DATA : bitvector [16],
 LOW_CNT,
 TEMP,
 ERR_FLG,
 GTR_FLG;

DODU_FLG = ZERO;
 PROM_DATA = -1;

do

begin
 PROM_DATA = .PROM_DATA + 1;
 BGNSUB;
 CLR_MBUS;
 ERR_FLG = ZERO;

!DROP UNIT FLAG
 !STORES PROM DATA CHECK SUM BITS
 !STORES PROM DATA
 !STORES SUM OF PROM DATA BITS <9:0>
 !TEMPORARY STORAGE
 !ERROR FLAG
 !SETS WHEN PROM DATA BIT 15 IS A ONE

!TEST ALL POSSIBLE PROM DATA COMBINATIONS

!INCREMENT PROM_DATA

11542 ;ML4

22-Oct-1980 10:47:44
22-Oct-1980 10:45:32

TOPS-20 Bliss-16 V2(206)
PA:<NEALE>BL2ML4.BLI.2 (65)

```

11543 :
11544 :
11545 :      5980      LOW CNT = ZEROES;
11546 :      5981      HIGH CNT = ZEROES;
11547 :      5982      GTR_FLG = ZERO;
11548 :      5983
11549 :      5984      incr BIT_CNT from 0 to 9 do          !COUNT PROM DATA BITS <9:0>
11550 :      5985
11551 :      5986          if .PROM_DATA [.BIT_CNT] IS_SET then LOW_CNT = .LOW_CNT + 1;
11552 :      5987
11553 :      5988      HIGH_CNT = .PROM_DATA<10, 3>;          !GET PROM DATA CHECK SUM BITS
11554 :      5989
11555 :      5990      if .PROM_DATA [13] IS_SET then HIGH_CNT = .HIGH_CNT + 1;      !ADD IN BIT 13 IF SET
11556 :      5991
11557 :      5992      if .PROM_DATA [14] IS_SET then HIGH_CNT = .HIGH_CNT + 1;      !ADD IN BIT 14 IF SET
11558 :      5993
11559 :      5994      if .PROM_DATA [15] IS_SET then HIGH_CNT = .HIGH_CNT + 1;      !ADD IN BIT 15 IF SET
11560 :      5995
11561 :      5996      if .PROM_DATA [15] IS_SET then GTR_FLG = ONE;          !SET FLAG IF BIT 15 IS SET
11562 :      5997
11563 :      5998      DAT_DM_XFER ();          !SET UP A DATA DIAG MODE XFERR
11564 :      5999      PROM_RW = ONE;          !SET PROM READ WRITE
11565 :      6000      MLPD = .PROM_DATA;          !LOAD MLPD WITH PROM_DATA
11566 :      6001      MLCS1 = write;          !DO A WRITE FUNCTION
11567 :      6002      DAT_CLK = ONE;          !CLOCK PROM DATA INTO UV PROM
11568 :      6003
11569 :      6004      if .UNS IS_SET          !SEE IF PROM DATA CAUSED A UV ERROR
11570 :      6005      then
11571 :      6006          begin
11572 :      6007
11573 :      6008          if .GTR_FLG IS_SET          !UNS IS_SET. SEE IF GTR FLG IS SET
11574 :      6009          then
11575 :      6010          begin
11576 :      6011          TEMP = .PROM_DATA;          !LOAD TEMP WITH PROM DATA
11577 :      6012          TEMP = .TEMP and %o'162000';          !SEE IF THESE BITS ARE SET IN PROM DATA
11578 :      6013
11579 :      6014          if .TEMP eql %o'162000'          !THESE BITS SET AUTOMATICALLY CAUSE A UNS
11580 :      6015          then
11581 :      6016          begin
11582 :      6017
11583 :      6018          if .LOW_CNT geq .HIGH_CNT          !LOW<9:0> SHOULD BE ISS THAN THE HIGH<15:10> IF GTR FLG IS S
11584 :      6019          then          !ERROR IF LOW<9:0> IS GEQ HIGH<15:10>
11585 :      6020          begin
11586 :      6021          ERRDF (72, ARR DAT, 0);
11587 :      6022          PRINTB (SIX FMT, WRD_34, PHR_5, WRD_32, WRD_6, WRD_33, WRD_24);
11588 :      6023          ERR_FLG = ONE;
11589 :      6024          end;
11590 :      6025
11591 :      6026          end;
11592 :      6027
11593 :      6028          end
11594 :      6029      else          !GTR FLG IS NOT SET
11595 :      6030          begin
11596 :      6031

```


22-Oct-1980 10:47:44 TOPS-20 Bliss-16 V2(206)
22-Oct-1980 10:45:32 PA:<NEALE>BL2ML4.BLI.2 (65)

```

11598 ;ML4
11599 :
11600 :
11601 : 6032 if .LOW_CNT eql .HIGH_CNT !LOW<9:0> SHOULD BE NEQ HIGH<15:10>
11602 : 6033 then
11603 : 6034 begin !ERROR IF EQL
11604 : 6035 ERRDF (73, ARR_DAT, 0);
11605 : 6036 PRINTB (SIX_FMT, WRD_34, PHR_5, WRD_32, WRD_6, WRD_33, WRD_24);
11606 : 6037 ERR_FLG = ONE;
11607 : 6038 end;
11608 : 6039
11609 : 6040 end
11610 : 6041
11611 : 6042 end
11612 : 6043 else !UNS WAS NOT SET
11613 : 6044 begin
11614 : 6045
11615 : 6046 if .GTR_FLG IS_SET !SEE IF GTR FLG IS_SET
11616 : 6047 then
11617 : 6048 begin
11618 : 6049 TEMP = .PROM_DATA; !LOAD TEMP WITH PROM DATA
11619 : 6050 TEMP = .TEMP and %o'162000'; !SEE IF THESE BITS ARE SET IN PROM DATA
11620 : 6051
11621 : 6052 if .TEMP neq %o'162000' !IF THESE BITS ARE SET THEN UNS SHOULD BE SET
11622 : 6053 then
11623 : 6054 begin !ERROR UNS IS NOT SET
11624 : 6055 ERRDF (74, ARR_DAT, 0);
11625 : 6056 PRINTB (SIX_FMT, WRD_34, PHR_1, WRD_32, WRD_5, WRD_33, WRD_24);
11626 : 6057 ERR_FLG = ONE;
11627 : 6058 end
11628 : 6059 else !BITS 162000 ARE NOT SET
11629 : 6060 begin
11630 : 6061
11631 : 6062 if .LOW_CNT lss .HIGH_CNT !LOW<9:0> SHOULD BE GEQ HIGH<15:10>
11632 : 6063 then
11633 : 6064 begin !ERROR IF LSS
11634 : 6065 ERRDF (75, ARR_DAT, 0);
11635 : 6066 PRINTB (SIX_FMT, WRD_34, PHR_1, WRD_32, WRD_5, WRD_33, WRD_24);
11636 : 6067 ERR_FLG = ONE;
11637 : 6068 end;
11638 : 6069
11639 : 6070 end
11640 : 6071
11641 : 6072 end
11642 : 6073 else !GTR_FLG IS NOT SET
11643 : 6074 begin
11644 : 6075
11645 : 6076 if .LOW_CNT neq .HIGH_CNT !LOW<9:0> SHOULD EQL HIGH<15:10>
11646 : 6077 then
11647 : 6078 begin !ERROR IF NEQ
11648 : 6079 ERRDF (108, ARR_DAT, 0);
11649 : 6080 PRINTB (SIX_FMT, WRD_34, PHR_1, WRD_32, WRD_5, WRD_33, WRD_24);
11650 : 6081 ERR_FLG = ONE;
11651 : 6082 end
11652 : 6083

```

22-Oct-1980 10:47:44 TOPS-20 Bliss-16 V2(206)
22-Oct-1980 10:45:32 PA:<NEALE>BL2ML4.BLI.2 (65)

```
11654 :ML4
11655 :
11656 :
11657 :      6084      end
11658 :      6085
11659 :      6086      end;
11660 :      6087
11661 :      6088      if .ERR_FLG IS_SET
11662 :      6089      then
11663 :      6090          begin
11664 :      6091              PRINTB (FMT 7, .PROM_DATA);
11665 :      6092              DODU_FLG = ONE;
11666 :      6093          end;
11667 :      6094
11668 :      6095      ENDSUB;
11669 :      6096      end
11670 :      6097      until .PROM_DATA eql %o'177777';
11671 :      6098
11672 :      6099      if .DODU_FLG IS_SET
11673 :      6100      then
11674 :      6101          begin
11675 :      6102              DODU (.ML_LUN);
11676 :      6103              DOCLN;
11677 :      6104          end;
11678 :      6105
11679 :      6106      ENDTST;
```

!SEE IF ERROR FLG GOT SET
!PRINT FAILING PROM_DATA AND SET DODU_FLG
!TRY ALL BIT COMBINATIONS
!DROP THIS UNIT IF DODU_FLG IS SET

```
11683
11687 041424 004167 142430      $T23: JSR      R1,$SAVE5      ;      5926
11688 041430 162706 000012      SUB      #12,SP
11689 041434 005066 000006      CLR      6(SP)      ; DODU.FLG      5971
11690 041440 012766 177777 000010      MOV      #-1,10(SP) ; *,PROM.DATA      5972
11691 041446 005266 000010      1$: INC      10(SP)      ; PROM.DATA      5976
11692 041452 104402      2$: TRAP     2
11693 041454 152777 000040 150256      BISB     #40,@ML.REG+40 ;      5977
11694 041462 016705 150502      MOV      ML.DUT,R5
11695 041466 042705 177770      BIC      #177770,R5
11696 041472 142777 000007 150240      BICB     #7,@ML.REG+40
11697 041500 150577 150234      BISB     R5,@ML.REG+40
11698 041504 005001      CLR      R1      ; ERR.FLG      5979
11699 041506 005066 000002      CLR      2(SP)      ; LOW.CNT      5980
11700 041512 005003      CLR      R3      ; HIGH.CNT      5981
11701 041514 005066 000004      CLR      4(SP)      ; GTR.FLG      5982
11702 041520 005005      CLR      R5      ; BIT.CNT      5984
11703 041522 010504      3$: MOV      R5,R4      ; BIT.CNT,*      5986
11704 041524 006204      ASR      R4
11705 041526 006204      ASR      R4
11706 041530 006204      ASR      R4
11707 041532 012702 000010      MOV      #10,R2
```

Address	OpCode	Operand 1	Operand 2	Operand 3	Operand 4	Instruction	Comments	Line No.
11709								
11710								
11711								
11712	041536	060602				ADD SP,R2	; PROM.DATA,*	
11713	041540	060204				ADD R2,R4		
11714	041542	010446				MOV R4,-(SP)		
11715	041544	010546				MOV R5,-(SP)	; BIT.CNT,*	
11716	041546	042716	177770			BIC #177770,(SP)		
11717	041552	012746	000001			MOV #1,-(SP)		
11718	041556	005046				CLR -(SP)		
11719	041560	004767	141316			JSR PC,BL\$GT2		
11720	041564	062706	000010			ADD #10,SP		
11721	041570	005300				DEC R0		
11722	041572	001002				BNE 4\$		
11723	041574	005266	000002			INC 2(SP)	; LOW.CNT	
11724	041600	005205		4\$:		INC R5	; BIT.CNT	5984
11725	041602	020527	000011			CMP R5,#11	; BIT.CNT,*	
11726	041606	003745				BLE 3\$		
11727	041610	016603	000010			MOV 10(SP),R3	; PROM.DATA,HIGH.CNT	5988
11728	041614	006203				ASR R3	; HIGH.CNT	
11729	041616	006203				ASR R3	; HIGH.CNT	
11730	041620	000303				SWAB R3	; HIGH.CNT	
11731	041622	042703	177770			BIC #177770,R3	; *,HIGH.CNT	
11732	041626	132766	000040	000011		BITB #40,11(SP)	; *,PROM.DATA+1	5990
11733	041634	001401				BEQ 5\$		
11734	041636	005203				INC R3	; HIGH.CNT	
11735	041640	132766	000100	000011	5\$:	BITB #100,11(SP)	; *,PROM.DATA+1	5992
11736	041646	001401				BEQ 6\$		
11737	041650	005203				INC R3	; HIGH.CNT	
11738	041652	005005		6\$:		CLR R5		5994
11739	041654	105766	000011			TSTB 11(SP)	; PROM.DATA+1	
11740	041660	100002				BPL 7\$		
11741	041662	005205				INC R5		
11742	041664	005203				INC R3	; HIGH.CNT	
11743	041666	006005		7\$:		ROR R5		5996
11744	041670	103003				BCC 8\$		
11745	041672	012766	000001	000004		MOV #1,4(SP)	; *,GTR.FLG	
11746	041700	004767	150722		8\$:	JSR PC,DAT.DM.XFER		5998
11747	041704	152777	000100	150106		BISB #100,@ML.REG+120		5999
11748	041712	016605	000010			MOV 10(SP),R5	; PROM.DATA,*	6000
11749	041716	010577	150206			MOV R5,@ML.REG+230		
11750	041722	012777	000061	147750		MOV #61,@ML.REG		6001
11751	041730	152777	000020	150062		BISB #20,@ML.REG+120		6002
11752	041736	032777	040000	150014		BIT #40000,@ML.REG+60		6004
11753	041744	001500				BEQ 12\$		
11754	041746	026627	000004	000001		CMP 4(SP),#1	; GTR.FLG,*	6008
11755	041754	001042				BNE 11\$		
11756	041756	010516				MOV R5,(SP)	; *,TEMP	6011
11757	041760	042716	015777			BIC #15777,(SP)	; *,TEMP	6012
11758	041764	021627	162000			CMP (SP),#-16000	; TEMP,*	6014
11759	041770	001003				BNE 9\$		
11760	041772	026603	000002			CMP 2(SP),R3	; LOW.CNT,HIGH.CNT	6018
11761	041776	002002				BGE 10\$		
11762	042000	000167	000422		9\$:	JMP 16\$		
11763	042004	104455		10\$:		TRAP 55		6021

22-Oct-1980 10:47:44 TOPS
22-Oct-1980 10:45:32 PA:<

```
11765          :ML4
11766          :
11767          :
11768 042006 000110      .WORD 110
11769 042010 007534      .WORD ARR.DAT
11770 042012 000000      .WORD 0
11771 042014 012746 006104  MOV #WRD.24,-(SP) ;
11772 042020 012746 006206  MOV #WRD.33,-(SP) ;
11773 042024 012746 005670  MOV #WRD.6,-(SP) ;
11774 042030 012746 006200  MOV #WRD.32,-(SP) ;
11775 042034 012746 006646  MOV #PHR.5,-(SP) ;
11776 042040 012746 006212  MOV #WRD.34,-(SP) ;
11777 042044 012746 005432  MOV #SIX.FMT,-(SP) ;
11778 042050 012746 000007  MOV #7,-(SP) ;
11779 042054 010600      MOV SP,R0 ; SP,*
11780 042056 104414      TRAP 14 ;
11781 042060 000556      BR 15$ ;
11782 042062 026603 000002 11$: CMP 2(SP),R3 ; LOW.CNT,HIGH.CNT
11783 042066 001157      BNE 16$ ;
11784 042070 104455      TRAP 55 ;
11785 042072 000111      .WORD 111 ;
11786 042074 007534      .WORD ARR.DAT ;
11787 042076 000000      .WORD 0 ;
11788 042100 012746 006104  MOV #WRD.24,-(SP) ;
11789 042104 012746 006206  MOV #WRD.33,-(SP) ;
11790 042110 012746 005670  MOV #WRD.6,-(SP) ;
11791 042114 012746 006200  MOV #WRD.32,-(SP) ;
11792 042120 012746 006646  MOV #PHR.5,-(SP) ;
11793 042124 012746 006212  MOV #WRD.34,-(SP) ;
11794 042130 012746 005432  MOV #SIX.FMT,-(SP) ;
11795 042134 012746 000007  MOV #7,-(SP) ;
11796 042140 010600      MOV SP,R0 ; SP,*
11797 042142 104414      TRAP 14 ;
11798 042144 000524      BR 15$ ;
11799 042146 026627 000004 000001 12$: CMP 4(SP),#1 ; GTR.FLG,*
11800 042154 001067      BNE 14$ ;
11801 042156 010516      MOV R5,(SP) ; *,TEMP
11802 042160 042716 015777  BIC #15777,(SP) ; *,TEMP
11803 042164 021627 162000  CMP (SP),#-16000 ; TEMP,*
11804 042170 001427      BEQ 13$ ;
11805 042172 104455      TRAP 55 ;
11806 042174 000112      .WORD 112 ;
11807 042176 007534      .WORD ARR.DAT ;
11808 042200 000000      .WORD 0 ;
11809 042202 012746 006104  MOV #WRD.24,-(SP) ;
11810 042206 012746 006206  MOV #WRD.33,-(SP) ;
11811 042212 012746 005662  MOV #WRD.5,-(SP) ;
11812 042216 012746 006200  MOV #WRD.32,-(SP) ;
11813 042222 012746 006542  MOV #PHR.1,-(SP) ;
11814 042226 012746 006212  MOV #WRD.34,-(SP) ;
11815 042232 012746 005432  MOV #SIX.FMT,-(SP) ;
11816 042236 012746 000007  MOV #7,-(SP) ;
11817 042242 010600      MOV SP,R0 ; SP,*
11818 042244 104414      TRAP 14 ;
11819 042246 000463      BR 15$ ;
```

6022

6023

6032

6035

6036

6037

6046

6049

6050

6052

6055

6056

6057

Line No	Address	Op Code	Op Data	Op Comment	Seq No
11821				:ML4	
11822				:	
11823				:	
11824	042250	026603	000002	13\$: CMP 2(SP),R3 ; LOW.CNT,HIGH.CNT	6062
11825	042254	002064		BGE 16\$	
11826	042256	104455		TRAP 55 ;	6065
11827	042260	000113		.WORD 113	
11828	042262	007534		.WORD ARR.DAT	
11829	042264	000000		.WORD 0	
11830	042266	012746	006104	MOV #WRD.24,-(SP) ;	6066
11831	042272	012746	006206	MOV #WRD.33,-(SP)	
11832	042276	012746	005662	MOV #WRD.5,-(SP)	
11833	042302	012746	006200	MOV #WRD.32,-(SP)	
11834	042306	012746	006542	MOV #PHR.1,-(SP)	
11835	042312	012746	006212	MOV #WRD.34,-(SP)	
11836	042316	012746	005432	MOV #SIX.FMT,-(SP)	
11837	042322	012746	000007	MOV #7,-(SP)	
11838	042326	010600		MOV SP,R0 ; SP,*	
11839	042330	104414		TRAP 14	
11840	042332	000431		BR 15\$;	6067
11841	042334	026603	000002	14\$: CMP 2(SP),R3 ; LOW.CNT,HIGH.CNT	6076
11842	042340	001432		BEQ 16\$	
11843	042342	104455		TRAP 55 ;	6079
11844	042344	000154		.WORD 154	
11845	042346	007534		.WORD ARR.DAT	
11846	042350	000000		.WORD 0	
11847	042352	012746	006104	MOV #WRD.24,-(SP) ;	6080
11848	042356	012746	006206	MOV #WRD.33,-(SP)	
11849	042362	012746	005662	MOV #WRD.5,-(SP)	
11850	042366	012746	006200	MOV #WRD.32,-(SP)	
11851	042372	012746	006542	MOV #PHR.1,-(SP)	
11852	042376	012746	006212	MOV #WRD.34,-(SP)	
11853	042402	012746	005432	MOV #SIX.FMT,-(SP)	
11854	042406	012746	000007	MOV #7,-(SP)	
11855	042412	010600		MOV SP,R0 ; SP,*	
11856	042414	104414		TRAP 14	
11857	042416	012701	000001	15\$: MOV #1,R1 ; *,ERR.FLG	6081
11858	042422	062706	000020	ADD #20,SP ;	6078
11859	042426	020127	000001	16\$: CMP R1,#1 ; ERR.FLG,*	6088
11860	042432	001014		BNE 17\$	
11861	042434	010546		MOV R5,-(SP) ;	6091
11862	042436	012746	004520	MOV #FMT.7,-(SP)	
11863	042442	012746	000002	MOV #2,-(SP)	
11864	042446	010600		MOV SP,R0 ; SP,*	
11865	042450	104414		TRAP 14	
11866	042452	012766	000001 000014	MOV #1,14(SP) ; *,DODU.FLG	6092
11867	042460	062706	000006	ADD #6,SP ;	6090
11868	042464	104467		17\$: TRAP 67 ;	6093
11869	042466	006000		ROR R0	
11870	042470	103002		BHIS 18\$	
11871	042472	000167	176754	JMP 2\$	
11872	042476	005205		18\$: INC R5 ;	6097
11873	042500	001402		BEQ 19\$	
11874	042502	000167	176740	JMP 1\$	
11875	042506	026627	000006 000001	19\$: CMP 6(SP),#1 ; DODU.FLG,*	6099

```
11877 ;ML4
11878 ;
11879 ;
11880 042514 001004 BNE 20$
11881 042516 016700 147444 MOV ML.LUN,R0 ;
11882 042522 104451 TRAP 51 ;
11883 042524 104444 TRAP 44 ;
11884 042526 062706 000012 20$: ADD #12,SP ;
11885 042532 000207 RTS PC ;
11886
11887 ; Routine Size: 292 words
11888 ; Maximum stack depth per invocation: 19 words
11893
11894
11898
11902 042534 T23::
11903 042534 004767 176664 1$: JSR PC,$T23 ;
11904 042540 104466 TRAP 66 ;
11905 042542 006000 ROR R0 ;
11906 042544 103773 BLO 1$ ;
11907 042546 000207 RTS PC ;
11908
11909 ; Routine Size: 6 words
11910 ; Maximum stack depth per invocation: 0 words
11915
11916
11917 ; 6107 !<BLF/PAGE>
```

22-Oct-1980 10:47:44 TOPS
22-Oct-1980 10:45:32 PA:<

6102

5926

6104

22-Oct-1980 10:47:44
 22-Oct-1980 10:45:32

TOPS-20 Bliss-16 V2(206)
 PA:<NEALE>BL2ML4.BLI.2 (66)

11919 :ML4

11920 :
 11921 :
 11922 : 6108
 11923 : 6109
 11924 : 6110
 11925 : 6111
 11926 : 6112
 11927 : 6113
 11928 : 6114
 11929 : 6115
 11930 : 6116
 11931 : 6117
 11932 : 6118
 11933 : 6119
 11934 : 6120
 11935 : 6121
 11936 : 6122
 11937 : 6123
 11938 : 6124
 11939 : 6125
 11940 : 6126
 11941 : 6127
 11942 : 6128
 11943 : 6129
 11944 : 6130
 11945 : 6131
 11946 : 6132
 11947 : 6133
 11948 : 6134
 11949 : 6135
 11950 : 6136
 11951 : 6137
 11952 : 6138
 11953 : 6139
 11954 : 6140
 11955 : 6141
 11956 : 6142
 11957 : 6143
 11958 : 6144
 11959 : 6145
 11960 : 6146
 11961 : 6147
 11962 : 6148
 11963 : 6149
 11964 : 6150
 11965 : 6151
 11966 : 6152
 11967 : 6153
 11968 : 6154
 11969 : 6155
 11970 : 6156
 11971 : 6157
 11972 : 6158
 11973 : 6159

BGNTST;

!++

TEST NUMBER: TST 24

TEST NAME: INITIAL ARRAY TEST

TEST DESCRIPTION:

DUE TO THE NATURE OF THE DEVICE
 THERE EXISTS KNOWN BAD ARRAY
 DATA LOCATIONS.

THEREFORE TO INITIALLY TEST THE ARRAYS'
 TIMING AND CONTROL LOGIC A BAD
 NIBBLE THRESHOLD OF 36 BAD NIBBLES
 OUT OF 100 NIBBLES TESTED WILL BE
 TOLERATED BEFORE DETERMINING CONTROL
 LOGIC TO BE IN ERROR.

THE ARRAYS' ARE INITIALLY TESTED BY:

1. VIA DAT DM MOD WRITE DATA PATTERNS
 OF 1'S AND 0'S TO 5 ARRAY
 WORDS.
2. TEST EACH NIBBLE (4 BITS) FOR
 1'S AND 0'S AND COUNT EACH BAD
 NIBBLE ENCOUNTERED.
3. IF ACCUMULATED BAD NIBBLES
 EXCEED 36 THEN REPORT AN ERROR.

IMPLICIT INPUTS: NONE

--

local

TST_PAT,
 BAD_NIB_CNT,
 ERR_FLG;

TST_PAT = ONES;
 BAD_NIB_CNT = ZERGES;

incr TWICE from 0 to 1 do

begin
 BGNSUB;
 CLR MBUS;
 MLDT = .TST_PAT;

!TEST PATTERN
 !NUMBER OF BAD NIBBLES FOUND
 !ERROR FLAG

!REPEAT LOOP TWICE

!LOAD TEST PATTERN INTO DIAG REGISTERS

22-Oct-1980 10:47:44 TOPS-20 Bliss-16 v2(206)
22-Oct-1980 10:45:32 PA:<NEALE>BL2ML4.BLI.2 (66)

```

11975 :ML4
11976 :
11977 :
11978 :      6160      MLD2 = .TST_PAT;
11979 :      6161      MLE2 = .TST_PAT;
11980 :      6162      DAT_DM = ONE;
11981 :      6163      FIRST_BLK_XFER ();
11982 :      6164      MLCS1 = write;
11983 :      6165
11984 :      6166      incr CNT from 0 to 4 do
11985 :      6167          begin
11986 :      6168          DELAY (ONE_US);
11987 :      6169          DAT_CLK = ONE;
11988 :      6170          end;
11989 :      6171
11990 :      6172      CLR_MBUS;
11991 :      6173      DAT_DM = ONE;
11992 :      6174      FIRST_BLK_XFER ();
11993 :      6175      MLCS1 = read;
11994 :      6176      DELAY (ONE_US);
11995 :      6177
11996 :      6178      incr ARR_WRD from 0 to 4 do
11997 :      6179          begin
11998 :      6180          DAT_CLK = ONE;
11999 :      6181          DELAY (ONE_US);
12000 :      6182          RD_LNG_WRD;
12001 :      6183
12002 :      6184          incr NIB_PTR from 0 to 9 do
12003 :      6185              begin
12004 :      6186                  TST_LNG_WRD (.NIB_PTR, .TST_PAT, ERR_FLG); !COMPARE TST PAT TO NIBBLE UNDER TEST
12005 :      6187
12006 :      6188                  if .ERR_FLG IS_SET then BAD_NIB_CNT = .BAD_NIB_CNT + 1;
12007 :      6189
12008 :      6190
12009 :      6191                  end;
12010 :      6192
12011 :      6193              end;
12012 :      6194
12013 :      6195      TST_PAT = not .TST_PAT;
12014 :      6196      ENDSUB;
12015 :      6197      end;
12016 :      6198
12017 :      6199      if .BAD_NIB_CNT gtr 36
12018 :      6200      then
12019 :      6201          begin
12020 :      6202          ERRDF (77, ASYNC, 0);
12021 :      6203          PRINTB (FIV_FMT, WRD_22, PHR_4, WRD_12, WRD_45, FNC_14);
12022 :      6204          DODU (.ML_LUN);
12023 :      6205          DOCLN;
12024 :      6206          end;
12025 :      6207
12026 :      6208      ENDTST;

```

!SET DATA DIAG MODE
!SET UP A FIRST BLK XFERR
!DO A MBUS WRITE FUNCTION

!CLOCK 5 WORDS INTO MEMORY

!SET DATA DIAG MODE
!SET UP A FIRST BLK XFERR
!DO A READ FUNCTION

!READ THE 5 WORD IN MEMORY

!CLOCK OUT A WORD INTO THE DIAG REGS

!READ THE DIAG REGISTERS

!READ THE 10 NIBBLES IN THE WORD

!COMPARE TST PAT TO NIBBLE UNDER TEST

!INCREMENT BAD_NIBBLE COUNT IFERR_FLG SET

!REPEAT WITH COMPLIMENT TST PAT

!SEE IF 36 OUT OF THE 100 XFERRED WHERE BAD

!ERROR IF GTR 36


```

12031 ;ML4
12032 ;
12033 ;
12034 ;
12038 042550 004167 141304 $T24: JSR R1,$SAVE5 ; 6106
12039 042554 024646 CMP -(SP),-(SP) ;
12040 042556 012701 177777 MOV #-1,R1 ; *,TST.PAT 6152
12041 042562 005046 CLR -(SP) ; BAD.NIB.CNT 6153
12042 042564 005005 CLR R5 ; TWICE 6155
12043 042566 104402 1$: TRAP 2 ; 6156
12044 042570 152777 000040 147142 BISB #40,@ML.REG+40 ; 6157
12045 042576 016704 147366 MOV ML.DUT,R4 ;
12046 042602 042704 177770 BIC #177770,R4 ;
12047 042606 142777 000007 147124 BICB #7,@ML.REG+40 ;
12048 042614 150477 147120 BISB R4,@ML.REG+40 ;
12049 042620 010177 147244 MOV R1,@ML.REG+170 ; TST.PAT,* 6159
12050 042624 010177 147250 MOV R1,@ML.REG+200 ; TST.PAT,* 6160
12051 042630 010177 147224 MOV R1,@ML.REG+160 ; TST.PAT,* 6161
12052 042634 152777 000010 147156 BISB #10,@ML.REG+120 ; 6162
12053 042642 004767 147652 JSR PC,FIRST.BLK.XFER ; 6163
12054 042646 012777 000061 147024 MOV #61,@ML.REG ; 6164
12055 042654 005002 CLR R2 ; CNT 6166
12056 042656 012703 000001 2$: MOV #1,R3 ; *,$STMP2 6168
12057 042662 001411 3$: BEQ 6$ ;
12058 042664 016704 137226 MOV LSDLY,R4 ; *,$STMP1
12059 042670 001404 BEQ 5$ ;
12060 042672 005066 000004 4$: CLR 4(SP) ; $STMP
12061 042676 005304 DEC R4 ; $STMP1
12062 042700 001374 BNE 4$ ;
12063 042702 005303 5$: DEC R3 ; $STMP2
12064 042704 000766 BR 3$ ;
12065 042706 152777 000020 147104 6$: BISB #20,@ML.REG+120 ; 6169
12066 042714 005202 INC R2 ; CNT 6166
12067 042716 020227 000004 CMP R2,#4 ; CNT,*
12068 042722 003755 BLE 2$ ;
12069 042724 152777 000040 147006 BISB #40,@ML.REG+40 ; 6170
12070 042732 016704 147232 MOV ML.DUT,R4 ;
12071 042736 042704 177770 BIC #177770,R4 ;
12072 042742 142777 000007 146770 BICB #7,@ML.REG+40 ;
12073 042750 150477 146764 BISB R4,@ML.REG+40 ;
12074 042754 152777 000010 147036 BISB #10,@ML.REG+120 ; 6173
12075 042762 004767 147532 JSR PC,FIRST.BLK.XFER ; 6174
12076 042766 012777 000071 146704 MOV #71,@ML.REG ; 6175
12077 042774 012703 000001 7$: MOV #1,R3 ; *,$STMP2 6176
12078 043000 001411 BEQ 10$ ;
12079 043002 016704 137110 MOV LSDLY,R4 ; *,$STMP1
12080 043006 001404 BEQ 9$ ;
12081 043010 005066 000004 8$: CLR 4(SP) ; $STMP
12082 043014 005304 DEC R4 ; $STMP1
12083 043016 001374 BNE 8$ ;
12084 043020 005303 9$: DEC R3 ; $STMP2

```

12086						:ML4				22-Oct-1980 10:47:44	TOPS
12087						:				22-Oct-1980 10:45:32	PA:<
12088											
12089	043022	000766					BR	7\$			
12090	043024	005002				10\$:	CLR	R2		; ARR.WRD	6178
12091	043026	152777	000020	146764		11\$:	BISB	#20,@ML.REG+120			6180
12092	043034	012703	000001				MOV	#1,R3		; *,\$\$TMP2	6181
12093	043040	001411				12\$:	BEQ	15\$			
12094	043042	016704	137050				MOV	L\$DLY,R4		; *,\$\$TMP1	
12095	043046	001404					BEQ	14\$			
12096	043050	005066	000004			13\$:	CLR	4(SP)		; \$\$TMP	
12097	043054	005304					DEC	R4		; \$\$TMP1	
12098	043056	001374					BNE	13\$			
12099	043060	005303				14\$:	DEC	R3		; \$\$TMP2	
12100	043062	000766					BR	12\$			
12101	043064	017767	147000	144622		15\$:	MOV	@ML.REG+170,D1.TEMP			
12102	043072	017767	147002	144616			MOV	@ML.REG+200,D2.TEMP			
12103	043100	017767	146754	144612			MOV	@ML.REG+160,E2.TEMP			
12104	043106	005004					CLR	R4		; NIB.PTR	6184
12105	043110	010446				16\$:	MOV	R4,-(SP)		; NIB.PTR,*	6186
12106	043112	010146					MOV	R1,-(SP)		; TST.PAT,*	
12107	043114	012746	000010				MOV	#10,-(SP)			
12108	043120	060616					ADD	SP,(SP)		; ERR.FLG,*	
12109	043122	004767	147532				JSR	PC,TST.LNG.WRD			
12110	043126	026627	000010	000001			CMP	10(SP),#1		; ERR.FLG,*	6188
12111	043134	001002					BNE	17\$			
12112	043136	005266	000006				INC	6(SP)		; BAD.NIB.CNT	
12113	043142	062706	000006			17\$:	ADD	#6,SP			6185
12114	043146	005204					INC	R4		; NIB.PTR	6184
12115	043150	020427	000011				CMP	R4,#11		; NIB.PTR,*	
12116	043154	003755					BLE	16\$			
12117	043156	005202					INC	R2		; ARR.WRD	6178
12118	043160	020227	000004				CMP	R2,#4		; ARR.WRD,*	
12119	043164	003720					BLE	11\$			
12120	043166	005101					COM	R1		; TST.PAT	6195
12121	043170	104467					TRAP	67			
12122	043172	006000					ROR	R0			
12123	043174	103002					BHIS	19\$			
12124	043176	000167	177364			18\$:	JMP	1\$			
12125	043202	005205				19\$:	INC	R5		; TWICE	6155
12126	043204	020527	000001				CMP	R5,#1		; TWICE,*	
12127	043210	003772					BLE	18\$			
12128	043212	021627	000044				CMP	(SP),#44		; BAD.NIB.CNT,*	6199
12129	043216	003432					BLE	20\$			
12130	043220	104455					TRAP	55			6202
12131	043222	000115					.WORD	115			
12132	043224	007444					.WORD	ASYNC			
12133	043226	000000					.WORD	0			
12134	043230	012746	007146				MOV	#FNC.14,-(SP)			6203
12135	043234	012746	006324				MOV	#WRD.45,-(SP)			
12136	043240	012746	005760				MOV	#WRD.12,-(SP)			
12137	043244	012746	006630				MOV	#PHR.4,-(SP)			
12138	043250	012746	006062				MOV	#WRD.22,-(SP)			
12139	043254	012746	005414				MOV	#FIV.FMT,-(SP)			
12140	043260	012746	000006				MOV	#6,-(SP)			

```
12142      ;ML4
12143      ;
12144
12145 043264 010600      MOV      SP,R0      ; SP,*
12146 043266 104414      TRAP     14
12147 043270 016700 146672  MOV     ML.LUN,R0      ;
12148 043274 104451      TRAP     51
12149 043276 104444      TRAP     44
12150 043300 062706 000016  ADD     #16,SP      ;
12151 043304 062706 000006 20$:    ADD     #6,SP      ;
12152 043310 000207      RTS      PC
12153
12154      ; Routine Size: 177 words
12155      ; Maximum stack depth per invocation: 16 words
12160
12161
12165
12169 043312      T24::
12170 043312 004767 177232 1$:     JSR     PC,$T24      ;
12171 043316 104466      TRAP     66
12172 043320 006000      ROR     R0
12173 043322 103773      BLO     1$
12174 043324 000207      RTS      PC
12175
12176      ; Routine Size: 6 words
12177      ; Maximum stack depth per invocation: 0 words
12182
12183
12184 ;      6209 !<BLF/PAGE>
```

22-Oct-1980 10:47:44
22-Oct-1980 10:45:32

TOPS-20 Bliss-16 V2(206)
PA:<NEALE>BL2ML4.BLI.2 (67)

```

12186 :ML4
12187 :
12188 :
12189 :      6210 !
12190 :      6211 !BGNTST;
12191 :      6212
12192 :      6213 !++
12193 :      6214 !TEST NUMBER: TST 25
12194 :      6215
12195 :      6216 !TEST NAME: PROM SELECTION TEST
12196 :      6217
12197 :      6218 !TEST DESCRIPTION:
12198 :      6219 !DUE TO THE NATURE OF THE DEVICE
12199 :      6220 !AND OF THE ARRAY MODULES' UV
12200 :      6221 !PROMS, ONLY PROM READS ARE
12201 :      6222 !ALLOWED DURING DIAG TESTING.
12202 :      6223
12203 :      6224 !THEREFORE THE ARRAY MODULE UV PROMS
12204 :      6225 !ARE TESTED FOR UNIQUE SELECTION BY:
12205 :      6226
12206 :      6227 !1. AT EACH PRESENT ARRAY MODULE WRITE 127 ARRAY WORDS WITH 1'S/0'S PATTERN.
12207 :      6228
12208 :      6229 !2. READ THE UV PROMS AT THEIR RESPECTIVE ARRAY WORD LOCATION AND SEE IF
12209 :      6230 !THE PROMS MASK BAD NIBBLE LOCATIONS (ENCOUNTERED BAD NIBBLES
12210 :      6231 !INDICATES INCORRECT MASKING). COUNT EACH BAD NIBBLE ENCOUNTERED AT AN
12211 :      6232 !ARRAY MODULE.
12212 :      6233
12213 :      6234 !3. ALLOW A THRESHOLD OF 5 BAD NIBBLES AT ANY ARRAY MODULE.
12214 :      6235
12215 :      6236 !4. REPORT PROM SEL ERRORS AT RESPECTIVE ARRAY MODULE IF THE
12216 :      6237 !THRESHOLD IS EXCEEDED.
12217 :      6238
12218 :      6239 !IMPLICIT INPUTS:
12219 :      6240 !PD TEMP:
12220 :      6241 !A BIT VECTOR OF 16 BITS WHERE
12221 :      6242 !THE READ PROM DATA IS STORED
12222 :      6243 !AND ACCESSED FROM.
12223 :      6244
12224 :      6245 !IO BUF
12225 :      6246 !A VECTOR OF 256 WORDS WHERE
12226 :      6247 !DATA FOR MBUS READS AND WRITE
12227 :      6248 !FUNCTION ARE FOUND.
12228 :      6249
12229 :      6250 !--
12230 :      6251
12231 :      6252 !local
12232 :      6253 !DODU_FLG, !DROP UNIT FLG
12233 :      6254 !ERR_FLG, !ERROR FLG
12234 :      6255 !TST_PAT, !TEST PATTERN
12235 :      6256 !ERR_CNT; !ERROR COUNT
12236 :      6257
12237 :      6258 !DODU_FLG = ZERO;
12238 :      6259 !TST_PAT = ONES;
12239 :      6260
12240 :      6261 !incr ARR_SEL from 0 to .LST_ARR by .ARR_INC do !TEST ALL PRESENT ARRAYS

```

22-Oct-1980 10:47:44 TOPS-20 Bliss-16 V2(206)
22-Oct-1980 10:45:32 PA:<NEALE>BL2ML4.BLI.2 (67)

```

12242 :ML4
12243 :
12244 :
12245 :      6262      begin
12246 :      6263      ERR_CNT = ZERO;
12247 :      6264      BGNSUB;
12248 :      6265
12249 :      6266      incr TWICE from 0 to 1 do
12250 :      6267      begin
12251 :      6268      CLR_MBUS;
12252 :      6269      MLD1 = .TST_PAT;
12253 :      6270      MLD2 = .TST_PAT;
12254 :      6271      MLE2 = .TST_PAT;
12255 :      6272      DAT_DM = ONE;
12256 :      6273      MLWC = not 255;
12257 :      6274      MLBA = IO BUF;
12258 :      6275      MLDA = .ARR_SEL;
12259 :      6276      MLCS1 = write;
12260 :      6277
12261 :      6278      incr CNT from 0 to 127 do
12262 :      6279      begin
12263 :      6280      DELAY (ONE_US);
12264 :      6281      DAT_CLK = ONE;
12265 :      6282      end;
12266 :      6283
12267 :      6284      CLR_MBUS;
12268 :      6285      DAT_DM = ONE;
12269 :      6286      MLWC = not 255;
12270 :      6287      MLBA = IO BUF;
12271 :      6288      MLDA = .ARR_SEL;
12272 :      6289      MLCS1 = read;
12273 :      6290      DELAY (ONE_US);
12274 :      6291
12275 :      6292      incr WD_CNT from 0 to 127 do
12276 :      6293      begin
12277 :      6294      PD_TEMP = .MLPD;
12278 :      6295      DAT_CLK = ONE;
12279 :      6296      DELAY (ONE_US);
12280 :      6297      RD_LNG_WRD;
12281 :      6298
12282 :      6299      incr NIB_PTR from 0 to 9 do
12283 :      6300
12284 :      6301      if .PD_TEMP [.NIB_PTR] IS_NOT_SET      !FIND GOOD NIBBLES
12285 :      6302      then
12286 :      6303      begin
12287 :      6304      TST_LNG_WRD (.NIB_PTR, .TST_PAT, ERR_FLG); !COMPARE NIBBLE TO TST_PAT
12288 :      6305
12289 :      6306      if .ERR_FLG IS_SET then ERR_CNT = .ERR_CNT + 1;
12290 :      6307
12291 :      6308      !INCREMENT ERROR COUNT IF ERROR FLG IS SET
12292 :      6309      end;
12293 :      6310
12294 :      6311      end;
12295 :      6312
12296 :      6313      TST_PAT = not .TST_PAT;

```

!REPEAT LOOP TWICE

!LOAD DATA DIAG REGISTERS WITH TST_PAT

!SET DATA DIAG MODE

!LOAD WORD COUNT

!LOAD UBUS ADRS

!LOAD SECTOR

!DO A WRITE FUNCTION

!CLOCK IN 10 WORDS

!SET DATA DIAG MODE

!LOAD WORD COUNT

!LOAD UBUS ADRS

!LOAD SECTOR

!DO A READ FUNCTION

!READ THE 10 WORDS

!GET PROM DATA FOR THIS WORD

!CLOCK THIS WORD INTO DIAG REG

!READ DIAG REG FOR THIS WORD

!LOOK AT ALL 10 NIBBLE

!INCREMENT ERROR COUNT IF ERROR FLG IS SET

!REPEAT WITH COMPLIMENT DATA

22-Oct-1980 10:47:44 TOPS-20 Bliss-16 V2(206)
22-Oct-1980 10:45:32 PA:<NEALE>BL2ML4.BLI.2 (67)

```

12298 :ML4
12299 :
12300 :
12301 :      6314      end;
12302 :      6315
12303 :      6316      if .ERR_CNT gtr 5      !ALLOW 5 ERROR BEFORE ERRORING
12304 :      6317      then
12305 :      6318      begin      !ERROR IF GTR 5
12306 :      6319      ERRDF (78, ARR_DAT, 0);
12307 :      6320      PRINTB (THR_FMT, WRD_35, WRD_37, WRD_10);
12308 :      6321      PRINTB (FMT_9, .ARR_SEL);
12309 :      6322      DODU_FLG = ONE;
12310 :      6323      end;
12311 :      6324
12312 :      6325      ENDSUB;
12313 :      6326
12314 :      6327      if .DODU_FLG IS_SET      !DROP THIS UNIT IF DODU_FLG IS SET
12315 :      6328      then
12316 :      6329      begin
12317 :      6330      DODU (.ML_LUN);
12318 :      6331      DOCLN;
12319 :      6332      end;
12320 :      6333
12321 :      6334      end;
12322 :      6335
12323 :      6336      ENDTST;
12327 :
12331 043326 004167 140526      $T25: JSR      R1,$SAVES      ;      6208
12332 043332 162706 000016      SUB      #16,SP      ;
12333 043336 005066 000010      CLR      10(SP)      ; DODU.FLG      6258
12334 043342 012702 177777      MOV      #-1,R2      ; *,TST.PAT      6259
12335 043346 016766 144766 000006      MOV      LST.ARR,6(SP)      ;      6261
12336 043354 016766 144744 000004      MOV      ARR.INC,4(SP)      ;
12337 043362 005001      CLR      R1      ; ARR.SEL
12338 043364 000167 000666      JMP      25$      ;
12339 043370 005066 000002      1$: CLR      2(SP)      ; ERR.CNT      6263
12340 043374 104402      2$: TRAP      2      ;
12341 043376 005016      CLR      (SP)      ; TWICE      6266
12342 043400 152777 000040 146332 3$: BISB      #40,@ML.REG+40      ;      6267
12343 043406 016705 146556      MOV      ML.DUT,R5
12344 043412 042705 177770      BIC      #177770,R5
12345 043416 142777 000007 146314      BICB      #7,@ML.REG+40
12346 043424 150577 146310      BISB      R5,@ML.REG+40
12347 043430 010277 146434      MOV      R2,@ML.REG+170      ; TST.PAT,*      6269
12348 043434 010277 146440      MOV      R2,@ML.REG+200      ; TST.PAT,*      6270
12349 043440 010277 146414      MOV      R2,@ML.REG+160      ; TST.PAT,*      6271
12350 043444 152777 000010 146346      BISB      #10,@ML.REG+120      ;      6272
12351 043452 012777 177400 146230      MOV      #-400,@ML.REG+10      ;      6273

```

12353													22-Oct-1980 10:47:44	TOPS
12354													22-Oct-1980 10:45:32	PA:<
12355														
12356	043460	012777	010342	146232		MOV	#IO.BUF,@ML.REG+20							6274
12357	043466	010177	146236			MOV	R1,@ML.REG+30							6275
12358	043472	012777	000061	146200		MOV	#61,@ML.REG							6276
12359	043500	005003				CLR	R3							6278
12360	043502	012704	000001		4\$:	MOV	#1,R4							6280
12361	043506	001411			5\$:	BEQ	8\$							
12362	043510	016705	136402			MOV	L\$DLY,R5							
12363	043514	001404				BEQ	7\$							
12364	043516	005066	000014		6\$:	CLR	14(SP)							
12365	043522	005305				DEC	R5							
12366	043524	001374				BNE	6\$							
12367	043526	005304			7\$:	DEC	R4							
12368	043530	000766				BR	5\$							
12369	043532	152777	000020	146260	8\$:	BISB	#20,@ML.REG+120							6281
12370	043540	005203				INC	R3							6278
12371	043542	020327	000177			CMP	R3,#177							
12372	043546	003755				BLE	4\$							
12373	043550	152777	000040	146162		BISB	#40,@ML.REG+40							6282
12374	043556	016705	146406			MOV	ML.DUT,R5							
12375	043562	042705	177770			BIC	#177770,R5							
12376	043566	142777	000007	146144		BICB	#7,@ML.REG+40							
12377	043574	150577	146140			BISB	R5,@ML.REG+40							
12378	043600	152777	000010	146212		BISB	#10,@ML.REG+120							6285
12379	043606	012777	177400	146074		MOV	#-400,@ML.REG+10							6286
12380	043614	012777	010342	146076		MOV	#IO.BUF,@ML.REG+20							6287
12381	043622	010177	146102			MOV	R1,@ML.REG+30							6288
12382	043626	012777	000071	146044		MOV	#71,@ML.REG							6289
12383	043634	012704	000001			MOV	#1,R4							6290
12384	043640	001411			9\$:	BEQ	12\$							
12385	043642	016705	136250			MOV	L\$DLY,R5							
12386	043646	001404				BEQ	11\$							
12387	043650	005066	000014		10\$:	CLR	14(SP)							
12388	043654	005305				DEC	R5							
12389	043656	001374				BNE	10\$							
12390	043660	005304			11\$:	DEC	R4							
12391	043662	000766				BR	9\$							
12392	043664	005003			12\$:	CLR	R3							6292
12393	043666	017767	146236	145766	13\$:	MOV	@ML.REG+230,PD.TEMP							6294
12394	043674	152777	000020	146116		BISB	#20,@ML.REG+120							6295
12395	043702	012704	000001			MOV	#1,R4							6296
12396	043706	001411			14\$:	BEQ	17\$							
12397	043710	016705	136202			MOV	L\$DLY,R5							
12398	043714	001404				BEQ	16\$							
12399	043716	005066	000014		15\$:	CLR	14(SP)							
12400	043722	005305				DEC	R5							
12401	043724	001374				BNE	15\$							
12402	043726	005304			16\$:	DEC	R4							
12403	043730	000766				BR	14\$							
12404	043732	017767	146132	143754	17\$:	MOV	@ML.REG+170,D1.TEMP							
12405	043740	017767	146134	143750		MOV	@ML.REG+200,D2.TEMP							
12406	043746	017767	146106	143744		MOV	@ML.REG+160,E2.TEMP							
12407	043754	005004				CLR	R4							6299

Address	OpCode	Operand1	Operand2	Label	Comment	SeqNo
12409				:ML4		
12410				:		
12411				:		
12412	043756	010405		18\$:	MOV R4,R5 ; NIB.PTR,*	6301
12413	043760	006205			ASR R5	
12414	043762	006205			ASR R5	
12415	043764	006205			ASR R5	
12416	043766	062705	011662		ADD #PD.TEMP,R5	
12417	043772	010546			MOV R5,-(SP)	
12418	043774	010446			MOV R4,-(SP) ; NIB.PTR,*	
12419	043776	042716	177770		BIC #177770,(SP)	
12420	044002	012746	000001		MOV #1,-(SP)	
12421	044006	005046			CLR -(SP)	
12422	044010	004767	137066		JSR PC,BL\$GT2	
12423	044014	062706	000010		ADD #10,SP	
12424	044020	005700			TST R0	
12425	044022	001017			BNE 20\$	
12426	044024	010446			MOV R4,-(SP) ; NIB.PTR,*	6304
12427	044026	010246			MOV R2,-(SP) ; TST.PAT,*	
12428	044030	012746	000020		MOV #20,-(SP)	
12429	044034	060616			ADD SP,(SP) ; ERR.FLG,*	
12430	044036	004767	146616		JSR PC,TST.LNG.WRD	
12431	044042	026627	000020	000001	CMP 20(SP),#1 ; ERR.FLG,*	6306
12432	044050	001002			BNE 19\$	
12433	044052	005266	000010		INC 10(SP) ; ERR.CNT	
12434	044056	062706	000006	19\$:	ADD #6,SP ;	6303
12435	044062	005204		20\$:	INC R4 ; NIB.PTR	6299
12436	044064	020427	000011		CMP R4,#11 ; NIB.PTR,*	
12437	044070	003732			BLE 18\$	
12438	044072	005203			INC R3 ; WD.CNT	6292
12439	044074	020327	000177		CMP R3,#177 ; WD.CNT,*	
12440	044100	003672			BLE 13\$	
12441	044102	005102			COM R2 ; TST.PAT	6313
12442	044104	005216			INC (SP) ; TWICE	6266
12443	044106	021627	000001		CMP (SP),#1 ; TWICE,*	
12444	044112	003002			BGT 21\$	
12445	044114	000167	177260		JMP 3\$	
12446	044120	026627	000002	000005	21\$:	
12447	044126	003434			CMP 2(SP),#5 ; ERR.CNT,*	6316
12448	044130	104455			BLE 22\$	
12449	044132	000116			TRAP 55 ;	6319
12450	044134	007534			.WORD 116	
12451	044136	000000			.WORD ARR.DAT	
12452	044140	012746	005740		.WORD 0	
12453	044144	012746	006232		MOV #WRD.10,-(SP) ;	6320
12454	044150	012746	006220		MOV #WRD.37,-(SP)	
12455	044154	012746	005366		MOV #WRD.35,-(SP)	
12456	044160	012746	000004		MOV #THR.FMT,-(SP)	
12457	044164	010600			MOV #4,-(SP)	
12458	044166	104414			MOV SP,R0 ; SP,*	
12459	044170	010116			TRAP 14	
12460	044172	012746	004602		MOV R1,(SP) ; ARR.SEL,*	6321
12461	044176	012746	000002		MOV #FMT.9,-(SP)	
12462	044202	010600			MOV #2,-(SP)	
12463	044204	104414			MOV SP,R0 ; SP,*	
					TRAP 14	


```

12465 ;ML4
12466 ;
12467 ;
12468 044206 012766 000001 000026 MOV #1,26(SP) ; *,DODU.FLG 6322
12469 044214 062706 000016 ADD #16,SP ; 6318
12470 044220 104467 22$: TRAP 67 ; 6323
12471 044222 006000 ROR R0
12472 044224 103002 BHIS 23$
12473 044226 000167 177142 JMP 2$
12474 044232 026627 000010 000001 23$: CMP 10(SP),#1 ; DODU.FLG,* 6327
12475 044240 001004 BNE 24$
12476 044242 016700 145720 MOV ML.LUN,R0 ; 6330
12477 044246 104451 TRAP 51
12478 044250 104444 TRAP 44
12479 044252 066601 000004 24$: ADD 4(SP),R1 ; *,ARR.SEL 6261
12480 044256 020166 000006 25$: CMP R1,6(SP) ; ARR.SEL,*
12481 044262 003002 BGT 26$
12482 044264 000167 177100 JMP 1$
12483 044270 062706 000016 26$: ADD #16,SP ; 6208
12484 044274 000207 RTS PC
12485
12486 ; Routine Size: 244 words
12487 ; Maximum stack depth per invocation: 20 words
12492
12493
12497
12501 044276 T25::
12502 044276 004767 177024 1$: JSR PC,$T25 ; 6334
12503 044302 104466 TRAP 66
12504 044304 006000 ROR R0
12505 044306 103773 BLO 1$
12506 044310 000207 RTS PC
12507
12508 ; Routine Size: 6 words
12509 ; Maximum stack depth per invocation: 0 words
12514
12515
12516 ; 6337 !<BLF/PAGE>

```

22-Oct-1980 10:47:44 TOPS
22-Oct-1980 10:45:32 PA:<

22-Oct-1980 10:47:44
 22-Oct-1980 10:45:32

TOPS-20 Bliss-16 V2(206)
 PA:<NEALE>BL2ML4.BLI.2 (68)

12518 :ML4
 12519 :
 12520 :
 12521 :
 12522 :
 12523 :
 12524 :
 12525 :
 12526 :
 12527 :
 12528 :
 12529 :
 12530 :
 12531 :
 12532 :
 12533 :
 12534 :
 12535 :
 12536 :
 12537 :
 12538 :
 12539 :
 12540 :
 12541 :
 12542 :
 12543 :
 12544 :
 12545 :
 12546 :
 12547 :
 12548 :
 12549 :
 12550 :
 12551 :
 12552 :
 12553 :
 12554 :
 12555 :
 12556 :
 12557 :
 12558 :
 12559 :
 12560 :
 12561 :
 12562 :
 12563 :
 12564 :
 12565 :
 12566 :
 12567 :
 12568 :
 12569 :
 12570 :
 12571 :
 12572 :

6338 !
 6339 BGNTST;
 6340
 6341 !++
 6342
 6343
 6344
 6345
 6346
 6347
 6348
 6349
 6350
 6351
 6352
 6353
 6354
 6355
 6356
 6357
 6358
 6359
 6360
 6361
 6362
 6363
 6364
 6365
 6366
 6367
 6368
 6369
 6370
 6371
 6372
 6373
 6374
 6375
 6376
 6377
 6378
 6379
 6380
 6381
 6382
 6383
 6384
 6385
 6386
 6387
 6388
 6389

BGNTST;

!++

TEST NUMBER: TST 26

TEST NAME: READ WRITE ARRAYS WITH PROM DATA

TEST DESCRIPTION:

COMBINE THE READING OF ARRAY
 MODULE DATA WITH ARRAY MODULE UV PROM DATA AND
 FIND A GOOD BLOCK WHERE FURTHER
 TESTING WILL BE PERFORMED BY:

1. STARTING AT BLOCK 0 WRITE THE BLOCK WITH SELECTED DATA PATTERNS
 AND READ THE BLOCK AVOIDING ANY BAD NIBBLES POINTED TO BY THE
 PROM DATA.

 SET ERROR FLAG IF ANY BAD NIBBLES ARE ENCOUNTERED IN BLOCK.
2. REPEAT WRITING/READING THIS BLOCK UNTIL ALL PATTERNS ARE TESTED
 OR THE ERROR FLAG IS SET.
3. IF ALL PATTERN HAVE BEEN TESTED AND THE ERROR FLAG IS NOT SET
 THEN SAVE THIS BLOCK ADDRESS AS THE GOOD BLOCK ADRS AND EXIT TEST.
4. ELSE IF THE ERROR FLG HAS SET THEN REPEAT TEST AT THE NEXT ROW.
 REPEAT UNTIL A GOOD BLOCK IS FOUND OR LAST ROW IS REACHED.
5. IF NO GOOD BLOCK IS FOUND BY LAST ROW THEN REPORT ERROR AND
 EXIT TEST.

IMPLICIT INPUTS:

RAS INC
 LOADED DURING THE INITIALIZATION CODE AND CONTAINS THE ROW ADDRESS
 INCREMENT VALUE FOR THIS DRIVE.

PD TEMP:

A BITVECTOR OF 16 BITS WHERE THE READ PROM DATA IS STORED AND
 ACCESSED FROM.

IO BUF:

A VECTOR OF 256 WORDS WHERE DATA FOR MBUS READS AND WRITE FUNCTION
 ARE FOUND.

--

local

WRD_CNT,
 NIB_PTR,

!WORD COUNT
 !NIBBLE POINTER

12574 :ML4

12575 :

12576 :

12577 : 6390

12578 : 6391

12579 : 6392

12580 : 6393

12581 : 6394

12582 : 6395

12583 : 6396

12584 : 6397

12585 : 6398

12586 : 6399

12587 : 6400

12588 : 6401

12589 : 6402

12590 : 6403

12591 : 6404

12592 : 6405

12593 : 6406

12594 : 6407

12595 : 6408

12596 : 6409

12597 : 6410

12598 : 6411

12599 : 6412

12600 : 6413

12601 : 6414

12602 : 6415

12603 : 6416

12604 : 6417

12605 : 6418

12606 : 6419

12607 : 6420

12608 : 6421

12609 : 6422

12610 : 6423

12611 : 6424

12612 : 6425

12613 : 6426

12614 : 6427

12615 : 6428

12616 : 6429

12617 : 6430

12618 : 6431

12619 : 6432

12620 : 6433

12621 : 6434

12622 : 6435

12623 : 6436

12624 : 6437

12625 : 6438

12626 : 6439

12627 : 6440

12628 : 6441

```

PASS CNT,
NIB PAT,
DONE FLG,
ERR FLG,
SECTOR_NO:

BGNSUB;
PASS_CNT = -1;
SECTOR_NO = ZEROES;
DONE_FLG = ZERO;

do
begin
do
begin
PASS_CNT = .PASS_CNT + 1;
CLR_MBUS;
DAT_DM = ONE;
MLDA = .SECTOR_NO;
MLWC = not 255;
MLBA = IO_BUF;

case .PASS_CNT from 0 to 3 of
set
[0] :
NIB_PAT = %o'000000';
[1] :
NIB_PAT = %o'17';
[2] :
NIB_PAT = %o'12';
[3] :
NIB_PAT = %o'15'
tes;

incr LD_CNT from 0 to 9 do
LD_LNG_WRD (.LD_CNT, .NIB_PAT);

WRT_LNG_WRD;
MLC51 = write;

incr WRT_CNT from 0 to 127 do
begin
DELAY (ONE US);
DAT_CLK = ONE;
end;

CLR_MBUS;

```

22-Oct-1980 10:47:44

22-Oct-1980 10:45:32

TOPS-20 Bliss-16 V2(206)

PA:<NEALE>BL2ML4.BLI.2 (68)

```

!PASS COUNT
!NIBBLE PATTERN
!DONE FLAG
!ERROR FLAG
!SECTOR NUMBER

```

!THIS LOOP DETERMINES WHEN TO STOP

!THIS LOOP RUNS THE PATTERNS

!INCREMENT THE PASS COUNT

```

!SET DATA DIAG MODE
!LOAD SECTOR NUMBER IN DSA
!LOAD WORD COUNT
!LOAD UBUS ADRS

```

!SELECT A NIBBLE PATTERN

!ZEROES

!ONES

!ALTERNATING ONE'S, ZEROES

!COMPLIMENT ONE'S, ZEROES

!LOAD NIBBLE PATTERN INTO NIBBLE SAVE

```

!LOAD THE DATA DIAG REGISTERS WITH NIBBLE SAVE
!DO A WRITE FUNCTION

```

!WRITE PATTERN INTO THIS BLOCK

```

12630 :ML4
12631 :
12632 :
12633 : 6442 DAT_DM = ONE; !SET DATA DIAG MODE
12634 : 6443 MLDA = .SECTOR_NO; !LOAD SECTOR NUMBER
12635 : 6444 MLWC = not 255; !LOAD WORD COUNT
12636 : 6445 MLBA = IO_BUF; !LOAD UBUS ADRS
12637 : 6446 MLCS1 = read; !DO A READ FUNCTION
12638 : 6447 DELAY (ONE_US);
12639 : 6448 WRD_CNT = -1; !RESET THE WRD COUNT
12640 : 6449
12641 : 6450 do
12642 : 6451 begin !TEST BLOCK FOR NIBBLE PATTERN
12643 : 6452 WRD_CNT = .WRD_CNT + 1; !INCREMENT WRD_CNT
12644 : 6453 PD_TEMP = .MLPD; !GET PROM DATA
12645 : 6454 DAT_CLK = ONE; !CLOCK DATA WRD INTO DIAG REG
12646 : 6455 DELAY (ONE_US);
12647 : 6456 RD_LNG_WRD; !READ DIAG REGISGTER
12648 : 6457 NIB_PTR = -1; !RESET NIBBLE POINTER
12649 : 6458
12650 : 6459 do
12651 : 6460 begin !SEARCH AND TEST GOOD NIBBLES
12652 : 6461 NIB_PTR = .NIB_PTR + 1; !INCREMENT NIBBLE POINTE
12653 : 6462
12654 : 6463 if .PD_TEMP [.NIB_PTR] IS_NOT_SET then TST_LNG_WRD (.NIB_PTR, .NIB_PAT, ERR_FLG);
12655 : 6464
12656 : 6465 end
12657 : 6466 until (.ERR_FLG) or (.NIB_PTR eql 9);
12658 : 6467
12659 : 6468 end
12660 : 6469 until (.ERR_FLG) or (.WRD_CNT eql 127);
12661 : 6470
12662 : 6471 end
12663 : 6472 until (.PASS_CNT eql 3) or (.ERR_FLG IS_SET); !REPEAT UNTIL ALL PAT TESTED OR ERROR FLG GETS SET
12664 : 6473
12665 : 6474 if (.PASS_CNT eql 3) and (.ERR_FLG IS_NOT_SET) !WAS THIS A GOOD BLOCK?
12666 : 6475 then
12667 : 6476 begin !YES
12668 : 6477 DONE_FLG = ONE; !SET DONE FLAG
12669 : 6478 GOOD_BLK = .SECTOR_NO; !GOOD BLOCK GETS THIS SECTOR NO
12670 : 6479 end
12671 : 6480 else
12672 : 6481 begin !NO
12673 : 6482 SECTOR_NO = .SECTOR_NO + .RAS_INC; !INCREMENT ROW NO
12674 : 6483 PASS_CNT = -1; !RESET PASS COUNT
12675 : 6484 end;
12676 : 6485
12677 : 6486 end
12678 : 6487 until (.DONE_FLG IS_SET) or (.SECTOR_NO eql .LST_ARR + .ARR_INC);
12679 : 6488
12680 : 6489 !REPEAT UNTIL GOOD BLK FOUND OR AT LST ROW
12681 : 6490 ENDSUB;
12682 : 6491
12683 : 6492 if .SECTOR_NO eql .LST_ARR + .ARR_INC !SEE IF WE'RE AT THE LAST BLOCK
12684 : 6493 then

```

22-Oct-1980 10:47:44
22-Oct-1980 10:45:32

TOPS-20 Bliss-16 V2(206)
PA:<NEALE>BL2ML4.BLI.2 (68)

22-Oct-1980 10:47:44 TOPS-20 Bliss-16 V2(206)
22-Oct-1980 10:45:32 PA:<NEALE>BL2ML4.BLI.2 (68)

!A GOOD BLK NOT FOUND BEFORE LAST BLK
!ERROR AND DROP UNIT

```

12686 :ML4
12687 :
12688 :
12689 :      6494      begin
12690 :      6495      ERRDF (79, ARR DAT, 0);
12691 :      6496      PRINTB (THR_FMT, WRD_14, PHR_10, FNC_15);
12692 :      6497      DODU (.ML_LUN);
12693 :      6498      DOCLN;
12694 :      6499      end;
12695 :      6500
12696 :      6501      ENDTST;
12700

```

12704	044312	004167	137542	\$T26:	JSR	R1,\$SAVE5	:		6336
12705	044316	162706	000014		SUB	#14,SP	:		
12706	044322	104402		1\$:	TRAP	2	:		6394
12707	044324	012766	177777	000002	MOV	#-1,2(SP)	:	*,PASS.CNT	6397
12708	044332	005002			CLR	R2	:	SECTOR.NO	6398
12709	044334	005066	000004		CLR	4(SP)	:	DONE.FLG	6399
12710	044340	005266	000002	2\$:	INC	2(SP)	:	PASS.CNT	6406
12711	044344	152777	000040	145366	BISB	#40,@ML.REG+40	:		
12712	044352	016705	145612		MOV	ML.DUT,R5	:		
12713	044356	042705	177770		BIC	#177770,R5	:		
12714	044362	142777	000007	145350	BICB	#7,@ML.REG+40	:		
12715	044370	150577	145344		BISB	R5,@ML.REG+40	:		
12716	044374	152777	000010	145416	BISB	#10,@ML.REG+120	:		6408
12717	044402	010277	145322		MOV	R2,@ML.REG+30	:	SECTOR.NO,*	6409
12718	044406	012777	177400	145274	MOV	#-400,@ML.REG+10	:		6410
12719	044414	012777	010342	145276	MOV	#10.BUF,@ML.REG+20	:		6411
12720	044422	016605	000002		MOV	2(SP),R5	:	PASS.CNT,*	6413
12721	044426	006305			ASL	R5	:		
12722	044430	066507	044434		ADD	3\$(R5),PC	:		
12723	044434	000010		3\$:	.WORD	4\$-3\$:		
12724	044436	000014			.WORD	5\$-3\$:		
12725	044440	000022			.WORD	6\$-3\$:		
12726	044442	000030			.WORD	7\$-3\$:		
12727	044444	005001		4\$:	CLR	R1	:	NIB.PAT	6417
12728	044446	000410			BR	8\$:		6413
12729	044450	012701	000017	5\$:	MOV	#17,R1	:	*,NIB.PAT	6420
12730	044454	000405			BR	8\$:		6413
12731	044456	012701	000012	6\$:	MOV	#12,R1	:	*,NIB.PAT	6423
12732	044462	000402			BR	8\$:		6413
12733	044464	012701	000015	7\$:	MOV	#15,R1	:	*,NIB.PAT	6426
12734	044470	005005		8\$:	CLR	R5	:	LD.CNT	6429
12735	044472	010546		9\$:	MOV	R5,-(SP)	:	LD.CNT,*	6430
12736	044474	010146			MOV	R1,-(SP)	:	NIB.PAT,*	
12737	044476	004767	147134		JSR	PC,LD.LNG.WRD	:		
12738	044502	022626			CMP	(SP)+,(SP)+	:		
12739	044504	005205			INC	R5	:	LD.CNT	6429

12797					:ML4					
12798					:					
12799										
12800	045006	017767	145056	142700	23\$:	MOV	@ML.REG+170,D1.TEMP			
12801	045014	017767	145060	142674		MOV	@ML.REG+200,D2.TEMP			
12802	045022	017767	145032	142670		MOV	@ML.REG+160,E2.TEMP			
12803	045030	012716	177777			MOV	#-1,(SP)	:	* ,NIB.PTR	6457
12804	045034	005216			24\$:	INC	(SP)	:	NIB.PTR	6461
12805	045036	011605				MOV	(SP),R5	:	NIB.PTR,*	6463
12806	045040	006205				ASR	R5			
12807	045042	006205				ASR	R5			
12808	045044	006205				ASR	R5			
12809	045046	062705	011662			ADD	#PD.TEMP,R5			
12810	045052	010546				MOV	R5,-(SP)			
12811	045054	016646	000002			MOV	2(SP),-(SP)	:	NIB.PTR,*	
12812	045060	042716	177770			BIC	#177770,(SP)			
12813	045064	012746	000001			MOV	#1,-(SP)			
12814	045070	005046				CLR	-(SP)			
12815	045072	004767	136004			JSR	PC,BL\$GT2			
12816	045076	062706	000010			ADD	#10,SP			
12817	045102	005700				TST	R0			
12818	045104	001011				BNE	25\$			
12819	045106	011646				MOV	(SP),-(SP)	:	NIB.PTR,*	
12820	045110	010146				MOV	R1,-(SP)	:	NIB.PAT,*	
12821	045112	012746	000016			MOV	#16,-(SP)			
12822	045116	060616				ADD	SP,(SP)	:	ERR.FLG,*	
12823	045120	004767	145534			JSR	PC,TST.LNG.WRD			
12824	045124	062706	000006			ADD	#6,SP			
12825	045130	016605	000010		25\$:	MOV	10(SP),R5	:	ERR.FLG,*	6466
12826	045134	032705	000001			BIT	#1,R5			
12827	045140	001012				BNE	26\$			
12828	045142	021627	000011			CMP	(SP),#11	:	NIB.PTR,*	
12829	045146	001332				BNE	24\$			
12830	045150	032705	000001			BIT	#1,R5	:		6469
12831	045154	001004				BNE	26\$			
12832	045156	026627	000006	000177		CMP	6(SP),#177	:	WRD.CNT,*	
12833	045164	001264				BNE	19\$			
12834	045166	005004			26\$:	CLR	R4	:		6472
12835	045170	026627	000002	000003		CMP	2(SP),#3	:	PASS.CNT,*	
12836	045176	001002				BNE	27\$			
12837	045200	005204				INC	R4			
12838	045202	000403				BR	28\$			
12839	045204	020527	000001		27\$:	CMP	R5,#1			
12840	045210	001030				BNE	31\$			
12841	045212	006004			28\$:	ROR	R4	:		6474
12842	045214	103010				BCC	29\$			
12843	045216	005705				TST	R5			
12844	045220	001006				BNE	29\$			
12845	045222	012766	000001	000004		MOV	#1,4(SP)	:	* ,DONE.FLG	6477
12846	045230	010267	143072			MOV	R2,GOOD.BLK	:	SECTOR.NO,*	6478
12847	045234	000405				BR	30\$:		6474
12848	045236	066702	144424		29\$:	ADD	RAS.INC,R2	:	* ,SECTOR.NO	6482
12849	045242	012766	177777	000002		MOV	#-1,2(SP)	:	* ,PASS.CNT	6483
12850	045250	026627	000004	000001	30\$:	CMP	4(SP),#1	:	DONE.FLG,*	6487
12851	045256	001410				BEQ	32\$			

```

12853      :ML4
12854      :
12855
12856 045260 016705 143054      MOV      LST.ARR,R5
12857 045264 066705 143034      ADD      ARR.INC,R5
12858 045270 020205                CMP      R2,R5                ; SECTOR.NO,*
12859 045272 001402                31$:    BEQ      32$
12860 045274 000167 177040      JMP      2$
12861 045300 104467                32$:    TRAP     67
12862 045302 006000                ROR      R0
12863 045304 103002                BHIS     33$
12864 045306 000167 177010      JMP      1$
12865 045312 016705 143022      33$:    MOV      LST.ARR,R5                ;
12866 045316 066705 143002      ADD      ARR.INC,R5                ; SECTOR.NO,*
12867 045322 020205                CMP      R2,R5
12868 045324 001026                BNE     34$
12869 045326 104455                TRAP     55                ;
12870 045330 000117                .WORD    117
12871 045332 007534                .WORD    ARR.DAT
12872 045334 000000                .WORD    0
12873 045336 012746 007162      MOV      #FNC.15,-(SP)                ;
12874 045342 012746 006740      MOV      #PHR.10,-(SP)
12875 045346 012746 005774      MOV      #WRD.14,-(SP)
12876 045352 012746 005366      MOV      #THR.FMT,-(SP)
12877 045356 012746 000004      MOV      #4,-(SP)
12878 045362 010600                MOV      SP,R0                ; SP,*
12879 045364 104414                TRAP     14
12880 045366 016700 144574      MOV      ML.LUN,R0                ;
12881 045372 104451                TRAP     51
12882 045374 104444                TRAP     44
12883 045376 062706 000012      34$:    ADD      #12,SP                ;
12884 045402 062706 000014      ADD      #14,SP                ;
12885 045406 000207                RTS      PC
12886
12887      ; Routine Size: 287 words
12888      ; Maximum stack depth per invocation: 17 words
12893
12894
12898
12902 045410                T26::
12903 045410 004767 176676      1$:    JSR      PC,$T26                ;
12904 045414 104466                TRAP     66
12905 045416 006000                ROR      R0
12906 045420 103773                BLO     1$
12907 045422 000207                RTS      PC

```


22-Oct-1980 10:47:44
22-Oct-1980 10:45:32

TOPS-20 Bliss-16 V2(206)
PA:<NEALE>BL2ML4.BLI.2 (69)

12913 :ML4
12914 :
12915 :
12916 :
12917 :
12918 :
12919 :
12920 :
12921 :
12922 :
12923 :
12924 :
12925 :
12926 :
12927 :
12928 :
12929 :
12930 :
12931 :
12932 :
12933 :
12934 :
12935 :
12936 :
12937 :
12938 :
12939 :
12940 :
12941 :
12942 :
12943 :
12944 :
12945 :
12946 :
12947 :
12948 :
12949 :
12950 :
12951 :
12952 :
12953 :
12954 :
12955 :
12956 :
12957 :
12958 :
12959 :
12960 :
12961 :
12962 :
12963 :
12964 :
12965 :
12966 :
12967 :

6503
6504
6505
6506
6507
6508
6509
6510
6511
6512
6513
6514
6515
6516
6517
6518
6519
6520
6521
6522
6523
6524
6525
6526
6527
6528
6529
6530
6531
6532
6533
6534
6535
6536
6537
6538
6539
6540
6541
6542
6543
6544
6545
6546
6547
6548
6549
6550
6551
6552
6553
6554

!
!
BGNTST;
!
!++
TEST NUMBER: TST 27
TEST NAME: REFRESH TIMING TEST
TEST DESCRIPTION:
TEST THE MEMORY ARRAY MODULES
REFRESH TIMING AND CONTROL LOGIC
REG BY:
1. WRITING ALL ONES INTO THE
GOOD BLOCK.
2. DELAY FOR 2 MS
3. READ THE GOOD BLOCK FOR ONES.
KEEP COUNT OF BAD NIBBLES
ENCOUNTERED
4. ALLOW A BAD NIBBLE THRESHOLD
OF 20 BAD NIBBLES OUT OF THE
1280 NIBBLES TESTED.
5. IF THRESHOLD IS EXCEEDED THEN
REPORT ERROR AND DROP THE UNIT.
IMPLICIT INPUTS:
PD TEMP:
A BIT VECTOR OF 16 BITS WHERE
THE READ PROM DATA IS STORED
AND ACCESSED FROM.
GLOBAL OWN LOCATION TO THIS
TST.
--
local
TST_PAT,
ERR_CNT,
ERR_FLG;
BGNSUB;
CLR_MBUS;
ERR_CNT = ZERO;

!TEST PATTERN
!ERROR COUNT
!ERROR FLAG

```

12969 :ML4
12970 :
12971 :
12972 : 6555 TST PAT = ONES;
12973 : 6556 MLD1 = .TST_PAT;
12974 : 6557 MLD2 = .TST_PAT;
12975 : 6558 MLE2 = .TST_PAT;
12976 : 6559 DAT_DM_XFER ();
12977 : 6560 MLC51 = write;
12978 : 6561
12979 : 6562 incr WRD_CNT from 0 to 127 do
12980 : 6563 begin
12981 : 6564 DELAY (ONE_US);
12982 : 6565 DAT_CLK = ONE;
12983 : 6566 end;
12984 : 6567
12985 : 6568 CLR_MBUS;
12986 : 6569 DAT_DM_XFER ();
12987 : 6570 MLC51 = read;
12988 : 6571 DELAY (TWO_TH_US);
12989 : 6572
12990 : 6573 incr WRD_CNT from 0 to 127 do
12991 : 6574 begin
12992 : 6575 PD_TEMP = .MLPD;
12993 : 6576 DAT_CLK = ONF;
12994 : 6577 DELAY (ONE_US);
12995 : 6578 RD_LNG_WRD;
12996 : 6579
12997 : 6580 incr NIB_PTR from 0 to 9 do
12998 : 6581 begin
12999 : 6582
13000 : 6583 if .PD_TEMP [.NIB_PTR] IS_NOT_SET then TST_LNG_WRD (.NIB_PTR, .TST_PAT, ERR_FLG);
13001 : 6584
13002 : 6585 !FIND GOOD NIBBLES AND COMPARE THEM
13003 : 6586
13004 : 6587 if .ERR_FLG IS_SET then ERR_CNT = .ERR_CNT + 1; !INCREMENT ERROR COUNT IF ERR_FLG IS SET
13005 : 6588
13006 : 6589 end;
13007 : 6590
13008 : 6591 end;
13009 : 6592
13010 : 6593 ENDSUB;
13011 : 6594
13012 : 6595 if .ERR_CNT gtr 20 !ALLOW 20 NIBBLES TO FAIL
13013 : 6596 then !ERROR IF GTR 20
13014 : 6597 begin
13015 : 6598 ERRDF (80, ASYNC, 0);
13016 : 6599 PRINTB (FIV_FMT, WRD_22, PHR_4, WRD_12, FNC_16, WRD_48);
13017 : 6600 DODU (.ML_LDN);
13018 : 6601 DOCLN;
13019 : 6602 end;
13020 : 6603
13021 : 6604 ENDTST;

```

22-Oct-1980 10:47:44 TOPS-20 Bliss-16 V2(206)
22-Oct-1980 10:45:32 PA:<NEALE>BL2ML4.BLI.2 (69)

!LOAD DATA DIAG REGS WITH TST PAT

!SET UP A DATA DIAG MODE XFERR
!DO A WRITE FUNCTION

!WRITE BLOCK WITH TST PAT

!SET UP A DATA DIAG MODE XFERR
!DO A READ FUNCTION
!DELAY FOR 2 MS

!READ THE BLOCK

!GET THE PROM DATA
!CLOCK DATA WORD INTO DIAG REG

!READ THE DIAG REG

!LOOK AT 10 NIBBLES

13029										
13033	045424	004167	136430		\$T27:	JSR	R1,\$SAVE5	:		6501
13034	045430	024646				CMP	-(SP),-(SP)	:		
13035	045432	104402			1\$:	TRAP	2	:		6550
13036	045434	152777	000040	144276		BISB	#40,@ML.REG+40	:		6552
13037	045442	016703	144522			MOV	ML.DUT,R3	:		
13038	045446	042703	177770			BIC	#177770,R3	:		
13039	045452	142777	000007	144260		BICB	#7,@ML.REG+40	:		
13040	045460	150377	144254			BISB	R3,@ML.REG+40	:		
13041	045464	005005				CLR	R5	:	ERR.CNT	6554
13042	045466	012704	177777			MOV	#-1,R4	:	*,TST.PAT	6555
13043	045472	010477	144372			MOV	R4,@ML.REG+170	:	TST.PAT,*	6556
13044	045476	010477	144376			MOV	R4,@ML.REG+200	:	TST.PAT,*	6557
13045	045502	010477	144352			MOV	R4,@ML.REG+160	:	TST.PAT,*	6558
13046	045506	004767	145114			JSR	PC,DAT.DM.XFER	:		6559
13047	045512	012777	000061	144160		MOV	#61,@ML.REG	:		6560
13048	045520	005001				CLR	R1	:	WRD.CNT	6562
13049	045522	012702	000001		2\$:	MOV	#1,R2	:	*,\$\$TMP2	6564
13050	045526	001411			3\$:	BEQ	6\$:		
13051	045530	016703	134362			MOV	L\$DLY,R3	:	*,\$\$TMP1	
13052	045534	001404				BEQ	5\$:		
13053	045536	005066	000002		4\$:	CLR	2(SP)	:	\$\$TMP	
13054	045542	005303				DEC	R3	:	\$\$TMP1	
13055	045544	001374				BNE	4\$:		
13056	045546	005302			5\$:	DEC	R2	:	\$\$TMP2	
13057	045550	000766				BR	3\$:		
13058	045552	152777	000020	144240	6\$:	BISB	#20,@ML.REG+120	:		6565
13059	045560	005201				INC	R1	:	WRD.CNT	6562
13060	045562	020127	000177			CMP	R1,#177	:	WRD.CNT,*	
13061	045566	003755				BLE	2\$:		
13062	045570	152777	000040	144142		BISB	#40,@ML.REG+40	:		6566
13063	045576	016703	144366			MOV	ML.DUT,R3	:		
13064	045602	042703	177770			BIC	#177770,R3	:		
13065	045606	142777	000007	144124		BICB	#7,@ML.REG+40	:		
13066	045614	150377	144120			BISB	R3,@ML.REG+40	:		
13067	045620	004767	145002			JSR	PC,DAT.DM.XFER	:		6569
13068	045624	012777	000071	144046		MOV	#71,@ML.REG	:		6570
13069	045632	012702	003720			MOV	#3720,R2	:	*,\$\$TMP2	6571
13070	045636	001411			7\$:	BEQ	10\$:		
13071	045640	016703	134252			MOV	L\$DLY,R3	:	*,\$\$TMP1	
13072	045644	001404				BEQ	9\$:		
13073	045646	005066	000002		8\$:	CLR	2(SP)	:	\$\$TMP	
13074	045652	005303				DEC	R3	:	\$\$TMP1	
13075	045654	001374				BNE	8\$:		
13076	045656	005302			9\$:	DEC	R2	:	\$\$TMP2	
13077	045660	000766				BR	7\$:		
13078	045662	005001			10\$:	CLR	R1	:	WRD.CNT	6573

Address	Op-Code	Op-Data	Op-Data	Op-Data	Op-Data	Op-Data	Op-Data	Op-Data	Op-Data	Op-Data	Op-Data	Op-Data	Op-Data	Op-Data
13080						:ML4							22-Oct-1980 10:47:44	TOPS
13081						:							22-Oct-1980 10:45:32	PA:<
13082														
13083	045664	017767	144240	143770	11\$:	MOV	@ML.REG+230,PD.TEMP	:						6575
13084	045672	152777	000020	144120		BISB	#20,@ML.REG+120	:						6576
13085	045700	012702	000001			MOV	#1,R2	:	*	,\$\$TMP2				6577
13086	045704	001411			12\$:	BEQ	15\$:						
13087	045706	016703	134204			MOV	L\$DLY,R3	:	*	,\$\$TMP1				
13088	045712	001404				BEQ	14\$:						
13089	045714	005066	000002		13\$:	CLR	2(SP)	:	\$\$TMP					
13090	045720	005303				DEC	R3	:	\$\$TMP1					
13091	045722	001374				BNE	13\$:						
13092	045724	005302			14\$:	DEC	R2	:	\$\$TMP2					
13093	045726	000766				BR	12\$:						
13094	045730	017767	144134	141756	15\$:	MOV	@ML.REG+170,D1.TEMP	:						
13095	045736	017767	144136	141752		MOV	@ML.REG+200,D2.TEMP	:						
13096	045744	017767	144110	141746		MOV	@ML.REG+160,E2.TEMP	:						
13097	045752	005002				CLR	R2	:	NIB.PTR					6580
13098	045754	010203			16\$:	MOV	R2,R3	:	NIB.PTR,*					6583
13099	045756	006203				ASR	R3							
13100	045760	006203				ASR	R3							
13101	045762	006203				ASR	R3							
13102	045764	062703	011662			ADD	#PD.TEMP,R3							
13103	045770	010346				MOV	R3,-(SP)							
13104	045772	010246				MOV	R2,-(SP)	:	NIB.PTR,*					
13105	045774	042716	177770			BIC	#177770,(SP)							
13106	046000	012746	000001			MOV	#1,-(SP)							
13107	046004	005046				CLR	-(SP)							
13108	046006	004767	135070			JSR	PC,BL\$GT2							
13109	046012	062706	000010			ADD	#10,SP							
13110	046016	005700				TST	R0							
13111	046020	001011				BNE	17\$							
13112	046022	010246				MOV	R2,-(SP)	:	NIB.PTR,*					
13113	046024	010446				MOV	R4,-(SP)	:	TST.PAT,*					
13114	046026	012746	000006			MOV	#6,-(SP)							
13115	046032	060616				ADD	SP,(SP)	:	ERR.FLG,*					
13116	046034	004767	144620			JSR	PC,TST.LNG.WRD							
13117	046040	062706	000006			ADD	#6,SP							
13118	046044	021627	000001		17\$:	CMP	(SP),#1	:	ERR.FLG,*					6587
13119	046050	001001				BNE	18\$							
13120	046052	005205				INC	R5	:	ERR.CNT					
13121	046054	005202			18\$:	INC	R2	:	NIB.PTR					6580
13122	046056	020227	000011			CMP	R2,#11	:	NIB.PTR,*					
13123	046062	003734				BLE	16\$							
13124	046064	005201				INC	R1	:	WRD.CNT					6573
13125	046066	020127	000177			CMP	R1,#177	:	WRD.CNT,*					
13126	046072	003674				BLE	11\$							
13127	046074	104467				TRAP	67	:						6591
13128	046076	006000				ROR	R0							
13129	046100	103002				BHIS	19\$							
13130	046102	000167	177324			JMP	1\$							
13131	046106	020527	000024		19\$:	CMP	R5,#24	:	ERR.CNT,*					6595
13132	046112	003432				BLE	20\$							
13133	046114	104455				TRAP	55	:						6598
13134	046116	000120				.WORD	120							

```
13136 ;ML4
13137 ;
13138
13139 046120 007444 .WORD ASYNC
13140 046122 000000 .WORD 0
13141 046124 012746 006360 MOV #WRD.48,-(SP) ;
13142 046130 012746 007174 MOV #FNC.16,-(SP) ;
13143 046134 012746 005760 MOV #WRD.12,-(SP) ;
13144 046140 012746 006630 MOV #PHR.4,-(SP) ;
13145 046144 012746 006062 MOV #WRD.22,-(SP) ;
13146 046150 012746 005414 MOV #FIV.FMT,-(SP) ;
13147 046154 012746 000006 MOV #6,-(SP) ;
13148 046160 010600 MOV SP,R0 ; SP,*
13149 046162 104414 TRAP 14 ;
13150 046164 016700 143776 MOV ML.LUN,R0 ;
13151 046170 104451 TRAP 51 ;
13152 046172 104444 TRAP 44 ;
13153 046174 062706 000016 ADD #16,SP ;
13154 046200 022626 20$: CMP (SP)+,(SP)+ ;
13155 046202 000207 RTS PC ;
13156
13157 ; Routine Size: 184 words
13158 ; Maximum stack depth per invocation: 15 words
13163
13164
13168
13172 046204 T27::
13173 046204 004767 177214 1$: JSR PC,$T27 ;
13174 046210 104466 TRAP 66 ;
13175 046212 006000 ROR R0 ;
13176 046214 103773 BLO 1$ ;
13177 046216 000207 RTS PC ;
13178
13179 ; Routine Size: 6 words
13180 ; Maximum stack depth per invocation: 0 words
13185
13186
13187 ; 6605 !<BLF/PAGE>
```

22-Oct-1980 10:47:44
22-Oct-1980 10:45:32

TOPS-20 Bliss-16 V2(206)
PA:<NEALE>BL2ML4.BLI.2 (70)

13189 :ML4
13190 :
13191 :
13192 :
13193 :
13194 :
13195 :
13196 :
13197 :
13198 :
13199 :
13200 :
13201 :
13202 :
13203 :
13204 :
13205 :
13206 :
13207 :
13208 :
13209 :
13210 :
13211 :
13212 :
13213 :
13214 :
13215 :
13216 :
13217 :
13218 :
13219 :
13220 :
13221 :
13222 :
13223 :
13224 :
13225 :
13226 :
13227 :
13228 :
13229 :
13230 :
13231 :
13232 :
13233 :
13234 :
13235 :
13236 :
13237 :
13238 :
13239 :
13240 :
13241 :
13242 :
13243 :

```

6606 !
6607 BGNTST;
6608
6609 !++
6610 TEST NUMBER: TST 28
6611
6612 TEST NAME: ADDRESS COUNTER TEST
6613
6614 TEST DESCRIPTION:
6615 TEST THE ABILITY OF THE ADDRESS
6616 COUNTER TO SUCCESSFULLY COUNT
6617 FROM BLOCK ZERO THROUGH THE
6618 DEVICES LAST BLOCK BY:
6619
6620 1. WRITING THE LAST BLOCK WITH ONES PATTERN.
6621
6622 2. STARTING AT BLOCK ZERO WRITE ZEROES INTO ALL BLOCK UP TO THE
6623 LAST BLOCK ADRS MINUS ONE.
6624
6625 READ THE LAST BLOCK FOR ONES AND ERROR IF ZEROES.
6626
6627 3. STARTING AT BLOCK ZERO WRITE ZEROES INTO ALL BLOCK THROUGH THE LAST
6628 BLOCK.
6629
6630 READ THE LAST BLOCK FOR ZEROES AND ERROR IF STILL ONES.
6631
6632 IMPLICIT INPUTS:
6633 PD TEMP:
6634 A BITVECTOR OF 16 BITS WHERE THE READ PROM DATA IS STORED AND ACCESSED FROM.
6635
6636 --
6637 Local
6638 DODU_FLG, !DROP UNIT FLAG
6639 ERR_FLG, !ERROR FLG
6640 END_CNT, !ENDING SECTOR NUMBER
6641 BG_PAT; !BACKGROUND PATTERN
6642
6643 BGNSUB;
6644 CLR MBUS;
6645 DODU_FLG = ZERO;
6646 BG_PAT = ONES; !BACKGROUND PAT OF ONES
6647 MLD1 = .BG_PAT; !LOAD DATA DIAGS WITH BG PAT
6648 MLD2 = .BG_PAT;
6649 MLE2 = .BG_PAT;
6650 DAT_DM = ONE; !SET DATA DIAG MODE
6651 LAST_BLK_XFER (); !SET UP A LAST BLOCK XFERR
6652 MLCST = write; !DO A WRITE FUNCTION
6653
6654 incr WRD_CNT from 0 to 127 do
6655 begin
6656 DELAY (ONE US);
6657 DAT_CLK = ONE;

```

```
13245 :ML4
13246 :
13247 :
13248 : 6658 end;
13249 : 6659
13250 : 6660 CLR_MBUS;
13251 : 6661 DAT_DM = ONE; !SET DATA DIAG MODE
13252 : 6662 LAST_BLK_XFER (); !SET UP A LAST BLOCK XFERR
13253 : 6663 MLCST = read; !DO A READ FUNCTION
13254 : 6664 DELAY (ONE_US);
13255 : 6665
13256 : 6666 incr WD_CNT from 0 to 127 do !READ THE LAST BLOCK FOR BG PATTERN
13257 : 6667 begin
13258 : 6668 PD_TEMP = .MLPD; !GET PROM DATA
13259 : 6669 DAT_CLK = ONE; !CLOCK OUT THE DATA WORD
13260 : 6670 DELAY (ONE_US);
13261 : 6671 RD_LNG_WRD; !READ THE DATA WORD
13262 : 6672
13263 : 6673 incr NIB_PTR from 0 to 9 do !LOOK AT 10 NIBBLES
13264 : 6674 begin
13265 : 6675
13266 : 6676 if .PD_TEMP [.NIB_PTR] IS_NOT_SET then TST_LNG_WRD (.NIB_PTR, .BG_PAT, ERR_FLG);
13267 : 6677
13268 : 6678 !FIND GOOD NIBBLES AND COMPARE THEM
13269 : 6679
13270 : 6680 if .ERR_FLG IS_SET !SEE IF ERROR WAS FOUND
13271 : 6681 then
13272 : 6682 begin !ERROR IF FLG IS SET
13273 : 6683 ERRDF (81, INTER, 0);
13274 : 6684 PRINTB (THR_FMT, PHR_4, WRD_13, FNC_22);
13275 : 6685 PRINTB (TWO_FMT, FNC_13, WRD_56);
13276 : 6686 EXIT_TST;
13277 : 6687 end;
13278 : 6688
13279 : 6689 end;
13280 : 6690
13281 : 6691 end;
13282 : 6692
13283 : 6693 END_CNT = .LST_BLK - 1; !END AT LAST BLOCK -1
13284 : 6694 MLD1 = ZEROES; !LOAD DATA DIAG REG WITH COMP BG PAT
13285 : 6695 MLD2 = ZEROES;
13286 : 6696 MLE2 = ZEROES;
13287 : 6697
13288 : 6698 incr TWICE from 0 to 1 do !REPEAT LOOP TWICE
13289 : 6699 begin
13290 : 6700 CLR_MBUS;
13291 : 6701 DAT_DM = ONE; !SET DATA DIAG MODE
13292 : 6702 FIRST_BLK_XFER (); !SET UP A FIRST BLOCK XFERR
13293 : 6703 MLCST = write; !DO A WRITE FUNCTION
13294 : 6704
13295 : 6705 incr BLK_CNT from 0 to .END_CNT do !CLOCK THE ADDRESS COUNTER UP TO END_CNT
13296 : 6706
13297 : 6707 incr ADRS_CNT from 0 to 127 do
13298 : 6708 begin
13299 : 6709 DAT_CLK = ONE;
```

22-Oct-1980 10:47:44
22-Oct-1980 10:45:32TOPS-20 Bliss-16 V2(206)
PA:<NEALE>BL2ML4.BLI.2 (70)

22-Oct-1980 10:47:44 TOPS-20 Bliss-16 V2(206)
22-Oct-1980 10:45:32 PA:<NEALE>BL2ML4.BLI.2 (70)

```

13301 :ML4
13302 :
13303 :
13304 :      6710      end;
13305 :      6711
13306 :      6712      CLR_MBUS;
13307 :      6713      DAT_DM = ONE;      !SET DATA DIAG MODE
13308 :      6714      LAST_BLK_XFER ();      !SET UP A LAST BLOCK XFERR
13309 :      6715      MLCST = read;      !DO A READ FUNCTION
13310 :      6716      DELAY (ONE_US);
13311 :      6717
13312 :      6718      incr WD_CNT from 0 to 127 do      !READ THE LAST BLOCK FOR BG PATTERN
13313 :      6719      begin
13314 :      6720      PD_TEMP = .MLPD;      !GET THE PROM DATA
13315 :      6721      DAT_CLK = ONE;      !CLOCK OUT DATA WORD
13316 :      6722      DELAY (ONE_US);
13317 :      6723      RD_LNG_WRD;      !READ DATA WORD
13318 :      6724
13319 :      6725      incr NIB_PTR from 0 to 9 do      !LOOK AT 10 NIBBLES
13320 :      6726      begin
13321 :      6727
13322 :      6728      if .PD_TEMP [.NIB_PTR] IS_NOT_SET then TST_LNG_WRD (.NIB_PTR, .BG_PAT, ERR_FLG);
13323 :      6729
13324 :      6730      !FIND GOOD NIBBLES AND COMPARE THEM
13325 :      6731
13326 :      6732      if .ERR_FLG IS_SET      !SEE IF ERROR WAS FOUND
13327 :      6733      then
13328 :      6734      begin      !ERROR IF FLG IS SET
13329 :      6735      ERRDF (82, ASYNC, 0);
13330 :      6736      PRINTB (THR_FMT, WRD_50, WRD_51, WRD_10);
13331 :      6737      DODU_FLG = ONE;
13332 :      6738      end;
13333 :      6739
13334 :      6740      end;
13335 :      6741
13336 :      6742      end;
13337 :      6743
13338 :      6744      END_CNT = .END_CNT + 1;      !NOW END AT THE LAST BLOCK
13339 :      6745      BG_PAT = not .BG_PAT;      !COMPLIMENT THE BG PATTERN AND REPEAT
13340 :      6746      end;
13341 :      6747
13342 :      6748      ENDSUB;
13343 :      6749
13344 :      6750      if .DODU_FLG IS_SET      !DROP THIS UNIT IF DODU_FLG IS_SET
13345 :      6751      then
13346 :      6752      begin
13347 :      6753      DODU (.ML_LUN);
13348 :      6754      DOCLN;
13349 :      6755      end;
13350 :      6756
13351 :      6757      ENDTST;
13355 :

```


22-Oct-1980 10:47:44 TOPS
22-Oct-1980 10:45:32 PA:<

13357				:ML4			
13358				:			
13359							
13363	046220	004167	135634	\$T28:	JSR	R1,\$SAVE5	
13364	046224	162706	000010		SUB	#10,SP	
13365	046230	104402		1\$:	TRAP	2	
13366	046232	152777	000040	143500	BISB	#40,@ML.REG+40	
13367	046240	016705	143724		MOV	ML.DUT,R5	
13368	046244	042705	177770		BIC	#177770,R5	
13369	046250	142777	000007	143462	BICB	#7,@ML.REG+40	
13370	046256	150577	143456		BISB	R5,@ML.REG+40	
13371	046262	005066	000002		CLR	2(SP)	: DODU.FLG
13372	046266	012701	177777		MOV	#-1,R1	: *,BG.PAT
13373	046272	010177	143572		MOV	R1,@ML.REG+170	: BG.PAT,*
13374	046276	010177	143576		MOV	R1,@ML.REG+200	: BG.PAT,*
13375	046302	010177	143552		MOV	R1,@ML.REG+160	: BG.PAT,*
13376	046306	152777	000010	143504	BISB	#10,@ML.REG+120	
13377	046314	004767	144254		JSR	PC,LAST.BLK.XFER	
13378	046320	012777	000061	143352	MOV	#61,@ML.REG	
13379	046326	005002			CLR	R2	: WRD.CNT
13380	046330	012703	000001	2\$:	MOV	#1,R3	: *,\$\$TMP2
13381	046334	001411		3\$:	BEQ	6\$	
13382	046336	016704	133554		MOV	LSDLY,R4	: *,\$\$TMP1
13383	046342	001404			BEQ	5\$	
13384	046344	005066	000006	4\$:	CLR	6(SP)	: \$\$TMP
13385	046350	005304			DEC	R4	: \$\$TMP1
13386	046352	001374			BNE	4\$	
13387	046354	005303		5\$:	DEC	R3	: \$\$TMP2
13388	046356	000766			BR	3\$	
13389	046360	152777	000020	143432	6\$:	BISB	#20,@ML.REG+120
13390	046366	005202			INC	R2	: WRD.CNT
13391	046370	020227	000177		CMP	R2,#177	: WRD.CNT,*
13392	046374	003755			BLE	2\$	
13393	046376	152777	000040	143334	BISB	#40,@ML.REG+40	
13394	046404	016705	143560		MOV	ML.DUT,R5	
13395	046410	042705	177770		BIC	#177770,R5	
13396	046414	142777	000007	143316	BICB	#7,@ML.REG+40	
13397	046422	150577	143312		BISB	R5,@ML.REG+40	
13398	046426	152777	000010	143364	BISB	#10,@ML.REG+120	
13399	046434	004767	144134		JSR	PC,LAST.BLK.XFER	
13400	046440	012777	000071	143232	MOV	#71,@ML.REG	
13401	046446	012703	000001		MOV	#1,R3	: *,\$\$TMP2
13402	046452	001411		7\$:	BEQ	10\$	
13403	046454	016704	133436		MOV	LSDLY,R4	: *,\$\$TMP1
13404	046460	001404			BEQ	9\$	
13405	046462	005066	000006	8\$:	CLR	6(SP)	: \$\$TMP
13406	046466	005304			DEC	R4	: \$\$TMP1
13407	046470	001374			BNE	8\$	
13408	046472	005303		9\$:	DEC	R3	: \$\$TMP2
13409	046474	000766			BR	7\$	
13410	046476	005002		10\$:	CLR	R2	: WD.CNT

6604

6641

6643

6645

6646

6647

6648

6649

6650

6651

6652

6654

6656

6657

6654

6658

6661

6662

6663

6664

6666

Address	OpCode	Operand 1	Operand 2	Operand 3	Operand 4	Instruction	Comments	Timestamp	Page
13412								22-Oct-1980 10:47:44	TOPS
13413								22-Oct-1980 10:45:32	PA:<
13414									
13415	046500	017767	143424	143154	11\$:	MOV @ML.REG+230,PD.TEMP			6668
13416	046506	152777	000020	143304		BISB #20,@ML.REG+120			6669
13417	046514	012703	000001			MOV #1,R3	; *,\$\$TMP2		6670
13418	046520	001411			12\$:	BEQ 15\$			
13419	046522	016704	133370			MOV L\$DLY,R4	; *,\$\$TMP1		
13420	046526	001404				BEQ 14\$			
13421	046530	005066	000006		13\$:	CLR 6(SP)	; \$\$TMP		
13422	046534	005304				DEC R4	; \$\$TMP1		
13423	046536	001374				BNE 13\$			
13424	046540	005303			14\$:	DEC R3	; \$\$TMP2		
13425	046542	000766				BR 12\$			
13426	046544	017767	143320	141142	15\$:	MOV @ML.REG+170,D1.TEMP			
13427	046552	017767	143322	141136		MOV @ML.REG+200,D2.TEMP			
13428	046560	017767	143274	141132		MOV @ML.REG+160,E2.TEMP			
13429	046566	005003				CLR R3	; NIB.PTR		6673
13430	046570	010304			16\$:	MOV R3,R4	; NIB.PTR,*		6676
13431	046572	006204				ASR R4			
13432	046574	006204				ASR R4			
13433	046576	006204				ASR R4			
13434	046600	062704	011662			ADD #PD.TEMP,R4			
13435	046604	010446				MOV R4,-(SP)			
13436	046606	010346				MOV R3,-(SP)	; NIB.PTR,*		
13437	046610	042716	177770			BIC #177770,(SP)			
13438	046614	012746	000001			MOV #1,-(SP)			
13439	046620	005046				CLR -(SP)			
13440	046622	004767	134254			JSR PC,BL\$GT2			
13441	046626	062706	000010			ADD #10,SP			
13442	046632	005700				TST R0			
13443	046634	001011				BNE 17\$			
13444	046636	010346				MOV R3,-(SP)	; NIB.PTR,*		
13445	046640	010146				MOV R1,-(SP)	; BG.PAT,*		
13446	046642	012746	000012			MOV #12,-(SP)			
13447	046646	060616				ADD SP,(SP)	; ERR.FLG,*		
13448	046650	004767	144004			JSR PC,TST.LNG.WRD			
13449	046654	062706	000006			ADD #6,SP			
13450	046660	026627	000004	000001	17\$:	CMP 4(SP),#1	; ERR.FLG,*		6680
13451	046666	001037				BNE 18\$			
13452	046670	104455				TRAP 55			6683
13453	046672	000121				.WORD 121			
13454	046674	007622				.WORD INTER			
13455	046676	000000				.WORD 0			
13456	046700	012746	007254			MOV #FNC.22,-(SP)			6684
13457	046704	012746	005770			MOV #WRD.13,-(SP)			
13458	046710	012746	006630			MOV #PHR.4,-(SP)			
13459	046714	012746	005366			MOV #THR.FMT,-(SP)			
13460	046720	012746	000004			MOV #4,-(SP)			
13461	046724	010600				MOV SP,R0	; SP,*		
13462	046726	104414				TRAP 14			
13463	046730	012716	006454			MOV #WRD.56,(SP)			6685
13464	046734	012746	007136			MOV #FNC.13,-(SP)			
13465	046740	012746	005356			MOV #TWO.FMT,-(SP)			
13466	046744	012746	000003			MOV #3,-(SP)			

Address	Op-Code	Operand 1	Operand 2	Operand 3	Label	Instruction	Comments	Address
13468								
13469					:ML4			
13470					:			
13471	046750	010600				MOV SP,R0	: SP,*	
13472	046752	104414				TRAP 14		
13473	046754	104463				TRAP 63		
13474	046756	062706	000020			ADD #20,SP	:	6680
13475	046762	000167	000604			JMP 37\$:	6682
13476	046766	005203		18\$:		INC R3	: NIB.PTR	6673
13477	046770	020327	000011			CMP R3,#11	: NIB.PTR,*	
13478	046774	003675				BLE 16\$		
13479	046776	005202				INC R2	: WD.CNT	6666
13480	047000	020227	000177			CMP R2,#177	: WD.CNT,*	
13481	047004	003635				BLE 11\$		
13482	047006	016716	141322			MOV LST.BLK,(SP)	: *,END.CNT	6693
13483	047012	005316				DEC (SP)	: END.CNT	
13484	047014	005077	143050			CLR @ML.REG+170	:	6694
13485	047020	005077	143054			CLR @ML.REG+200	:	6695
13486	047024	005077	143030			CLR @ML.REG+160	:	6696
13487	047030	005005				CLR R5	: TWICE	6698
13488	047032	152777	000040	142700	19\$:	BISB #40,@ML.REG+40	:	6699
13489	047040	016704	143124			MOV ML.DUT,R4		
13490	047044	042704	177770			BIC #177770,R4		
13491	047050	142777	000007	142662		BICB #7,@ML.REG+40		
13492	047056	150477	142656			BISB R4,@ML.REG+40		
13493	047062	152777	000010	142730		BISB #10,@ML.REG+120	:	6701
13494	047070	004767	143424			JSR PC,FIRST.BLK.XFER	:	6702
13495	047074	012777	000061	142576		MOV #61,@ML.REG	:	6703
13496	047102	005003				CLR R3	: BLK.CNT	6705
13497	047104	000411				BR 22\$		
13498	047106	005004			20\$:	CLR R4	: ADRS.CNT	6707
13499	047110	152777	000020	142702	21\$:	BISB #20,@ML.REG+120	:	6709
13500	047116	005204				INC R4	: ADRS.CNT	6707
13501	047120	020427	000177			CMP R4,#177	: ADRS.CNT,*	
13502	047124	003771				BLE 21\$		
13503	047126	005203				INC R3	: BLK.CNT	6705
13504	047130	020316			22\$:	CMP R3,(SP)	: BLK.CNT,END.CNT	
13505	047132	003765				BLE 20\$		
13506	047134	152777	000040	142576		BISB #40,@ML.REG+40	:	6710
13507	047142	016704	143022			MOV ML.DUT,R4		
13508	047146	042704	177770			BIC #177770,R4		
13509	047152	142777	000007	142560		BICB #7,@ML.REG+40		
13510	047160	150477	142554			BISB R4,@ML.REG+40		
13511	047164	152777	000010	142626		BISB #10,@ML.REG+120	:	6713
13512	047172	004767	143376			JSR PC,LAST.BLK.XFER	:	6714
13513	047176	012777	000071	142474		MOV #71,@ML.REG	:	6715
13514	047204	012703	000001			MOV #1,R3	: *,\$\$TMP2	6716
13515	047210	001411			23\$:	BEQ 26\$		
13516	047212	016704	132700			MOV L\$DLY,R4	: *,\$\$TMP1	
13517	047216	001404				BEQ 25\$		
13518	047220	005066	000006		24\$:	CLR 6(SP)	: \$\$TMP	
13519	047224	005304				DEC R4	: \$\$TMP1	
13520	047226	001374				BNE 24\$		
13521	047230	005303			25\$:	DEC R3	: \$\$TMP2	
13522	047232	000766				BR 23\$		

22-Oct-1980 10:47:44 TOPS
22-Oct-1980 10:45:32 PA:<

```
13580 ;ML4
13581 ;
13582 ;
13583 047502 020327 000011      CMP    R3,#11      ; NIB.PTR,*
13584 047506 003707            BLE    32$
13585 047510 005202            INC    R2          ; WD.CNT          6718
13586 047512 020227 000177      CMP    R2,#177    ; WD.CNT,*
13587 047516 003647            BLE    27$
13588 047520 005216            INC    (SP)       ; END.CNT        6744
13589 047522 005101            COM    R1         ; BG.PAT         6745
13590 047524 005205            INC    R5         ; TWICE         6698
13591 047526 020527 000001      CMP    R5,#1      ; TWICE,*
13592 047532 003002            BGT   35$
13593 047534 000167 177272      JMP    19$
13594 047540 104467            35$: TRAP 67      ;
13595 047542 006000            ROR   R0
13596 047544 103002            BHIS  36$
13597 047546 000167 176456      JMP    1$
13598 047552 026627 000002 000001 36$: CMP  2(SP),#1    ; DODU.FLG,*    6750
13599 047560 001004            BNE   37$
13600 047562 016700 142400      MOV   ML.LUN,R0  ;
13601 047566 104451            TRAP  51          ;
13602 047570 104444            TRAP  44          ;
13603 047572 062706 000010      37$: ADD #10,SP  ;
13604 047576 000207            RTS   PC          ;
13605
13606 ; Routine Size: 376 words
13607 ; Maximum stack depth per invocation: 18 words
13612
13613
13617
13621 047600      T28::
13622 047600 004767 176414      1$: JSR PC,$T28  ;
13623 047604 104466            TRAP  66
13624 047606 006000            ROR   R0
13625 047610 103773            BLO   1$
13626 047612 000207            RTS   PC
13627
13628 ; Routine Size: 6 words
13629 ; Maximum stack depth per invocation: 0 words
13634 ;ML4
13635 ;
13636 ;
13637 ; 6758 !<BLF/PAGE>
```

22-Oct-1980 10:47:44 TOPS-20 Bliss-16 V2(206)
22-Oct-1980 10:45:32 PA:<NEALE>BL2ML4.BLI.2 (70)

22-Oct-1980 10:47:44
22-Oct-1980 10:45:32

TOPS-20 Bliss-16 V2(206)
PA:<NEALE>BL2ML4.BLI.2 (71)

```

13639 :ML4
13640 :
13641 :
13642 : 6759 :
13643 : 6760 :
13644 : 6761 : BGNTST;
13645 : 6762 :
13646 : 6763 : ++
13647 : 6764 : TEST NUMBER: TST 29
13648 : 6765 :
13649 : 6766 : TEST NAME: ARRAY MODULE SELECTION TEST
13650 : 6767 :
13651 : 6768 : TEST DESCRIPTION:
13652 : 6769 :
13653 : 6770 : TEST FOR UNIQUE ARRAY MODULE
13654 : 6771 : SELECTION BY:
13655 : 6772 :
13656 : 6773 : 1. WRITING THE RESPECTIVE ARRAY
13657 : 6774 : MODULE POSITION NUMBER INTO
13658 : 6775 : THE FIRST GOOD NIBBLE FOUND
13659 : 6776 : IN THE ARRAY. DO FOR ALL
13660 : 6777 : PRESENT ARRAYS.
13661 : 6778 :
13662 : 6779 : 2. READ THE ARRAYS FOR THEIR
13663 : 6780 : RESPECTIVE POSTION NUMBERS.
13664 : 6781 :
13665 : 6782 : IMPLICIT INPUTS:
13666 : 6783 : PD TEMP:
13667 : 6784 : A BIT VECTOR OF 16 BITS WHERE
13668 : 6785 : THE READ PROM DATA IS STORED
13669 : 6786 : AND ACCESSED FROM.
13670 : 6787 :
13671 : 6788 : GLOBAL OWN LOCATION TO THIS
13672 : 6789 : TST.
13673 : 6790 :
13674 : 6791 : IO BUF :
13675 : 6792 : A VECTOR OF 256 WORDS WHERE
13676 : 6793 : DATA FOR MBUS READS AND WRITE
13677 : 6794 : FUNCTION ARE FOUND.
13678 : 6795 :
13679 : 6796 : A GLOBAL OWN LOCATION TO
13680 : 6797 : THIS TEST.
13681 : 6798 :
13682 : 6799 :
13683 : 6800 : --
13684 : 6801 :
13685 : 6802 : local
13686 : 6803 : WRDS_TSTED,
13687 : 6804 : ARR_SEL,
13688 : 6805 : FND_GD_NIB,
13689 : 6806 : ARR_NUM;
13690 : 6807 :
13691 : 6808 : ARR_SEL = ZEROES;
13692 : 6809 : ARR_SEL = .ARR_SEL - .ARR_INC;
13693 : 6810 :

```

```

!WORDS TESTED
!ARRAY SELECT
!FOUND GOOD NIBBLE
!ARRAY NUMBER

```

```
!START ARR_SEL BACK ONE ARRAY
```

```

13695 ;ML4
13696 ;
13697 ;
13698 ; 6811 incr ARR_CNT from 0 to .OP_NUM_ARR do
13699 ; 6812 begin
13700 ; 6813 CLR_MBUS;
13701 ; 6814 FND_GD_NIB = ZERO;
13702 ; 6815 WRDS_TSTED = ZERO;
13703 ; 6816 DAT_DM = ONE;
13704 ; 6817 MLWC = not 255;
13705 ; 6818 MLBA = IO_BUF;
13706 ; 6819 ARR_SEL = .ARR_SEL + .ARR_INC;
13707 ; 6820 MLD1 = .ARR_SEL;
13708 ; 6821 MLCS1 = write;
13709 ; 6822
13710 ; 6823 do
13711 ; 6824 begin
13712 ; 6825 DELAY (ONE_US);
13713 ; 6826 PD_TEMP = .MLPD;
13714 ; 6827 WRDS_TSTED = .WRDS_TSTED + 1;
13715 ; 6828
13716 ; 6829 incr CNT from 0 to 8 do
13717 ; 6830
13718 ; 6831 if .PD_TEMP [.CNT] IS_NOT_SET
13719 ; 6832 then
13720 ; 6833 begin
13721 ; 6834
13722 ; 6835 case .CNT from 0 to 8 of
13723 ; 6836 set
13724 ; 6837
13725 ; 6838 [0] :
13726 ; 6839 (MLD1)<0, 4> = .ARR_CNT; !NIBBLE 0
13727 ; 6840
13728 ; 6841 [1] :
13729 ; 6842 (MLD1)<4, 4> = .ARR_CNT; !NIBBLE 1
13730 ; 6843
13731 ; 6844 [2] :
13732 ; 6845 (MLD1)<8, 4> = .ARR_CNT; !NIBBLE 2
13733 ; 6846
13734 ; 6847 [3] :
13735 ; 6848 (MLD1)<12, 4> = .ARR_CNT; !NIBBLE 3
13736 ; 6849
13737 ; 6850 [4] :
13738 ; 6851 (MLD2)<0, 4> = .ARR_CNT; !NIBBLE 4
13739 ; 6852
13740 ; 6853 [5] :
13741 ; 6854 (MLD2)<4, 4> = .ARR_CNT; !NIBBLE 5
13742 ; 6855
13743 ; 6856 [6] :
13744 ; 6857 (MLD2)<8, 4> = .ARR_CNT; !NIBBLE 6
13745 ; 6858
13746 ; 6859 [7] :
13747 ; 6860 (MLD2)<12, 4> = .ARR_CNT; !NIBBLE 7
13748 ; 6861
13749 ; 6862 [8] :

```

22-Oct-1980 10:47:44 TOPS-20 Bliss-16 V2(206)
22-Oct-1980 10:45:32 PA:<NEALE>BL2ML4.BLI.2 (71)

!TEST ALL PRESENT ARRAYS

!SET DATA DIAG MODE
!LOAD WORD COUNT
!LOAD IO_BUF
!LOOK AT NEXT ARRAY
!LOAD DSA WITH SELECTED ARRAY AT BLOCK ZERO
!DO A WRITE FUNCTION

!THIS LOOP WRITES ARRAY NUMBERS TO THE ARRAYS

!GET THE PROM DATA
!COUNT WORDS TESTED

!LOOK AT 9 NIBBLES

!FIND A GOOD NIBBLE

!SELECT AND LOAD GOOD NIBBLE WITH ARRAY CNT

```

13751 :ML4
13752 :
13753 :
13754 : 6863 (MLE2)<8, 4> = .ARR_CNT; !NIBBLE 8
13755 : 6864 tes;
13756 : 6865
13757 : 6866 FND_GD_NIB = ONE; !SET FOUND GOOD NIBBLE FLG
13758 : 6867 exitloop; !EXIT THE LOOP
13759 : 6868 end;
13760 : 6869
13761 : 6870 DAT_CLK = ONE; !CLOCK GOOD NIBBLE INTO MEMORY & GET NXT PROM WRD
13762 : 6871 end
13763 : 6872 until (.FND_GD_NIB IS_SET ) or (.WRDS_TSTED eql 14); !DO UNTIL FOUND GOOD NIBBLE OR 14 WRDS TSTED
13764 : 6873
13765 : 6874 if .WRDS_TSTED eql 14 !IF 14 WORDS TSTED
13766 : 6875 then
13767 : 6876 begin !THEN ERROR AND EXIT TESTED
13768 : 6877 ERRDF (111, INTER, 0);
13769 : 6878 PRINTB (SEV_FMT, WRD_14, PHR_10, FNC_15, WRD_12, FNC_17, WRD_37, WRD_56);
13770 : 6879 EXIT_TST;
13771 : 6880 end;
13772 : 6881
13773 : 6882 end;
13774 : 6883
13775 : 6884 ARR_SEL = ZEROES;
13776 : 6885 ARR_SEL = .ARR_SEL - .ARR_INC; !START ARR SEL BACK ONE AGAIN
13777 : 6886
13778 : 6887 incr ARR_CNT from 0 to .OP_NUM_ARR do !TEST ALL PRESENT ARRAYS
13779 : 6888 begin
13780 : 6889 BGNSUB;
13781 : 6890 CLR_MBUS;
13782 : 6891 FND_GD_NIB = ZERO;
13783 : 6892 DAT_DM = ONE; !SET DATA DIAG MODE
13784 : 6893 MLWC = not 255; !LOAD WORD COUNT
13785 : 6894 MLBA = IO_BUF; !LOAD UBUS ADDRESS
13786 : 6895 ARR_SEL = .ARR_SEL + .ARR_INC; !LOOK AT NEXT ARRAY
13787 : 6896 MLDA = .ARR_SEL; !LOAD DSA WITH ARRAY SELECT
13788 : 6897 MLCS1 = read; !DO A READ FUNCTION
13789 : 6898
13790 : 6899 do !THIS LOOP READS ARRAYS FOR ARRAY NUMBERS
13791 : 6900 begin
13792 : 6901 DELAY (ONE_US);
13793 : 6902 PD_TEMP = .MLPD; !GET THE PROM DATA
13794 : 6903
13795 : 6904 incr CNT from 0 to 8 do !LOOK AT 9 NIBBLES
13796 : 6905
13797 : 6906 if .PD_TEMP [CNT] IS_NOT_SET !FIND THE GOOD NIBBLES WHERE ARR NUM IS STORED
13798 : 6907 then
13799 : 6908 begin
13800 : 6909 DAT_CLK = ONE; !CLOCK ARRAY WORD OUT
13801 : 6910
13802 : 6911 case .CNT from 0 to 8 of !SELECT AND READ GOOD NIBBLE
13803 : 6912 set
13804 : 6913 [0] :
13805 : 6914

```

22-Oct-1980 10:47:44 TOPS-20 Bliss-16 V2(206)
22-Oct-1980 10:45:32 PA:<NEALE>BL2ML4.BLI.2 (71)

22-Oct-1980 10:47:44
22-Oct-1980 10:45:32

TOPS-20 Bliss-16 V2(206)
PA:<NEALE>BL2ML4.BLI.2 (71)

```

13807 :ML4
13808 :
13809 :
13810 : 6915 ARR_NUM = .(MLD1)<0, 4>; !NIBBLE 0
13811 : 6916
13812 : 6917 [1] :
13813 : 6918 ARR_NUM = .(MLD1)<4, 4>; !NIBBLE 1
13814 : 6919
13815 : 6920 [2] :
13816 : 6921 ARR_NUM = .(MLD1)<8, 4>; !NIBBLE 2
13817 : 6922
13818 : 6923 [3] :
13819 : 6924 ARR_NUM = .(MLD1)<12, 4>; !NIBBLE 3
13820 : 6925
13821 : 6926 [4] :
13822 : 6927 ARR_NUM = .(MLD2)<0, 4>; !NIBBLE 4
13823 : 6928
13824 : 6929 [5] :
13825 : 6930 ARR_NUM = .(MLD2)<4, 4>; !NIBBLE 5
13826 : 6931
13827 : 6932 [6] :
13828 : 6933 ARR_NUM = .(MLD2)<8, 4>; !NIBBLE 6
13829 : 6934
13830 : 6935 [7] :
13831 : 6936 ARR_NUM = .(MLD2)<12, 4>; !NIBBLE 7
13832 : 6937
13833 : 6938 [8] :
13834 : 6939 ARR_NUM = .(MLE2)<8, 4>; !NIBBLE 8
13835 : 6940 tes;
13836 : 6941
13837 : 6942 FND_GD_NIB = ONE; !SET FND GD NIB FLG
13838 : 6943 exitloop; !EXIT LOOP
13839 : 6944 end;
13840 : 6945
13841 : 6946 DAT_CLK = ONE; !CLOCK OUT NEXT PROM LOCATION
13842 : 6947 end
13843 : 6948 until .FND_GD_NIB IS_SET; !REPEAT UNTIL FOUND THE GOOD NIBBLE
13844 : 6949
13845 : 6950 if .ARR_CNT neq .ARR_NUM !SEE IF ARRAY CONTAINS IT'S ARR NUM
13846 : 6951 then
13847 : 6952 begin !ERROR IF NOT THERE
13848 : 6953 ERRDF (83, ASYNC, 0);
13849 : 6954 PRINTB (FOR_FMT, WRD_39, FNC 17, WRD_37, WRD_10);
13850 : 6955 PRINTB (FMT_14, .ARR_CNT, .ARR_NUM);
13851 : 6956 end;
13852 : 6957
13853 : 6958 ENDSUB;
13854 : 6959 end;
13855 : 6960
13856 : 6961 ENDTST;
13860 :

```

Line	Address	Instruction	Comments	Sequence Number
13862		:ML4		
13863		:		
13864				
13868	047614	004167	134240	\$T29: JSR R1,\$SAVE5
13869	047620	162706	000006	SUB #6,SP
13870	047624	005005		CLR R5 ; ARR.SEL
13871	047626	166705	140472	SUB ARR.INC,R5 ; *,ARR.SEL
13872	047632	016766	140464	000002 MOV OP.NUM.ARR,2(SP)
13873	047640	005003		CLR R3 ; ARR.CNT
13874	047642	000167	000702	JMP 27\$
13875	047646	152777	000040	142064 1\$: BISB #40,@ML.REG+40
13876	047654	016702	142310	MOV ML.DUT,R2
13877	047660	042702	177770	BIC #177770,R2
13878	047664	142777	000007	142046 BICB #7,@ML.REG+40
13879	047672	150277	142042	BISB R2,@ML.REG+40
13880	047676	005016		CLR (SP) ; FND.GD.NIB
13881	047700	005004		CLR R4 ; WRDS.TSTED
13882	047702	152777	000010	142110 BISB #10,@ML.REG+120
13883	047710	012777	177400	141772 MOV #-400,@ML.REG+10
13884	047716	012777	010342	141774 MOV #10.BUF,@ML.REG+20
13885	047724	066705	140374	ADD ARR.INC,R5 ; *,ARR.SEL
13886	047730	010577	141774	MOV R5,@ML.REG+30 ; ARR.SEL,*
13887	047734	012777	000061	141736 MOV #61,@ML.REG
13888	047742	012701	000001	2\$: MOV #1,R1 ; *,\$\$TMP2
13889	047746	001411		3\$: BEQ 6\$
13890	047750	016702	132142	MOV L\$DLY,R2 ; *,\$\$TMP1
13891	047754	001404		BEQ 5\$
13892	047756	005066	000004	4\$: CLR 4(SP) ; \$\$TMP
13893	047762	005302		DEC R2 ; \$\$TMP1
13894	047764	001374		BNE 4\$
13895	047766	005301		5\$: DEC R1 ; \$\$TMP2
13896	047770	000766		BR 3\$
13897	047772	017767	142132	141662 6\$: MOV @ML.REG+230,PD.TEMP
13898	050000	005204		INC R4 ; WRDS.TSTED
13899	050002	005002		CLR R2 ; CNT
13900	050004	010201		7\$: MOV R2,R1 ; CNT,*
13901	050006	006201		ASR R1
13902	050010	006201		ASR R1
13903	050012	006201		ASR R1
13904	050014	062701	011662	ADD #PD.TEMP,R1
13905	050020	010146		MOV R1,-(SP)
13906	050022	010246		MOV R2,-(SP) ; CNT,*
13907	050024	042716	177770	BIC #177770,(SP)
13908	050030	012746	000001	MOV #1,-(SP)
13909	050034	005046		CLR -(SP)
13910	050036	004767	133040	JSR PC,BL\$GT2
13911	050042	062706	000010	ADD #10,SP
13912	050046	005700		TST R0
13913	050050	001155		BNE 23\$
13914	050052	010201		MOV R2,R1 ; CNT,*
13915	050054	006301		ASL R1
13916	050056	066107	050062	ADD 8\$(R1),PC

22-Oct-1980 10:47:44 TOPS
22-Oct-1980 10:45:32 PA:<

6757

6808

6809

6811

6812

6814

6815

6816

6817

6818

6819

6820

6821

6825

6826

6827

6829

6831

6835

Line No.	Instruction	Address	Comment	Label	Operation	Address	Code	Count	Time
13918									
13919									
13920									
13921	050062	000022							
13922	050064	000040							
13923	050066	000072							
13924	050070	000112							
13925	050072	000146							
13926	050074	000164							
13927	050076	000216							
13928	050100	000236							
13929	050102	000272							
13930	050104	010301		9\$:	MOV	R3,R1			
13931	050106	042701	177760		BIC	#177760,R1		: ARR.CNT,*	6839
13932	050112	142777	000017 141750		BICB	#17,@AML.REG+170			
13933	050120	000412			BR	11\$			
13934	050122	010301		10\$:	MOV	R3,R1		: ARR.CNT,*	6842
13935	050124	006301			ASL	R1			
13936	050126	006301			ASL	R1			
13937	050130	006301			ASL	R1			
13938	050132	006301			ASL	R1			
13939	050134	042701	177417		BIC	#177417,R1			
13940	050140	142777	000360 141722		BICB	#360,@AML.REG+170			
13941	050146	150177	141716	11\$:	BISB	R1,@AML.REG+170			
13942	050152	000511			BR	22\$: ; ARR.CNT,*	6835
13943	050154	010301		12\$:	MOV	R3,R1		: ; ARR.CNT,*	6845
13944	050156	000301			SWAB	R1			
13945	050160	042701	170377		BIC	#170377,R1			
13946	050164	042777	007400 141676		BIC	#7400,@AML.REG+170			
13947	050172	000413			BR	14\$			
13948	050174	010301		13\$:	MOV	R3,R1		: ; ARR.CNT,*	6848
13949	050176	000301			SWAB	R1			
13950	050200	006301			ASL	R1			
13951	050202	006301			ASL	R1			
13952	050204	006301			ASL	R1			
13953	050206	006301			ASL	R1			
13954	050210	042701	007777		BIC	#7777,R1			
13955	050214	042777	170000 141646		BIC	#170000,@AML.REG+170			
13956	050222	050177	141642	14\$:	BIS	R1,@AML.REG+170			
13957	050226	000463			BR	22\$: ; ARR.CNT,*	6835
13958	050230	010301		15\$:	MOV	R3,R1		: ; ARR.CNT,*	6851
13959	050232	042701	177760		BIC	#177760,R1			
13960	050236	142777	000017 141634		BICB	#17,@AML.REG+200			
13961	050244	000412			BR	17\$			
13962	050246	010301		16\$:	MOV	R3,R1		: ; ARR.CNT,*	6854
13963	050250	006301			ASL	R1			
13964	050252	006301			ASL	R1			
13965	050254	006301			ASL	R1			
13966	050256	006301			ASL	R1			
13967	050260	042701	177417		BIC	#177417,R1			
13968	050264	142777	000360 141606		BICB	#360,@AML.REG+200			
13969	050272	150177	141602	17\$:	BISB	R1,@AML.REG+200			
13970	050276	000437			BR	22\$: ; ARR.CNT,*	6835
13971	050300	010301		18\$:	MOV	R3,R1		: ; ARR.CNT,*	6857
13972	050302	000301			SWAB	R1			

22-Oct-1980 10:47:44 TOPS
22-Oct-1980 10:45:32 PA:<

14030				:ML4			
14031				:			
14032							
14033	050554	003002			BGT	28\$	
14034	050556	000167	177064		JMP	1\$	
14035	050562	005005		28\$:	CLR	R5	: ARR.SEL
14036	050564	166705	137534		SUB	ARR.INC,R5	: *,ARR.SEL
14037	050570	016766	137526	000002	MOV	OP.NUM.ARR,2(SP)	:
14038	050576	005004			CLR	R4	: ARR.CNT
14039	050600	000167	000516		JMP	55\$	
14040	050604	104402		29\$:	TRAP	2	:
14041	050606	152777	000040	141124	BISB	#40,@ML.REG+40	:
14042	050614	016702	141350		MOV	ML.DUT,R2	
14043	050620	042702	177770		BIC	#177770,R2	
14044	050624	142777	000007	141106	BICB	#7,@ML.REG+40	
14045	050632	150277	141102		BISB	R2,@ML.REG+40	
14046	050636	005016			CLR	(SP)	: FND.GD.NIB
14047	050640	152777	000010	141152	BISB	#10,@ML.REG+120	:
14048	050646	012777	177400	141034	MOV	#-400,@ML.REG+10	:
14049	050654	012777	010342	141036	MOV	#10.BUF,@ML.REG+20	:
14050	050662	066705	137436		ADD	ARR.INC,R5	: *,ARR.SEL
14051	050666	010577	141036		MOV	R5,@ML.REG+30	: ARR.SEL,*
14052	050672	012777	000071	141000	MOV	#71,@ML.REG	:
14053	050700	012701	000001		MOV	#1,R1	: *,\$\$TMP2
14054	050704	001411		30\$:	BEQ	34\$	
14055	050706	016702	131204		MOV	LSDLY,R2	: *,\$\$TMP1
14056	050712	001404			BEQ	33\$	
14057	050714	005066	000004	32\$:	CLR	4(SP)	: \$\$TMP
14058	050720	005302			DEC	R2	: \$\$TMP1
14059	050722	001374			BNE	32\$	
14060	050724	005301		33\$:	DEC	R1	: \$\$TMP2
14061	050726	000766			BR	31\$	
14062	050730	017767	141174	140724	34\$:	MOV	@ML.REG+230,PD.TEMP
14063	050736	005002			CLR	R2	: CNT
14064	050740	010201		35\$:	MOV	R2,R1	: CNT,*
14065	050742	006201			ASR	R1	
14066	050744	006201			ASR	R1	
14067	050746	006201			ASR	R1	
14068	050750	062701	011662		ADD	#PD.TEMP,R1	
14069	050754	010146			MOV	R1,-(SP)	
14070	050756	010246			MOV	R2,-(SP)	: CNT,*
14071	050760	042716	177770		BIC	#177770,(SP)	
14072	050764	012746	000001		MOV	#1,-(SP)	
14073	050770	005046			CLR	-(SP)	
14074	050772	004767	132104		JSR	PC,BL\$GT2	
14075	050776	062706	000010		ADD	#10,SP	
14076	051002	005700			TST	R0	
14077	051004	001070			BNE	50\$	
14078	051006	152777	000020	141004	BISB	#20,@ML.REG+120	:
14079	051014	010201			MOV	R2,R1	: CNT,*
14080	051016	006301			ASL	R1	
14081	051020	066107	051024		ADD	36\$(R1),PC	
14082	051024	000022		36\$:	.WORD	37\$-36\$	
14083	051026	000030			.WORD	38\$-36\$	
14084	051030	000036			.WORD	39\$-36\$	

Address	OpCode	Operand 1	Operand 2	Label	Comment	Seq	
14086							
14087							
14088							
14089	051032	000044			.WORD 40\$-36\$		
14090	051034	000052			.WORD 41\$-36\$		
14091	051036	000060			.WORD 42\$-36\$		
14092	051040	000076			.WORD 44\$-36\$		
14093	051042	000104			.WORD 45\$-36\$		
14094	051044	000122			.WORD 47\$-36\$		
14095	051046	117703	141016	37\$:	MOVB @ML.REG+170,R3 ; *,ARR.NUM	6915	
14096	051052	000440			BR 49\$		
14097	051054	117703	141010	38\$:	MOVB @ML.REG+170,R3 ; *,ARR.NUM	6918	
14098	051060	000413			BR 43\$		
14099	051062	017703	141002	39\$:	MOV @ML.REG+170,R3 ; *,ARR.NUM	6921	
14100	051066	000431			BR 48\$		
14101	051070	017703	140774	40\$:	MOV @ML.REG+170,R3 ; *,ARR.NUM	6924	
14102	051074	000417			BR 46\$		
14103	051076	117703	140776	41\$:	MOVB @ML.REG+200,R3 ; *,ARR.NUM	6927	
14104	051102	000424			BR 49\$		
14105	051104	117703	140770	42\$:	MOVB @ML.REG+200,R3 ; *,ARR.NUM	6930	
14106	051110	006203		43\$:	ASR R3 ; ARR.NUM		
14107	051112	006203			ASR R3 ; ARR.NUM		
14108	051114	006203			ASR R3 ; ARR.NUM		
14109	051116	006203			ASR R3 ; ARR.NUM		
14110	051120	000415			BR 49\$		
14111	051122	017703	140752	44\$:	MOV @ML.REG+200,R3 ; *,ARR.NUM	6933	
14112	051126	000411			BR 48\$		
14113	051130	017703	140744	45\$:	MOV @ML.REG+200,R3 ; *,ARR.NUM	6936	
14114	051134	006203		46\$:	ASR R3 ; ARR.NUM		
14115	051136	006203			ASR R3 ; ARR.NUM		
14116	051140	006203			ASR R3 ; ARR.NUM		
14117	051142	006203			ASR R3 ; ARR.NUM		
14118	051144	000402			BR 48\$		
14119	051146	017703	140706	47\$:	MOV @ML.REG+160,R3 ; *,ARR.NUM	6939	
14120	051152	000303		48\$:	SWAB R3 ; ARR.NUM		
14121	051154	042703	177760	49\$:	BIC #177760,R3 ; *,ARR.NUM		
14122	051160	012716	000001		MOV #1,(SP) ; *,FND.GD.NIB	6942	
14123	051164	000404			BR 51\$;	6943	
14124	051166	005202		50\$:	INC R2 ; CNT	6904	
14125	051170	020227	000010		CMP R2,#10 ; CNT,*		
14126	051174	003661			BLE 35\$		
14127	051176	152777	000020	140614	51\$:	BISB #20,@ML.REG+120 ;	6946
14128	051204	021627	000001		CMP (SP),#1 ; FND.GD.NIB,*	6948	
14129	051210	001233			BNE 30\$		
14130	051212	020403			CMP R4,R3 ; ARR.CNT,ARR.NUM	6950	
14131	051214	001434			BEQ 52\$		
14132	051216	104455			TRAP 55 ;	6953	
14133	051220	000123			.WORD 123		
14134	051222	007444			.WORD ASYNC		
14135	051224	000000			.WORD 0		
14136	051226	012746	005740		MOV #WRD.10,-(SP) ;	6954	
14137	051232	012746	006232		MOV #WRD.37,-(SP)		
14138	051236	012746	007206		MOV #FNC.17,-(SP)		
14139	051242	012746	006250		MOV #WRD.39,-(SP)		
14140	051246	012746	005400		MOV #FOR.FMT,-(SP)		

```
14142      ;ML4
14143      ;
14144
14145 051252 012746 000005      MOV      #5,-(SP)
14146 051256 010600      MOV      SP,R0      ; SP,*
14147 051260 104414      TRAP     14
14148 051262 010316      MOV      R3,(SP)      ; ARR.NUM,*
14149 051264 010446      MOV      R4,-(SP)      ; ARR.CNT,*
14150 051266 012746 005046      MOV      #FMT.14,-(SP)
14151 051272 012746 000003      MOV      #3,-(SP)
14152 051276 010600      MOV      SP,R0      ; SP,*
14153 051300 104414      TRAP     14
14154 051302 062706 000022      ADD      #22,SP
14155 051306 104467 52$:      TRAP     67
14156 051310 006000      ROR      R0
14157 051312 103002      BHIS     54$
14158 051314 000167 177264 53$:      JMP      29$
14159 051320 005204 54$:      INC      R4      ; ARR.CNT
14160 051322 020466 000002 55$:      CMP      R4,2(SP)      ; ARR.CNT,*
14161 051326 003772      BLE     53$
14162 051330 062706 000006 56$:      ADD      #6,SP
14163 051334 000207      RTS      PC
14164
14165      ; Routine Size: 425 words
14166      ; Maximum stack depth per invocation: 18 words
14171
14172
14176
14180 051336      T29::
14181 051336 004767 176252 1$:      JSR      PC,$T29
14182 051342 104466      TRAP     66
14183 051344 006000      ROR      R0
14184 051346 103773      BLO     1$
14185 051350 000207      RTS      PC
14186
14187      ; Routine Size: 6 words
14188      ; Maximum stack depth per invocation: 0 words
14193
14194
14195 ;      6962 !<BLF/PAGE>
```

22-Oct-1980 10:47:44
22-Oct-1980 10:45:32

TOPS-20 Bliss-16 V2(206)
PA:<NEALE>BL2ML4.BLI.2 (72)

14197 :ML4
14198 :
14199 :
14200 :
14201 :
14202 :
14203 :
14204 :
14205 :
14206 :
14207 :
14208 :
14209 :
14210 :
14211 :
14212 :
14213 :
14214 :
14215 :
14216 :
14217 :
14218 :
14219 :
14220 :
14221 :
14222 :
14223 :
14224 :
14225 :
14226 :
14227 :
14228 :
14229 :
14230 :
14231 :
14232 :
14233 :
14234 :
14235 :
14236 :
14237 :
14238 :
14239 :
14240 :
14241 :
14242 :
14243 :
14244 :
14245 :
14246 :
14247 :
14248 :
14249 :
14250 :
14251 :

6963 !
6964 !
6965 !
6966 !
6967 !
6968 !
6969 !
6970 !
6971 !
6972 !
6973 !
6974 !
6975 !
6976 !
6977 !
6978 !
6979 !
6980 !
6981 !
6982 !
6983 !
6984 !
6985 !
6986 !
6987 !
6988 !
6989 !
6990 !
6991 !
6992 !
6993 !
6994 !
6995 !
6996 !
6997 !
6998 !
6999 !
7000 !
7001 !
7002 !
7003 !
7004 !
7005 !
7006 !
7007 !
7008 !
7009 !
7010 !
7011 !
7012 !
7013 !
7014 !

BGNTST;

!++

TEST NUMBER: TST 30

TEST NAME: SEQUENCER EXISTENCE TEST

TEST DESCRIPTION:

TEST SEQUENCER TIMING AND CONTROL
LOGIC FOR EXISTENCE BY:

1. FIRST WRITING THE GOOD BLOCK VIA DAT_DM MODE WITH A BACKGROUND PATTERN OF ONES.
2. VIA A MBUS WRITE FUNCTION WRITE ONES INTO THE GOOD BLOCK.
3. THEN VIA DAT_DM READ GOOD NIBBLES IN THE GOOD BLOCK AND XOR THEM AGAINST THE BACKGROUND PATTERN.

RECORD THE NUMBER OF NIBBLES THAT WERE UNCHANGED OR PARTIALLY CHANGED BY THE MBUS WIRTE.
4. REPORT A FATAL ERROR AND DROP THIS UNIT IF THE NIBBLES TESTED EQUALS THE COUNT OF UNCHANGED NIBBLES.

REPORT AN INTERMEDIATE DIAGNOSIIC MESSAGE IF AT LEAST SOME NIBBLES WERE CHANGED BY THE MBUS WRITE.

IMPLICIT INPUTS:

PD_TEMP:

A BIT VECTOR OF 16 BITS WHERE THE READ PROM DATA IS STORED AND ACCESSED FROM.

IO_BUF:

A VECTOR OF 256 WORDS WHERE DATA FOR MBUS READS AND WRITE FUNCTIONS ARE FOUND.

--

local

BG_PAT,
SUM_BAD,
NIB_TSTED,
ALL_BAD,
RESULT;

!BACK GROUND PATTERN
!SUM NIBBLE ARE BAD
!NIBBLES TESTED
!ALL NIBBLES ARE BAD
!RESULTS OF XOR

CLR MBUS;

BG_PAT = ONES;
MLD1 = .BG_PAT;
MLD2 = .BG_PAT;
MLE2 = .BG_PAT;

!BACKGROUND EQL ONES
!LOAD DATA DIAG REG WITH BG PAT

14253 :ML4

14254 :

14255 :

14256 :

14257 :

14258 :

14259 :

14260 :

14261 :

14262 :

14263 :

14264 :

14265 :

14266 :

14267 :

14268 :

14269 :

14270 :

14271 :

14272 :

14273 :

14274 :

14275 :

14276 :

14277 :

14278 :

14279 :

14280 :

14281 :

14282 :

14283 :

14284 :

14285 :

14286 :

14287 :

14288 :

14289 :

14290 :

14291 :

14292 :

14293 :

14294 :

14295 :

14296 :

14297 :

14298 :

14299 :

14300 :

14301 :

14302 :

14303 :

14304 :

14305 :

14306 :

14307 :

```

7015 DAT_DM_XFER ();
7016 MLC51 = write;
7017
7018 incr WD_CNT from 0 to 127 do
7019     begin
7020     DELAY (ONE_US);
7021     DAT_CLK = ONE;
7022     end;
7023
7024 CLR MBUS;
7025 IO_BUF = ZEROES;
7026 BAI = ONE;
7027 GD_BLK_XFER ();
7028 MLC51 = write;
7029
7030 do
7031     0
7032 until .DRY IS_SET;
7033
7034 BGNSUB;
7035 CLR_MBUS;
7036 ALL_BAD = ZEROES;
7037 SUM_BAD = ZEROES;
7038 NIB_TSTED = ZEROES;
7039 DAT_DM_XFER ();
7040 MLC51 = read;
7041 DELAY (ONE_US);
7042
7043 incr WD_CNT from 0 to 112 do
7044     begin
7045     PD_TEMP = .MLPD;
7046     DAT_CLK = ONE;
7047     DELAY (ONE_US);
7048     RD_LNG_WRD;
7049
7050     incr NIB_PTR from 0 to 8 do
7051
7052         if .PD_TEMP [.NIB_PTR] eql ZERO
7053         then
7054             begin
7055             NIB_TSTED = .NIB_TSTED + 1;
7056             XOR_LNG_WRD (.NIB_PTR, .BG_PAT, RESULT);
7057
7058             if .RESULT<0, 4> eql ZERO
7059             then
7060                 ALL_BAD = .ALL_BAD + 1
7061             else
7062
7063                 if .RESULT<0, 4> neq %'17' then SUM_BAD = .SUM_BAD + 1;
7064
7065             end;
7066

```

22-Oct-1980 10:47:44

TOPS-20 Bliss-16 V2(206)

22-Oct-1980 10:45:32

PA:<NEALE>BL2ML4.BLI.2 (72)

!SET UP A DATA DIAG XFERR AT THE GOOD BLOCK
!DO A WRITE FUNCTION

!WRITE BLOCK WITH BG PAT

!LOAD FIRST WORD OF IO_BUF
!MAKE XFERR SET ON ONE_ADRS
!SET UP A GOOD BLOCK XFERR
!DO A WRITE FUNCTION

!DELAY UNTIL XFER TO COMPLETE

!SET UP SAME DATA DIAG XFERR
!DO A READ FUNCTION

!READ 113 LONG WORDS

!GET PROM DATA
!CLOCK OUT THE DATA WORD

!READ THE DATA WORD

!LOOK AT 9 NIBBLES

!FIND GOOD NIBBLE

!INCREMENT COUNT OF NIBBLES TESTED
!XOR NIBBLE WITH BG PAT

!SEE IF ALL BITS IN NIBBLE WERE BAD?

!INCREMENT COUNT IF ALL BAD

!SEE IF SOME BITS IN NIBBLE WERE BAD

!INCREMENT COUNT IF SOME BAD

22-Oct-1980 10:47:44 TOPS-20 Bliss-16 V2(206)
 22-Oct-1980 10:45:32 PA:<NEALE>BL2ML4.BLI.2 (72)

```

14309 :ML4
14310 :
14311 :
14312 : 7067
14313 : 7068     end;
14314 : 7069
14315 : 7070     ENDSUB;
14316 : 7071
14317 : 7072     if .NIB_TSTED eql .ALL_BAD           !WHERE ALL NIBBLES XFERR'ED ALL BAD?
14318 : 7073     then                               !ERROR IF ALL BAD
14319 : 7074         begin
14320 : 7075         ERRDF (84, SYNC, 0);
14321 : 7076         PRINTB (SEV_FMT, WRD_22, PHR_4, WRD_9, WRD_12, WRD_23, FNC_5, WRD_19);
14322 : 7077         DODU (.ML_LUN);
14323 : 7078         DOCLN;
14324 : 7079         end
14325 : 7080     else
14326 : 7081
14327 : 7082         if .SUM_BAD gtr ZERO           !SEE IF SEE NIBBLE WERE BAD?
14328 : 7083         then
14329 : 7084         begin                               !SOME BAD IS OK SO GIVE INTERMEDIATE ERROR
14330 : 7085         ERRDF (85, INTER, 0);
14331 : 7086         PRINTB (SIX_FMT, PHR_4, WRD_9, WRD_12, WRD_23, FNC_5, WRD_19);
14332 : 7087         end;
14333 : 7088
14334 : 7089     ENDTST;
14338 :
  
```

```

14342 051352 004167 132502      $T30:  JSR      R1,$SAVE5           ; 6961
14343 051356 162706 000010      SUB      #10,SP           ;
14344 051362 152777 000040 140350  BISB    #40,@ML.REG+40    ; 7008
14345 051370 016705 140574      MOV     ML.DUT,R5
14346 051374 042705 177770      BIC    #177770,R5
14347 051400 142777 000007 140332  BICB   #7,@ML.REG+40
14348 051406 150577 140326      BISB   R5,@ML.REG+40
14349 051412 012766 177777 000002  MOV    #-1,2(SP)         ; *.BG.PAT 7011
14350 051420 012777 177777 140442  MOV    #-1,@ML.REG+170   ; 7012
14351 051426 016677 000002 140444  MOV    2(SP),@ML.REG+200 ; BG.PAT,* 7013
14352 051434 016677 000002 140416  MOV    2(SP),@ML.REG+160 ; BG.PAT,* 7014
14353 051442 004767 141160      JSR    PC,DAT.DM.XFER    ; 7015
14354 051446 012777 000061 140224  MOV    #61,@ML.REG      ; 7016
14355 051454 005000      CLR    R0                ; WD.CNT 7018
14356 051456 012701 000001      1$:   MOV    #1,R1        ; *,$$TMP2 7020
14357 051462 001411      2$:   BEQ    5$
14358 051464 016702 130426      MOV    L$DLY,R2         ; *,$$TMP1
14359 051470 001404      BEQ    4$
14360 051472 005066 000006      3$:   CLR    6(SP)         ; $$TMP
14361 051476 005302      DEC    R2                ; $$TMP1
14362 051500 001374      BNE    3$
  
```

22-Oct-1980 10:47:44 TOPS
22-Oct-1980 10:45:32 PA:<

14364						;ML4			
14365						:			
14366						:			
14367	051502	005301				4\$:	DEC	R1	: \$\$TMP2
14368	051504	000766					BR	2\$	
14369	051506	152777	000020	140304		5\$:	BISB	#20,@ML.REG+120	: 7021
14370	051514	005200					INC	R0	: : WD.CNT 7018
14371	051516	020027	000177				CMP	R0,#177	: : WD.CNT,*
14372	051522	003755					BLE	1\$	
14373	051524	152777	000040	140206			BISB	#40,@ML.REG+40	: 7022
14374	051532	016705	140432				MOV	ML.DUT,R5	
14375	051536	042705	177770				BIC	#177770,R5	
14376	051542	142777	000007	140170			BICB	#7,@ML.REG+40	
14377	051550	150577	140164				BISB	R5,@ML.REG+40	
14378	051554	005067	136562				CLR	IO.BUF	: 7025
14379	051560	152777	000010	140152			BISB	#10,@ML.REG+40	: 7026
14380	051566	004767	140750				JSR	PC,GD.BLK.XFER	: 7027
14381	051572	012777	000061	140100			MOV	#61,@ML.REG	: 7028
14382	051600	105777	140144			6\$:	TSTB	@ML.REG+50	: 7032
14383	051604	100375					BPL	6\$	
14384	051606	104402				7\$:	TRAP	2	
14385	051610	152777	000040	140122			BISB	#40,@ML.REG+40	: 7034
14386	051616	016703	140346				MOV	ML.DUT,R3	
14387	051622	042703	177770				BIC	#177770,R3	
14388	051626	142777	000007	140104			BICB	#7,@ML.REG+40	
14389	051634	150377	140100				BISB	R3,@ML.REG+40	
14390	051640	005004					CLR	R4	: ALL.BAD 7036
14391	051642	005016					CLR	(SP)	: SUM.BAD 7037
14392	051644	005005					CLR	R5	: NIB.TSTED 7038
14393	051646	004767	140754				JSR	__DAT.DM.XFER	: 7039
14394	051652	012777	000071	140020			MOV	#71,@ML.REG	: 7040
14395	051660	012701	000001				MOV	#1,R1	: *,\$\$TMP2 7041
14396	051664	001411				8\$:	BEQ	11\$	
14397	051666	016702	130224				MOV	LSDLY,R2	: *,\$\$TMP1
14398	051672	001404					BEQ	10\$	
14399	051674	005066	000006			9\$:	CLR	6(SP)	: \$\$TMP
14400	051700	005302					DEC	R2	: \$\$TMP1
14401	051702	001374					BNE	9\$	
14402	051704	005301				10\$:	DEC	R1	: \$\$TMP2
14403	051706	000766					BR	8\$	
14404	051710	005003				11\$:	CLR	R3	: WD.CNT 7043
14405	051712	017767	140212	137742		12\$:	MOV	@ML.REG+230,PD.TEMP	: 7045
14406	051720	152777	000020	140072			BISB	#20,@ML.REG+120	: 7046
14407	051726	012701	000001				MOV	#1,R1	: *,\$\$TMP2 7047
14408	051732	001411				13\$:	BEQ	16\$	
14409	051734	016702	130156				MOV	LSDLY,R2	: *,\$\$TMP1
14410	051740	001404					BEQ	15\$	
14411	051742	005066	000006			14\$:	CLR	6(SP)	: \$\$TMP
14412	051746	005302					DEC	R2	: \$\$TMP1
14413	051750	001374					BNE	14\$	
14414	051752	005301				15\$:	DEC	R1	: \$\$TMP2
14415	051754	000766					BR	13\$	
14416	051756	017767	140106	135730		16\$:	MOV	@ML.REG+170,D1.TEMP	
14417	051764	017767	140110	135724			MOV	@ML.REG+200,D2.TEMP	
14418	051772	017767	140062	135720			MOV	@ML.REG+160,E2.TEMP	

22-Oct-1980 10:47:44 TOPS
22-Oct-1980 10:45:32 PA:<

14420			:ML4				
14421			:				
14422							
14423	052000	005001		CLR	R1	: NIB.PTR	7050
14424	052002	010102	17\$:	MOV	R1,R2	: NIB.PTR,*	7052
14425	052004	006202		ASR	R2		
14426	052006	006202		ASR	R2		
14427	052010	006202		ASR	R2		
14428	052012	062702	011662	ADD	#PD.TEMP,R2		
14429	052016	010246		MOV	R2,-(SP)		
14430	052020	010146		MOV	R1,-(SP)	: NIB.PTR,*	
14431	052022	042716	177770	BIC	#177770,(SP)		
14432	052026	012746	000001	MOV	#1,-(SP)		
14433	052032	005046		CLR	-(SP)		
14434	052034	004767	131042	JSR	PC,BL\$GT2		
14435	052040	062706	000010	ADD	#10,SP		
14436	052044	005700		TST	R0		
14437	052046	001027		BNE	20\$		
14438	052050	005205		INC	R5	: NIB.TSTED	7055
14439	052052	010146		MOV	R1,-(SP)	: NIB.PTR,*	7056
14440	052054	016646	000004	MOV	4(SP),-(SP)	: BG.PAT,*	
14441	052060	012746	000012	MOV	#12,-(SP)		
14442	052064	060616		ADD	SP,(SP)	: RESULT,*	
14443	052066	004767	141262	JSR	PC,XOR.LNG.WRD		
14444	052072	016602	000012	MOV	12(SP),R2	: RESULT,*	7058
14445	052076	042702	177760	BIC	#177760,R2		
14446	052102	001002		BNE	18\$		
14447	052104	005204		INC	R4	: ALL.BAD	7060
14448	052106	000405		BR	19\$:	7058
14449	052110	020227	000017	18\$:	CMP	R2,#17	7063
14450	052114	001402		BEQ	19\$:	
14451	052116	005266	000006	INC	6(SP)	: SUM.BAD	
14452	052122	062706	000006	19\$:	ADD	#6,SP	7054
14453	052126	005201		20\$:	INC	R1	: NIB.PTR
14454	052130	020127	000010	CMP	R1,#10	: NIB.PTR,*	7050
14455	052134	003722		BLE	17\$		
14456	052136	005203		INC	R3	: WD.CNT	7043
14457	052140	020327	000160	CMP	R3,#160	: WD.CNT,*	
14458	052144	003662		BLE	12\$		
14459	052146	104467		TRAP	67	:	7068
14460	052150	006000		ROR	R0		
14461	052152	103615		BLO	7\$		
14462	052154	020504		CMP	R5,R4	: NIB.TSTED,ALL.BAD	7072
14463	052156	001037		BNE	21\$		
14464	052160	104455		TRAP	55	:	7075
14465	052162	000124		.WORD	124		
14466	052164	007500		.WORD	SYNC		
14467	052166	000000		.WORD	0		
14468	052170	012746	006040	MOV	#WRD.19,-(SP)	:	7076
14469	052174	012746	007020	MOV	#FNC.5,-(SP)		
14470	052200	012746	006076	MOV	#WRD.23,-(SP)		
14471	052204	012746	005760	MOV	#WRD.12,-(SP)		
14472	052210	012746	005726	MOV	#WRD.9,-(SP)		
14473	052214	012746	006630	MOV	#PHR.4,-(SP)		
14474	052220	012746	006062	MOV	#WRD.22,-(SP)		

22-Oct-1980 10:47:44 TOPS
22-Oct-1980 10:45:32 PA:<

```

14476      ;ML4
14477      ;
14478
14479 052224 012746 005452      MOV      #SEV.FMT,-(SP)
14480 052230 012746 000010      MOV      #10,-(SP)
14481 052234 010600              MOV      SP,RO      ; SP,*
14482 052236 104414              TRAP     14
14483 052240 016700 137722      MOV      ML.LUN,RO  ;
14484 052244 104451              TRAP     51
14485 052246 104444              TRAP     44
14486 052250 062706 000022      ADD      #22,SP      ;
14487 052254 000432              BR       22$         ;
14488 052256 005716              21$: TST      (SP)      ; SUM.BAD
14489 052260 003430              BLE     22$         ;
14490 052262 104455              TRAP     55         ;
14491 052264 000125              .WORD   125
14492 052266 007622              .WORD   INTER
14493 052270 000000              .WORD   0
14494 052272 012746 006040      MOV      #WRD.19,-(SP) ;
14495 052276 012746 007020      MOV      #FNC.5,-(SP) ;
14496 052302 012746 006076      MOV      #WRD.23,-(SP) ;
14497 052306 012746 005760      MOV      #WRD.12,-(SP) ;
14498 052312 012746 005726      MOV      #WRD.9,-(SP) ;
14499 052316 012746 006630      MOV      #PHR.4,-(SP) ;
14500 052322 012746 005432      MOV      #SIX.FMT,-(SP) ;
14501 052326 012746 000007      MOV      #7,-(SP)
14502 052332 010600              MOV      SP,RO      ; SP,*
14503 052334 104414              TRAP     14
14504 052336 062706 000020      ADD      #20,SP      ;
14505 052342 062706 000010      22$: ADD      #10,SP  ;
14506 052346 000207      RTS      PC
14507
14508      ; Routine Size: 255 words
14509      ; Maximum stack depth per invocation: 19 words
14514
14515
14519
14523 052350              T30::
14524 052350 004767 176776      1$: JSR      PC,$T30 ;
14525 052354 104466              TRAP     66
14526 052356 006000              ROR      RO
14527 052360 103773              BLO     1$
14528 052362 000207      RTS      PC
14529
14530      ;ML4
14531      ;

```

22-Oct-1980 10:47:44 TOPS
22-Oct-1980 10:45:32 PA:<

22-Oct-1980 10:47:44
22-Oct-1980 10:45:32

TOPS-20 Bliss-16 V2(206)
PA:<NEALE>BL2ML4.BLI.2 (73)

14537 :ML4
14538 :
14539 :
14540 :
14541 :
14542 :
14543 :
14544 :
14545 :
14546 :
14547 :
14548 :
14549 :
14550 :
14551 :
14552 :
14553 :
14554 :
14555 :
14556 :
14557 :
14558 :
14559 :
14560 :
14561 :
14562 :
14563 :
14564 :
14565 :
14566 :
14567 :
14568 :
14569 :
14570 :
14571 :
14572 :
14573 :
14574 :
14575 :
14576 :
14577 :
14578 :
14579 :
14580 :
14581 :
14582 :
14583 :
14584 :
14585 :
14586 :
14587 :
14588 :
14589 :
14590 :
14591 :

7091
7092
7093
7094
7095
7096
7097
7098
7099
7100
7101
7102
7103
7104
7105
7106
7107
7108
7109
7110
7111
7112
7113
7114
7115
7116
7117
7118
7119
7120
7121
7122
7123
7124
7125
7126
7127
7128
7129
7130
7131
7132
7133
7134
7135
7136
7137
7138
7139
7140
7141
7142

BGNTST;

!++

TEST NUMBER: TST 31

TEST NAME: SYNC DATA BUS CONTINUITY TEST (WRITE PATH)

TEST DESCRIPTION:

TEST THE CONTINUITY OF THE
SYNCHRONOUS MODULE WRITE PATH
DATA BUS BY:

1. VIA DAT_DM MODE WRITE A
BACKGROUND PATTERN OF ONES
INTO THE GOOD BLOCK.
2. VIA MBUS WRITE FUNCTION
WRITE A ZEROES PATTERN INTO
THE GOOD BLOCK.
3. VIA DAT_DM MODE READ GOOD
NIBBLES IN THE GOOD BLOCK FOR
ZEROES.
4. REPEAT WITH COMPLIMENT
DATA AND BACKGROUND PATTERNS.

IMPLICIT INPUTS:

PD_TEMP

A BIT VECTOR OF 16 BITS WHERE
THE READ PROM DATA IS STORED
AND ACCESSED FROM.

GLOBAL OWN LOCATION TO THIS TST.

IO_BUF

A VECTOR OF 256 WORDS WHERE
DATA FOR MBUS READS AND WRITE
FUNCTION ARE FOUND.

A GLOBAL OWN LOCATION TO
THIS TEST.

--

22-Oct-1980 10:47:44 TOPS-20 Bliss-16 V2(206)
22-Oct-1980 10:45:32 PA:<NEALE>BL2ML4.BLI.2 (73)

```

14593 :ML4
14594 :
14595 :
14596 :      7143 local
14597 :      7144     DODU FLG,
14598 :      7145     BG PAT,
14599 :      7146     RESULT,
14600 :      7147     TST_PAT;
14601 :      7148
14602 :      7149 DODU FLG = ZERO;
14603 :      7150 BG PAT = ZEROES;
14604 :      7151 TST_PAT = ONES;
14605 :      7152
14606 :      7153 incr TWICE from 0 to 1 do
14607 :      7154     begin
14608 :      7155     BGNSUB;
14609 :      7156     CLR_MBUS;
14610 :      7157     MLD1 = .BG_PAT;
14611 :      7158     MLD2 = .BG_PAT;
14612 :      7159     MLE2 = .BG_PAT;
14613 :      7160     DAT_DM_XFER ();
14614 :      7161     MLC51 = write;
14615 :      7162
14616 :      7163     incr WD_CNT from 0 to 127 do
14617 :      7164         begin
14618 :      7165         DELAY (ONE_US);
14619 :      7166         DAT_CLK = ONE;
14620 :      7167         end;
14621 :      7168
14622 :      7169     CLR_MBUS;
14623 :      7170     BAI = ONE;
14624 :      7171     IO_BUF = .TST_PAT;
14625 :      7172     GD_BLK_XFER ();
14626 :      7173     MLC51 = write;
14627 :      7174
14628 :      7175     do
14629 :      7176         0
14630 :      7177     until .DRY IS_SET;
14631 :      7178
14632 :      7179     CLR_MBUS;
14633 :      7180     DAT_DM_XFER ();
14634 :      7181     MLC51 = read;
14635 :      7182     DELAY (ONE_US);
14636 :      7183
14637 :      7184     incr WD_CNT from 0 to 112 do
14638 :      7185         begin
14639 :      7186         PD_TEMP = .MLPD;
14640 :      7187         DAT_CLK = ONE;
14641 :      7188         DELAY (ONE_US);
14642 :      7189         RD_LNG_WRD;
14643 :      7190
14644 :      7191         incr NIB_PTR from 0 to 8 do
14645 :      7192             if .PD_TEMP [.NIB_PTR] IS_NOT_SET
14646 :      7193                 then
14647 :      7194

```

```

!DROP UNIT FLAG
!BACKGROUND PATTERN
!RESULTS FROM XOR
!TEST PATTERN

!BG PAT EQL 0'S
!TST PAT EQL 1'S

!REPEAT LOOP TWICE

!LOAD DATA DIAG REG WITH BG PAT

!SET UP A DATA DIAG XFERR AT THE GOOD BLK
!DO A WRITE FUNCTION

!LOAD BLOCK WITH BG PAT

!SET ON FIRST IO BUF ADRS
!FIRST IO_BUF WORD EQL'S TST_PAT
!SET UP A GOOD BLOCK XFERR
!DO A WRITE FUNCTION

!DELAY UNTIL XFERR TO COMPLETE

!SET UP A DATA DIAG XFERR AT SAME BLOCK
!DO A READ FUNCTION

!READ 113 LONG WORDS

!GET THE PROM DATA
!CLOCK OUT THE DATA WORD

!READ THE DATA WORD

!LOOK AT 9 NIBBLES

!FIND GOOD NIBBLES

```

22-Oct-1980 10:47:44 TOPS-20 Bliss-16 V2(206)
22-Oct-1980 10:45:32 PA:<NEALE>BL2ML4.BLI.2 (73)

```

14649 :ML4
14650 :
14651 :
14652 :      7195      begin
14653 :      7196      XOR_LNG_WRD (.NIB_PTR, .TST_PAT, RESULT);      !XOR NIBBLE DATA WITH TST_PAT
14654 :      7197
14655 :      7198      if .RESULT<0, 4> neq ZERO      !SEE IF EQUAL
14656 :      7199      then
14657 :      7200      begin      !ERROR IF NEQ
14658 :      7201      ERRDF (86, SYNC, 0);
14659 :      7202      PRINTB (FIV_FMT, WRD 24, WRD 25, WRD 10, WRD 12, FNC_5);
14660 :      7203      PRINTB (FMT_5, .TST_PAT, .RESULT, .NIB_PTR);
14661 :      7204      DODU_FLG = ONE;
14662 :      7205      end
14663 :      7206
14664 :      7207      end
14665 :      7208
14666 :      7209      end;
14667 :      7210
14668 :      7211      ENDSUB;
14669 :      7212
14670 :      7213      if .DODU_FLG IS_SET      !DROP THIS UNIT IF DODU IS_SET
14671 :      7214      !AND EXITS TEST
14672 :      7215      then
14673 :      7216      begin
14674 :      7217      DODU (.ML_LUN);
14675 :      7218      DOCLN;
14676 :      7219      end;
14677 :      7220
14678 :      7221      BG_PAT = not .BG_PAT;      !COMPLIMENT BG_PAT
14679 :      7222      TST_PAT = not .TST_PAT;      !COMPLIMENT TST_PAT AND REPEAT
14680 :      7223      end;
14681 :      7224
14682 :      7225      ENDTST;
14686
14690 052364 004167 131470      $T31: JSR      R1,$SAVE5      ;      7089
14691 052370 024646      CMP      -(SP),-(SP)
14692 052372 005046      CLR      -(SP)      ; DODU.FLG      7149
14693 052374 005001      CLR      R1      ; BG.PAT      7150
14694 052376 012702 177777      MOV      #-1,R2      ; *,TST.PAT      7151
14695 052402 005046      CLR      -(SP)      ; TWICE      7153
14696 052404 104402      TRAP     2      ;      7154
14697 052406 152777 000040 137324      BISB     #40,@ML.REG+40      ;      7155
14698 052414 016705 137550      MOV      ML.DUT,R5
14699 052420 042705 177770      BIC      #177770,R5
14700 052424 142777 000007 137306      BICB     #7,@ML.REG+40
14701 052432 150577 137302      BISB     R5,@ML.REG+40
14702 052436 010177 137426      MOV      R1,@ML.REG+170      ; BG.PAT,*      7157

```


Address	Instruction	Comments	Sequence	Address	Instruction	Comments	Sequence	Address	Instruction	Comments	Sequence
14704											
14705											
14706											
14707	052442	010177	137432		MOV	R1,@ML.REG+200			:	BG.PAT,*	7158
14708	052446	010177	137406		MOV	R1,@ML.REG+160			:	BG.PAT,*	7159
14709	052452	004767	140150		JSR	PC,DAT.DM.XFER			:		7160
14710	052456	012777	000061	137214	MOV	#61,@ML.REG			:		7161
14711	052464	005003			CLR	R3			:	WD.CNT	7163
14712	052466	012704	000001	2\$:	MOV	#1,R4			:	*,SSTMP2	7165
14713	052472	001411		3\$:	BEQ	6\$:		
14714	052474	016705	127416		MOV	L\$DLY,R5			:	*,SSTMP1	
14715	052500	001404			BEQ	5\$:		
14716	052502	005066	000006	4\$:	CLR	6(SP)			:	SSTMP	
14717	052506	005305			DEC	R5			:	SSTMP1	
14718	052510	001374			BNE	4\$:		
14719	052512	005304		5\$:	DEC	R4			:	SSTMP2	
14720	052514	000766			BR	3\$:		
14721	052516	152777	000020	137274	6\$:	BISB	#20,@ML.REG+120		:		7166
14722	052524	005203			INC	R3			:	WD.CNT	7163
14723	052526	020327	000177		CMP	R3,#177			:	WD.CNT,*	
14724	052532	003755			BLE	2\$:		
14725	052534	152777	000040	137176	BISB	#40,@ML.REG+40			:		7167
14726	052542	016705	137422		MOV	ML.DUT,R5			:		
14727	052546	042705	177770		BIC	#177770,R5			:		
14728	052552	142777	000007	137160	BICB	#7,@ML.REG+40			:		
14729	052560	150577	137154		BISB	R5,@ML.REG+40			:		
14730	052564	152777	000010	137146	BISB	#10,@ML.REG+40			:		7170
14731	052572	010267	135544		MOV	R2,IO.BUF			:	TST.PAT,*	7171
14732	052576	004767	137740		JSR	PC,GD.BLK.XFER			:		7172
14733	052602	012777	000061	137070	MOV	#61,@ML.REG			:		7173
14734	052610	105777	137134	7\$:	TSTB	@ML.REG+50			:		7177
14735	052614	100375			BPL	7\$:		
14736	052616	152777	000040	137114	BISB	#40,@ML.REG+40			:		
14737	052624	016705	137340		MOV	ML.DUT,R5			:		
14738	052630	042705	177770		BIC	#177770,R5			:		
14739	052634	142777	000007	137076	BICB	#7,@ML.REG+40			:		
14740	052642	150577	137072		BISB	R5,@ML.REG+40			:		
14741	052646	004767	137754		JSR	PC,DAT.DM.XFER			:		7180
14742	052652	012777	000071	137020	MOV	#71,@ML.REG			:		7181
14743	052660	012704	000001		MOV	#1,R4			:	*,SSTMP2	7182
14744	052664	001411		8\$:	BEQ	11\$:		
14745	052666	016705	127224		MOV	L\$DLY,R5			:	*,SSTMP1	
14746	052672	001404			BEQ	10\$:		
14747	052674	005066	000006	9\$:	CLR	6(SP)			:	SSTMP	
14748	052700	005305			DEC	R5			:	SSTMP1	
14749	052702	001374			BNE	9\$:		
14750	052704	005304		10\$:	DEC	R4			:	SSTMP2	
14751	052706	000766			BR	8\$:		
14752	052710	005003		11\$:	CLR	R3			:	WD.CNT	7184
14753	052712	017767	137212	136742	12\$:	MOV	@ML.REG+230,PD.TEMP		:		7186
14754	052720	152777	000020	137072	BISB	#20,@ML.REG+120			:		7187
14755	052726	012704	000001		MOV	#1,R4			:	*,SSTMP2	7188
14756	052732	001411		13\$:	BEQ	16\$:		
14757	052734	016705	127156		MOV	L\$DLY,R5			:	*,SSTMP1	
14758	052740	001404			BEQ	15\$:		

22-Oct-1980 10:47:44 TOPS
22-Oct-1980 10:45:32 PA:<

```

14760          :ML4
14761          :
14762          :
14763 052742 005066 000006 14$: CLR 6(SP) ; S$TMP
14764 052746 005305          DEC R5 ; S$TMP1
14765 052750 001374          BNE 14$
14766 052752 005304 15$: DEC R4 ; S$TMP2
14767 052754 000766          BR 13$
14768 052756 017767 137106 134730 16$: MOV @ML.REG+170,D1.TEMP
14769 052764 017767 137110 134724 MOV @ML.REG+200,D2.TEMP
14770 052772 017767 137062 134720 MOV @ML.REG+160,E2.TEMP
14771 053000 005004          CLR R4 ; NIB.PTR 7191
14772 053002 010405 17$: MOV R4,R5 ; NIB.PTR,* 7193
14773 053004 006205          ASR R5
14774 053006 006205          ASR R5
14775 053010 006205          ASR R5
14776 053012 062705 011662 ADD #PD.TEMP,R5
14777 053016 010546          MOV R5,-(SP)
14778 053020 010446          MOV R4,-(SP) ; NIB.PTR,*
14779 053022 042716 177770 BIC #177770,(SP)
14780 053026 012746 000001 MOV #1,-(SP)
14781 053032 005046          CLR -(SP)
14782 053034 004767 130042 JSR PC,BL$GT2
14783 053040 062706 000010 ADD #10,SP
14784 053044 005700          TST R0
14785 053046 001060          BNE 19$
14786 053050 010446          MOV R4,-(SP) ; NIB.PTR,* 7196
14787 053052 010246          MOV R2,-(SP) ; TST.PAT,*
14788 053054 012746 000012 MOV #12,-(SP)
14789 053060 060616          ADD SP,(SP) ; RESULT,*
14790 053062 004767 140266 JSR PC,XOR.LNG.WRD
14791 053066 032766 000017 000012 BIT #17,12(SP) ; *,RESULT 7198
14792 053074 001443          BEQ 18$
14793 053076 104455          TRAP 55 ; 7201
14794 053100 000126          .WORD 126
14795 053102 007500          .WORD SYNC
14796 053104 000000          .WORD 0
14797 053106 012746 007020 MOV #FNC.5,-(SP) ; 7202
14798 053112 012746 005760 MOV #WRD.12,-(SP)
14799 053116 012746 005740 MOV #WRD.10,-(SP)
14800 053122 012746 006112 MOV #WRD.25,-(SP)
14801 053126 012746 006104 MOV #WRD.24,-(SP)
14802 053132 012746 005414 MOV #F IV.FMT,-(SP)
14803 053136 012746 000006 MOV #6,-(SP)
14804 053142 010600          MOV SP,R0 ; SP,*
14805 053144 104414          TRAP 14
14806 053146 010416          MOV R4,(SP) ; NIB.PTR,* 7203
14807 053150 016646 000030 MOV 30(SP),-(SP) ; RESULT,*
14808 053154 010246          MOV R2,-(SP) ; TST.PAT,*
14809 053156 012746 004366 MOV #FMT.5,-(SP)
14810 053162 012746 000004 MOV #4,-(SP)
14811 053166 010600          MOV SP,R0 ; SP,*
14812 053170 104414          TRAP 14
14813 053172 012766 000001 000036 MOV #1,36(SP) ; *,DODU.FLG 7204
14814 053200 062706 000026 ADD #26,SP ; 7200
  
```

14816										22-Oct-1980 10:47:44	TOPS
14817				:ML4						22-Oct-1980 10:45:32	PA:<
14818				:							
14819	053204	062706	000006	18\$:	ADD	#6,SP		:			7195
14820	053210	005204		19\$:	INC	R4		:	NIB.PTR		7191
14821	053212	020427	000010		CMP	R4,#10		:	NIB.PTR,*		
14822	053216	003671			BLE	17\$					
14823	053220	005203			INC	R3		:	WD.CNT		7184
14824	053222	020327	000160		CMP	R3,#160		:	WD.CNT,*		
14825	053226	003631			BLE	12\$					
14826	053230	104467			TRAP	67		:			7209
14827	053232	006000			ROR	R0					
14828	053234	103002			BHIS	21\$					
14829	053236	000167	177142	20\$:	JMP	1\$					
14830	053242	026627	000002	000001	21\$:	CMP	2(SP),#1		:	DODU.FLG,*	7213
14831	053250	001004			BNE	22\$					
14832	053252	016700	136710		MOV	ML.LUN,R0		:			7217
14833	053256	104451			TRAP	51					
14834	053260	104444			TRAP	44					
14835	053262	005101		22\$:	COM	R1		:	BG.PAT		7221
14836	053264	005102			COM	R2		:	TST.PAT		7222
14837	053266	005216			INC	(SP)		:	TWICE		7153
14838	053270	021627	000001		CMP	(SP),#1		:	TWICE,*		
14839	053274	003760			BLE	20\$					
14840	053276	062706	000010		ADD	#10,SP		:			7089
14841	053302	000207			RTS	PC					

: Routine Size: 232 words
 : Maximum stack depth per invocation: 24 words

14854											
14858	053304			T31::							
14859	053304	004767	177054	1\$:	JSR	PC,\$T31		:			7223
14860	053310	104466			TRAP	66					
14861	053312	006000			ROR	R0					
14862	053314	103773			BLO	1\$					
14863	053316	000207			RTS	PC					

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

14876 : 7226 !<BLF/PAGE>

22-Oct-1980 10:47:44
 22-Oct-1980 10:45:32

TOPS-20 Bliss-16 V2(206)
 PA:<NEALE>BL2ML4.BLI.2 (74)

```

14878 :ML4
14879 :
14880 :
14881 : 7227 !
14882 : 7228 BGNTST;
14883 : 7229
14884 : 7230 !++
14885 : 7231 TEST NUMBER: TST 32
14886 : 7232
14887 : 7233 TEST NAME: SYNC DATA BUS CONIINUITY /READ PATH
14888 : 7234
14889 : 7235 TEST DESCRIPTION:
14890 : 7236 TEST THE CONITNUITY OF THE SYNCHRONOUS MODULE READ
14891 : 7237 DATA BUS BY:
14892 : 7238
14893 : 7239 1. VIA MBUS WRITE FUNCTION WRITE ONES INTO THE GOOD BLOCK.
14894 : 7240
14895 : 7241 2. VIA MBUS READ FUNCTION READ THE GOOD BLOCK FOR ONES.
14896 : 7242
14897 : 7243 3. REPEAT WITH COMPLIMENT DATA PATTERN.
14898 : 7244
14899 : 7245 IMPLICIT INPUTS:
14900 : 7246 IO_BUF
14901 : 7247 A VECTOR OF 256 WORDS WHERE DATA FOR MBUS READS AND WRITE
14902 : 7248 FUNCTION ARE FOUND.
14903 : 7249 !--
14904 : 7250
14905 : 7251 Local
14906 : 7252 DODU_FLG, !DROP UNIT FLAG
14907 : 7253 TST_PAT, !TEST PATTERN
14908 : 7254 BG_PAT; !BACKGROUND PATTERN
14909 : 7255
14910 : 7256 DODU_FLG = ZERO;
14911 : 7257 TST_PAT = ONES;
14912 : 7258 BG_PAT = ZEROES;
14913 : 7259
14914 : 7260 incr TWICE from 0 to 1 do !REPEAT LOOP TWICE
14915 : 7261 begin
14916 : 7262 CLR_MBUS;
14917 : 7263 BAI = ONE; !SET ON FIRST IO_BUF ADRS
14918 : 7264 IO_BUF = .TST_PAT; !FIRST IO_BUF ADRS GET TST_PAT
14919 : 7265 GD_BLK_XFER (); !SET UP A GOOD BLOCK XFERR
14920 : 7266 MLCS1 = write; !DO A WRITE FUNCTION (WRITES THE TST_PAT)
14921 : 7267
14922 : 7268 do !DELAY UNTIL XFER TO COMPLETE
14923 : 7269 0
14924 : 7270 until .DRY IS_SET;
14925 : 7271
14926 : 7272 BGNSUB;
14927 : 7273 CLR_MBUS;
14928 : 7274
14929 : 7275 incr IO_CNT from 0 to 255 do !LOAD IO_BUF WITH BG PAT
14930 : 7276 IO_BUF [.IO_CNT] = .BG_PAT;
14931 : 7277
14932 : 7278 GD_BLK_XFER (); !SET UP A GOOD BLOCK XFERR
  
```

14934 :ML4

22-Oct-1980 10:47:44 TOPS-20 Bliss-16 V2(206)
22-Oct-1980 10:45:32 PA:<NEALE>BL2ML4.BLI.2 (74)

```

14935 :
14936 :
14937 :       7279   MLCS1 = read;
14938 :       7280
14939 :       7281   do
14940 :       7282     0
14941 :       7283   until .DRY IS_SET;
14942 :       7284
14943 :       7285   incr IO_CNT from 0 to 255 do
14944 :       7286
14945 :       7287     if .IO_BUF [.IO_CNT] neq .TST_PAT
14946 :       7288     then
14947 :       7289       begin
14948 :       7290         ERRDF (87, SYNC, 0);
14949 :       7291         PRINTB (SEV_FMT, WRD_24, WRD_25, WRD_10, WRD_12, WRD_23, FNC_6, WRD_19);
14950 :       7292         PRINTB (FMT_2, .TST_PAT, .IO_BUF [.IO_CNT], (.TST_PAT xor .IO_BUF [.IO_CNT]));
14951 :       7293         DODU_FLG = ONE;
14952 :       7294       end;
14953 :       7295
14954 :       7296   ENDSUB;
14955 :       7297
14956 :       7298   if .DODU_FLG IS_SET
14957 :       7299   then
14958 :       7300     begin
14959 :       7301       DODU (.ML_LUN);
14960 :       7302       DOCLN;
14961 :       7303     end;
14962 :       7304
14963 :       7305   TST_PAT = not .TST_PAT;
14964 :       7306   BG_PAT = not .BG_PAT;
14965 :       7307   end;
14966 :       7308
14967 :       7309   ENDTST;

```

```

!DO A READ FUNCTION (READ THE TST PAT)
!DELAY UNTIL XFER TO COMPLETE
!READ THE IO_BUF FOR TEST PATTERN
!SEE IF IO_BUF WORD EQLS TST PAT
!ERROR AND SET DODU_FLG IF NEQ
!DROP THIS UNIT IF DODU_FLG IS SET
!COMPLIMENT TST PAT
!COMPLIMENT BG_PAT AND REPEAT

```

14975	053320	004167	130534	\$T32:	JSR	R1,\$SAVE5	:	7225
14976	053324	005046			CLR	-(SP)	: DODU.FLG	7256
14977	053326	012701	177777		MOV	#-1,R1	: *.TST.PAT	7257
14978	053332	005004			CLR	R4	: BG.PAT	7258
14979	053334	005005			CLR	R5	: TWICE	7260
14980	053336	152777	000040	136374	1\$:	BISB	#40,@ML.REG+40	7261
14981	053344	016703	136620			MOV	ML.DUT,R3	
14982	053350	042703	177770			BIC	#177770,R3	
14983	053354	142777	000007	136356		BICB	#7,@ML.REG+40	
14984	053362	150377	136352			BISB	R3,@ML.REG+40	
14985	053366	152777	000010	136344		BISB	#10,@ML.REG+40	7263
14986	053374	010167	134742			MOV	R1,IO.BUF	7264
14987	053400	004767	137136			JSR	PC,GD.BLK.XFER	7265

```

14989           ;ML4
14990           ;
14991           ;
14992 053404 012777 000061 136266      MOV    #61,@ML.REG           ;
14993 053412 105777 136332      2$:  TSTB  @ML.REG+50         ;
14994 053416 100375              BPL   2$                    ;
14995 053420 104402              3$:  TRAP  2                    ;
14996 053422 152777 000040 136310      BISB  #40,@ML.REG+40       ;
14997 053430 016703 136534      MOV    ML.DUT,R3           ;
14998 053434 042703 177770      BIC   #177770,R3          ;
14999 053440 142777 000007 136272      BICB  #7,@ML.REG+40       ;
15000 053446 150377 136266      BISB  R3,@ML.REG+40       ;
15001 053452 005002              CLR   R2                    ; IO.CNT
15002 053454 010203              4$:  MOV    R2,R3            ; IO.CNT,*
15003 053456 006303              ASL   R3                    ;
15004 053460 010463 010342      MOV    R4,IO.BUF(R3)       ; BG.PAT,*
15005 053464 005202              INC   R2                    ; IO.CNT
15006 053466 020227 000377      CMP   R2,#377              ; IO.CNT,*
15007 053472 003770              BLE  4$                     ;
15008 053474 004767 137042      JSR   PC,GD.BLK.XFER       ;
15009 053500 012777 000071 136172      MOV    #71,@ML.REG         ;
15010 053506 105777 136236      5$:  TSTB  @ML.REG+50         ;
15011 053512 100375              BPL   5$                     ;
15012 053514 005002              CLR   R2                    ; IO.CNT
15013 053516 010203              6$:  MOV    R2,R3            ; IO.CNT,*
15014 053520 006303              ASL   R3                    ;
15015 053522 062703 010342      ADD   #IO.BUF,R3           ;
15016 053526 021301              CMP   (R3),R1              ; *,TST.PAT
15017 053530 001454              BEQ   7$                     ;
15018 053532 104455              TRAP  55                    ;
15019 053534 000127              .WORD 127                   ;
15020 053536 007500              .WORD SYNC                  ;
15021 053540 000000              .WORD 0                     ;
15022 053542 012746 006040      MOV    #WRD.19,-(SP)        ;
15023 053546 012746 007030      MOV    #FNC.6,-(SP)        ;
15024 053552 012746 006076      MOV    #WRD.23,-(SP)        ;
15025 053556 012746 005760      MOV    #WRD.12,-(SP)        ;
15026 053562 012746 005740      MOV    #WRD.10,-(SP)        ;
15027 053566 012746 006112      MOV    #WRD.25,-(SP)        ;
15028 053572 012746 006104      MOV    #WRD.24,-(SP)        ;
15029 053576 012746 005452      MOV    #SEV.FMT,-(SP)       ;
15030 053602 012746 000010      MOV    #10,-(SP)           ;
15031 053606 010600              MOV    SP,R0                ; SP,*
15032 053610 104414              TRAP  14                    ;
15033 053612 011316              MOV    (R3),(SP)           ;
15034 053614 010146              MOV    R1,-(SP)            ; TST.PAT,*
15035 053616 046616 000002      BIC   2(SP),(SP)           ;
15036 053622 040166 000002      BIC   R1,2(SP)             ; TST.PAT,*
15037 053626 052616              BIS   (SP)+,(SP)           ;
15038 053630 011346              MOV    (R3),-(SP)          ;
15039 053632 010146              MOV    R1,-(SP)            ; TST.PAT,*
15040 053634 012746 004224      MOV    #FMT.2,-(SP)        ;
15041 053640 012746 000004      MOV    #4,-(SP)            ;
15042 053644 010600              MOV    SP,R0                ; SP,*
15043 053646 104414              TRAP  14

```

22-Oct-1980 10:47:44 TOPS
22-Oct-1980 10:45:32 PA:<

7266
7270
7272
7275
7276
7275
7278
7279
7283
7285
7287
7290
7291
7292

```

15045 ;ML4
15046 ;
15047 ;
15048 053650 012766 000001 000032 MOV #1,32(SP) ; *,DODU.FLG 7293
15049 053656 062706 000032 ADD #32,SP ; 7289
15050 053662 005202 7$: INC R2 ; IO.CNT 7285
15051 053664 020227 000377 CMP R2,#377 ; IO.CNT,*
15052 053670 003712 BLE 6$ ;
15053 053672 104467 TRAP 67 ; 7294
15054 053674 006000 ROR R0 ;
15055 053676 103650 BLO 3$ ;
15056 053700 021627 000001 CMP (SP),#1 ; DODU.FLG,* 7298
15057 053704 001004 BNE 8$ ;
15058 053706 016700 136254 MOV ML.LUN,R0 ; 7301
15059 053712 104451 TRAP 51 ;
15060 053714 104444 TRAP 44 ;
15061 053716 005101 8$: COM R1 ; TST.PAT 7305
15062 053720 005104 COM R4 ; BG.PAT 7306
15063 053722 005205 INC R5 ; TWICE 7260
15064 053724 020527 000001 CMP R5,#1 ; TWICE,*
15065 053730 003602 BLE 1$ ;
15066 053732 005726 TST (SP)+ ; 7225
15067 053734 000207 RTS PC ;
15068 ;
15069 ; Routine Size: 135 words
15070 ; Maximum stack depth per invocation: 20 words
15075 ;
15076 ;
15080 ;
15084 053736 T32::
15085 053736 004767 177356 1$: JSR PC,$T32 ; 7307
15086 053742 104466 TRAP 66 ;
15087 053744 006000 ROR R0 ;
15088 053746 103773 BLO 1$ ;
15089 053750 000207 RTS PC ;
15090 ;
15091 ; Routine Size: 6 words
15092 ; Maximum stack depth per invocation: 0 words
15097 ;
15098 ; 7310 !<BLF/PAGE>

```

22-Oct-1980 10:47:44
22-Oct-1980 10:45:32

TOPS-20 Bliss-16 V2(206)
PA:<NEALE>BL2ML4.BLI.2 (75)

```

15100 :ML4
15101 :
15102 :
15103 : 7311 !
15104 : 7312 !
15105 : 7313 BGNTST;
15106 : 7314
15107 : 7315 !++
15108 : 7316 TEST NUMBER: TST 33
15109 : 7317
15110 : 7318 TEST NAME: RAM BUS ADRS COUNTER TEST /WRITE PATH
15111 : 7319
15112 : 7320 TEST DESCRIPTION:
15113 : 7321
15114 : 7322 TEST ABILITY OF THE RAM_BUS ADDRESS
15115 : 7323 COUNTERS TO LOAD/UNLOAD THE SKIP
15116 : 7324 RAM DURING WRITE FUNCTIONS BY:
15117 : 7325
15118 : 7326 1. LOADING A REPEATING COUNT OF 0
15119 : 7327 TO 63 INTO THE NIBBLES OF THE
15120 : 7328 FIRST 64 WORDS OF THE IO_BUF.
15121 : 7329
15122 : 7330 2. VIA MBUS WRITE FUNCTION WRITE
15123 : 7331 THE CONTENTS OF THE IO_BUF
15124 : 7332 INTO THE GOOD BLOCK.
15125 : 7333
15126 : 7334 3. VIA DAT_DM READ GOOD NIBBLES IN
15127 : 7335 THE GOOD BLOCK FOR THE UNBROKEN
15128 : 7336 COUNT OF 0 TO 63.
15129 : 7337
15130 : 7338 ONCE A BAD NIBBLE IS ENCOUNTERED
15131 : 7339 MASK THAT NIBBLE FROM FURTHER
15132 : 7340 READS.
15133 : 7341
15134 : 7342 4. REPEAT READING NIBBLES UNTIL
15135 : 7343 113 WORDS ARE READ OR ALL 10 NIBBLES
15136 : 7344 ARE MASKED.
15137 : 7345
15138 : 7346 IMPLICIT INPUTS:
15139 : 7347
15140 : 7348 PD_TEMP
15141 : 7349
15142 : 7350 A BIT VECTOR OF 16 BITS WHERE
15143 : 7351 THE READ PROM DATA IS STORED
15144 : 7352 AND ACCESSED FROM.
15145 : 7353 GLOBAL OWN LOCATION TO THIS
15146 : 7354 TST.
15147 : 7355
15148 : 7356 IO_BUF
15149 : 7357 A VECTOR OF 256 WORDS WHERE
15150 : 7358 DATA FOR MBUS READS AND WRITE
15151 : 7359 FUNCTION ARE FOUND.
15152 : 7360
15153 : 7361 GLOBAL OWN LOCATION TO THIS TEST.
15154 : 7362

```


22-Oct-1980 10:47:44
 22-Oct-1980 10:45:32

TOPS-20 Bliss-16 V2(206)
 PA:<NEALE>BL2ML4.BLI.2 (75)

```

15156 :ML4
15157 :
15158 :
15159 : 7363 !
15160 : 7364 !--
15161 : 7365
15162 : 7366 local
15163 : 7367 DODU_FLG, !DROP UNIT FLAG
15164 : 7368 WRD_CNT, !WORD COUNT
15165 : 7369 NIB_IGNORE : bitvector [16], !NIBBLE IGNORE FLAGS
15166 : 7370 ERR_FLG, !ERROR FLAG
15167 : 7371 BAD_NIB_CNT, !BAD NIBBLE COUNT
15168 : 7372 PASS_CNT, !PASS COUNT
15169 : 7373 NIB_PAT; !NIBBLE PATTERN
15170 : 7374
15171 : 7375 DODU_FLG = ZERO;
15172 : 7376 WRD_CNT = ZERO;
15173 : 7377
15174 : 7378 incr CNT from 0 to 63 do !LOAD 64 WORDS WITH REPEATING COUNTS OF 0-63
15175 : 7379
15176 : 7380 incr PAT_CNT from -1 to 11 by 4 do !LOAD NIBBLES IN WORD WITH REPEATING COUNTS OF 0-63
15177 : 7381 begin
15178 : 7382 (IO_BUF [.WRD_CNT])<0, 4> = .PAT_CNT + 1; !LOAD FIRST NIBBLE IN WORD
15179 : 7383 (IO_BUF [.WRD_CNT])<4, 4> = .PAT_CNT + 2; !LOAD SECOND NIBBLE IN WORD
15180 : 7384 (IO_BUF [.WRD_CNT])<8, 4> = .PAT_CNT + 3; !LOAD THIRD NIBBLE IN WORD
15181 : 7385 (IO_BUF [.WRD_CNT])<12, 4> = .PAT_CNT + 4; !LOAD FORTH NIBBLE IN WORD
15182 : 7386 WRD_CNT = .WRD_CNT + 1; !INCREMENT TO NEXT WORD
15183 : 7387 end;
15184 : 7388
15185 : 7389 BGNSUB;
15186 : 7390 CLR MBUS;
15187 : 7391 GD_BLK_XFER (); !SET UP A GOOD BLOCK XFERR
15188 : 7392 MLC$1 = write; !DO A WRITE FUNCTION
15189 : 7393
15190 : 7394 do !DELAY UNTIL XFER TO COMPLETE
15191 : 7395 0
15192 : 7396 until .DRY IS_SET;
15193 : 7397
15194 : 7398 CLR MBUS;
15195 : 7399 NIB_IGNORE = ZEROES;
15196 : 7400 PASS_CNT = ZEROES;
15197 : 7401 NIB_PAT = ZEROES;
15198 : 7402 BAD_NIB_CNT = ZEROES;
15199 : 7403 DAT_DM_XFER (); !SET UP A DATA DIAG MODE AT THE GOOD BLOCK
15200 : 7404 MLC$1 = read; !DO A READ FUNCTION
15201 : 7405 DELAY (ONE_US);
15202 : 7406
15203 : 7407 do !LOOP UNTIL THE BLOCK IS READ OR 9 BAD NIBBLES FOUND
15204 : 7408 begin
15205 : 7409 PD_TEMP = .MLPD; !GET THE PROM DATA
15206 : 7410 DAT_CLK = ONE; !CLOCK OUT THE DATA WORD
15207 : 7411 DELAY (ONE_US);
15208 : 7412 RD_LNG_WRD; !READ THE DATA DIAG REGISTERS
15209 : 7413
15210 : 7414 incr NIB_PTR from 0 to 8 do !LOOK AT 9 NIBBLES

```

22-Oct-1980 10:47:44 TOPS-20 Bliss-16 V2(206)
22-Oct-1980 10:45:32 PA:<NEALE>BL2ML4.BLI.2 (75)

```

15212 :ML4
15213 :
15214 :
15215 :      7415      begin
15216 :      7416
15217 :      7417      if .PD_TEMP [.NIB_PTR] IS_NOT_SET      !FIND GOOD NIBBLES
15218 :      7418      then
15219 :      7419          begin
15220 :      7420
15221 :      7421          if .NIB_IGNORE [.NIB_PTR] IS_NOT_SET      !SEE IF THIS NIBBLE FOUND BAD BEFORE
15222 :      7422          then
15223 :      7423              begin
15224 :      7424                  TST_LNG_WRD (.NIB_PTR, .NIB_PAT, ERR_FLG);      !TEST THE NIBBLE IF NEVER FOUND BAD
15225 :      7425
15226 :      7426                  if .ERR_FLG IS_SET      !SEE IF TEST FOUND AN ERROR
15227 :      7427                  then
15228 :      7428                      begin      !ERROR AND SET DODU_FLG IS SET
15229 :      7429                          ERRDF (88, ARR_DAT, 0);
15230 :      7430                          PRINTB (SIX_FMT, FNC_18, WRD_50, WRD_10, WRD_12, FNC_5, WRD_19);
15231 :      7431                          DODU_FLG = ONE;
15232 :      7432                          end
15233 :      7433                      end
15234 :      7434                  end
15235 :      7435
15236 :      7436      else
15237 :      7437          !THIS NIBBLE IS BAD
15238 :      7438          begin
15239 :      7439              NIB_IGNORE [.NIB_PTR] = ONE;      !SET THIS NIBBLE NIB IGNORE FLAG
15240 :      7440              BAD_NIB_CNT = .BAD_NIB_CNT + 1;      !INCREMENT BAD NIB COUNT
15241 :      7441          end;
15242 :      7442
15243 :      7443          NIB_PAT = .NIB_PAT + 1;      !INCREMENT NIB PAT
15244 :      7444          end;
15245 :      7445
15246 :      7446          PASS_CNT = .PASS_CNT + 1;      !INCREMENT PASS COUNT
15247 :      7447      end
15248 :      7448      until (.PASS_CNT eql 113) or (.BAD_NIB_CNT eql 9);      !REPEAT UNTIL COMPLETE
15249 :      7449
15250 :      7450      ENDSUB;
15251 :      7451
15252 :      7452      if .DODU_FLG IS_SET      !DROP THIS UNIT IF DODU_FLG SET
15253 :      7453      then
15254 :      7454          begin
15255 :      7455              DODU (.ML_LUN);
15256 :      7456              DOCLN;
15257 :      7457          end;
15258 :      7458
15259 :      7459      ENDTST;
15263 :

```

Address	OpCode	Operand 1	Operand 2	Label	Instruction	Comments	Seq
15271	053752	004167	130102	\$T33:	JSR R1,\$SAVE5	:	7309
15272	053756	162706	000006		SUB #6,SP	:	
15273	053762	005046			CLR -(SP)	: DODU.FLG	7375
15274	053764	005003			CLR R3	: WRD.CNT	7376
15275	053766	005002			CLR R2	: CNT	7378
15276	053770	012700	177777	1\$:	MOV #-1,R0	: *,PAT.CNT	7380
15277	053774	010301		2\$:	MOV R3,R1	: WRD.CNT,*	7382
15278	053776	006301			ASL R1		
15279	054000	062701	010342		ADD #10.BUF,R1		
15280	054004	010005			MOV R0,R5	: PAT.CNT,*	
15281	054006	005205			INC R5		
15282	054010	042705	177760		BIC #177760,R5		
15283	054014	142711	000017		BICB #17,(R1)		
15284	054020	150511			BISB R5,(R1)		
15285	054022	010005			MOV R0,R5	: PAT.CNT,*	7383
15286	054024	062705	000002		ADD #2,R5		
15287	054030	006305			ASL R5		
15288	054032	006305			ASL R5		
15289	054034	006305			ASL R5		
15290	054036	006305			ASL R5		
15291	054040	042705	177417		BIC #177417,R5		
15292	054044	142711	000360		BICB #360,(R1)		
15293	054050	150511			BISB R5,(R1)		
15294	054052	010005			MOV R0,R5	: PAT.CNT,*	7384
15295	054054	062705	000003		ADD #3,R5		
15296	054060	000305			SWAB R5		
15297	054062	042705	170377		BIC #170377,R5		
15298	054066	042711	007400		BIC #7400,(R1)		
15299	054072	050511			BIS R5,(R1)		
15300	054074	010005			MOV R0,R5	: PAT.CNT,*	7385
15301	054076	062705	000004		ADD #4,R5		
15302	054102	000305			SWAB R5		
15303	054104	006305			ASL R5		
15304	054106	006305			ASL R5		
15305	054110	006305			ASL R5		
15306	054112	006305			ASL R5		
15307	054114	042705	007777		BIC #7777,R5		
15308	054120	042711	170000		BIC #170000,(R1)		
15309	054124	050511			BIS R5,(R1)		
15310	054126	005203			INC R3	: WRD.CNT	7386
15311	054130	062700	000004		ADD #4,R0	: *,PAT.CNT	7380
15312	054134	020027	000013		CMP R0,#13	: PAT.CNT,*	
15313	054140	003715			BLE 2\$		
15314	054142	005202			INC R2	: CNT	7378
15315	054144	020227	000077		CMP R2,#77	: CNT,*	
15316	054150	003707			BLE 1\$		
15317	054152	104402		3\$:	TRAP 2	:	7387
15318	054154	152777	000040	135556	BISB #40,@ML.REG+40	:	7389
15319	054162	016702	136002		MOV ML.DUT,R2		
15320	054166	042702	177770		BIC #177770,R2		
15321	054172	142777	000007	135540	BICB #7,@ML.REG+40		


```
15435 ;ML4
15436 ;
15437 ;
15438 054716 020427 000161 CMP R4,#161 ; PASS.CNT,* 7448
15439 054722 001405 BEQ 17$
15440 054724 020527 000011 CMP R5,#11 ; BAD.NIB.CNT,*
15441 054730 001402 BEQ 17$
15442 054732 000167 177372 JMP 8$
15443 054736 104467 17$: TRAP 67
15444 054740 006000 ROR R0
15445 054742 103002 BHIS 18$
15446 054744 000167 177202 JMP 3$
15447 054750 021627 000001 18$: CMP (SP),#1 ; DODU.FLG,* 7452
15448 054754 001004 BNE 19$
15449 054756 016700 135204 MOV ML.LUN,R0 ; 7455
15450 054762 104451 TRAP 51
15451 054764 104444 TRAP 44
15452 054766 062706 000010 19$: ADD #10,SP ; 7309
15453 054772 000207 RTS PC
15454
15455 ; Routine Size: 265 words
15456 ; Maximum stack depth per invocation: 21 words
15461
15462
15466
15470 054774 T33::
15471 054774 004767 176752 1$: JSR PC,$T33 ; 7457
15472 055000 104466 TRAP 66
15473 055002 006000 ROR R0
15474 055004 103773 BLO 1$
15475 055006 000207 RTS PC
15476
15477 ; Routine Size: 6 words
15478 ; Maximum stack depth per invocation: 0 words
15483
15484
15485 ; 7460 !<BLF/PAGE>
```

22-Oct-1980 10:47:44
22-Oct-1980 10:45:32

TCPS-20 Bliss-16 V2(206)
PA:<NEALE>BL2ML4.BLI.2 (76)

15487 :ML4
15488 :
15489 :
15490 :
15491 :
15492 :
15493 :
15494 :
15495 :
15496 :
15497 :
15498 :
15499 :
15500 :
15501 :
15502 :
15503 :
15504 :
15505 :
15506 :
15507 :
15508 :
15509 :
15510 :
15511 :
15512 :
15513 :
15514 :
15515 :
15516 :
15517 :
15518 :
15519 :
15520 :
15521 :
15522 :
15523 :
15524 :
15525 :
15526 :
15527 :
15528 :
15529 :
15530 :
15531 :
15532 :
15533 :
15534 :
15535 :
15536 :
15537 :
15538 :
15539 :
15540 :
15541 :

7461
7462
7463
7464
7465
7466
7467
7468
7469
7470
7471
7472
7473
7474
7475
7476
7477
7478
7479
7480
7481
7482
7483
7484
7485
7486
7487
7488
7489
7490
7491
7492
7493
7494
7495
7496
7497
7498
7499
7500
7501
7502
7503
7504
7505
7506
7507
7508
7509
7510
7511
7512

BGNTST;

!++

TEST NUMBER: TST 34

TEST NAME: RAM BUS ADRS COUNTER TEST /READ PATH

TEST DESCRIPTION:

TEST ABILITY OF RAM/BUS ADRS
COUNTERS TO LOAD/UNLOAD THE SKIP
RAM DURING READ FUNCTIONS BY:

1. LOADING A REPEATING COUNT OF 0
TO 63 INTO THE NIBBLES OF THE
FIRST 64 WORDS OF THE IO_BUF.
2. VIA MBUS WRITE FUNCTION WRITE
THE CONTENTS OF THE IO_BUF
INTO THE GOOD BLOCK.
3. CLEAR OUT THE IO_BUF
4. VIA MBUS READ FUNCTION READ
THE GOOD BLOCK FOR THE REPEATING
COUNT OF 0 TO 63.

IMPLICIT INPUTS:

IO_BUF

A VECTOR OF 256 WORDS WHERE
DATA FOR MBUS READS AND WRITE
FUNCTION ARE FOUND.

A GLOBAL OWN LOCATION TO THIS TEST.

--

local

DODU_FLG,
WRD_CNT,
PAT_INC,
SIZ_EXP,
POS_EXP,
TEMP;

CLR MBUS;

DODU_FLG = ZERO;

!DROP UNJT FLAG
!WORD COUNT
!PATTERN INCREMENT
!SIZE EXPRESSION
!POSITIONAL EXPRESSION
!TEMPORARY STORAGE LOCATION

22-Oct-1980 10:47:44 TOPS-20 Bliss-16 V2(206)
22-Oct-1980 10:45:32 PA:<NEALE>BL2ML4.BLI.2 (76)

```

15543 :ML4
15544 :
15545 :
15546 :      7513 WRD_CNT = ZERO;
15547 :      7514
15548 :      7515 incr COUNT from 0 to 63 do      !LOAD 64 WORDS WITH REPEATING COUNTS OF 0-63
15549 :      7516
15550 :      7517      incr PAT_CNT from -1 to 11 by 4 do      !LOAD NIBBLES IN WORD WITH REPEATING COUNT OF 0-63
15551 :      7518          begin
15552 :      7519              (IO_BUF [.WRD_CNT])<0, 4> = .PAT_CNT + 1;      !LOAD FIRST NIBBLE IN WORD
15553 :      7520              (IO_BUF [.WRD_CNT])<4, 4> = .PAT_CNT + 2;      !LOAD SECOND NIBBLE IN WORD
15554 :      7521              (IO_BUF [.WRD_CNT])<8, 4> = .PAT_CNT + 3;      !LOAD THIRD NIBBLE IN WORD
15555 :      7522              (IO_BUF [.WRD_CNT])<12, 4> = .PAT_CNT + 4;      !LOAD FORTH NIBBLE IN WORD
15556 :      7523              WRD_CNT = .WRD_CNT + 1;      !INCREMENT TO NEXT WORD
15557 :      7524          end;
15558 :      7525
15559 :      7526 GD_BLK_XFER ();      !SET UP A GOOD BLOCK XFERR
15560 :      7527 MLCS1 = write;      !DO A WRITE FUNCTION
15561 :      7528
15562 :      7529 do      !DELAY UNTIL XFER TO COMPLETE
15563 :      7530          0
15564 :      7531 until .DRY IS_SET;
15565 :      7532
15566 :      7533 BGNSUB;
15567 :      7534
15568 :      7535 incr IO_CNT from 0 to 255 do      !CLEAR OUT IO_BUF
15569 :      7536          IO_BUF [.IO_CNT] = ZEROES;
15570 :      7537
15571 :      7538 CLR_MBUS;
15572 :      7539 GD_BLK_XFER ();      !SET UP A GOOD BLOCK XFERR
15573 :      7540 MLCS1 = read;      !DO A READ FUNCTION
15574 :      7541
15575 :      7542 do      !DELAY UNTIL XFER TO COMPLETE
15576 :      7543          0
15577 :      7544 until .DRY IS_SET;
15578 :      7545
15579 :      7546 CLR_MBUS;
15580 :      7547 SIZ_EXP = 4;      !FIELD SIZE FOR NIBBLES ALWAYS 4 BITS
15581 :      7548 WRD_CNT = 0;
15582 :      7549
15583 :      7550 incr COUNT from 0 to 63 do      !READ 64 WORDS IN IO_BUF
15584 :      7551
15585 :      7552      incr PAT_CNT from -1 to 11 by 4 do      !READ REPEATING COUNTS OF 0-63
15586 :      7553          begin
15587 :      7554              POS_EXP = ZERO;      !FIELD SELECTOR SELECTS THE FOUR NIBBLES
15588 :      7555              PAT_INC = ONE;
15589 :      7556              TEMP = .IO_BUF [.WRD_CNT];      !GET A WORD OUT OF IO_BUF
15590 :      7557
15591 :      7558          incr CNT from 0 to 3 do      !READ THE FOUR NIBBLES IN WORD
15592 :      7559              begin
15593 :      7560
15594 :      7561              if .TEMP<.POS_EXP, .SIZ_EXP> neq (.PAT_CNT + .PAT_INC)
15595 :      7562                  !COMPARE NIBBLE WITH RESPECTIVE 0 -63 CNT
15596 :      7563              then
15597 :      7564                  begin      !ERROR AND SET DODU_FLG IF NEQ

```


22-Oct-1980 10:47:44 TOPS-20 Bliss-16 V2(206)
22-Oct-1980 10:45:32 PA:<NEALE>BL2ML4.BLI.2 (76)

```

15599 :ML4
15600 :
15601 :
15602 : 7565 ERRDF (89, ARR_DAT, 0);
15603 : 7566 PRINTB (SIX_FMT, FNC_18, WRD_50, WRD_10, WRD_12, FNC_6, WRD_19);
15604 : 7567 DODU_FLG = ONE;
15605 : 7568 end;
15606 : 7569
15607 : 7570 POS_EXP = .POS_EXP + 4; !POINT TO THE NEXT NIBBLE IN WORD
15608 : 7571 PAT_INC = .PAT_INC + 1; !INCREMENT THE 0-63 COUNT
15609 : 7572 end;
15610 : 7573
15611 : 7574 WRD_CNT = .WRD_CNT + 1; !GET THE NEXT IO_BUF WORD
15612 : 7575 end;
15613 : 7576
15614 : 7577 ENDSUB;
15615 : 7578
15616 : 7579 if .DODU_FLG IS_SET !DROP THIS UNIT IF DODU_FLG SET
15617 : 7580 then
15618 : 7581 begin
15619 : 7582 DODU (.ML_LUN);
15620 : 7583 DOCLN;
15621 : 7584 end;
15622 : 7585
15623 : 7586 ENDTST;
15627 :

```

```

15631 055010 004167 127044 $T34: JSR R1,$SAVE5 ; 7459
15632 055014 162706 000012 SUB #12,SP ;
15633 055020 152777 000040 134712 BISB #40,@ML.REG+40 ; 7509
15634 055026 016705 135136 MOV ML.DUT,R5
15635 055032 042705 177770 BIC #177770,R5
15636 055036 142777 000007 134674 BICB #7,@ML.REG+40
15637 055044 150577 134670 BISB R5,@ML.REG+40
15638 055050 005066 000010 CLR 10(SP) ; DODU.FLG 7512
15639 055054 005066 000002 CLR 2(SP) ; WRD.CNT 7513
15640 055060 005002 CLR R2 ; COUNT 7515
15641 055062 012703 177777 1$: MOV #-1,R3 ; *,PAT.CNT 7517
15642 055066 016604 000002 2$: MOV 2(SP),R4 ; WRD.CNT,* 7519
15643 055072 006304 ASL R4
15644 055074 062704 010342 ADD #10.BUF,R4
15645 055100 010305 MOV R3,R5 ; PAT.CNT,*
15646 055102 005205 INC R5
15647 055104 042705 177760 BIC #177760,R5
15648 055110 142714 000017 BICB #17,(R4)
15649 055114 150514 BISB R5,(R4)
15650 055116 010305 MOV R3,R5 ; PAT.CNT,* 7520
15651 055120 062705 000002 ADD #2,R5
15652 055124 006305 ASL R5

```

Line No.	Address	Op Code	Op Data	Op Comment	Label	Count
15654					:ML4	
15655					:	
15656						
15657	055126	006305		ASL R5		
15658	055130	006305		ASL R5		
15659	055132	006305		ASL R5		
15660	055134	042705	177417	BIC #177417,R5		
15661	055140	142714	000360	BICB #360,(R4)		
15662	055144	150514		BISB R5,(R4)		
15663	055146	010305		MOV R3,R5	: PAT.CNT,*	7521
15664	055150	062705	000003	ADD #3,R5		
15665	055154	000305		SWAB R5		
15666	055156	042705	170377	BIC #170377,R5		
15667	055162	042714	007400	BIC #7400,(R4)		
15668	055166	050514		BIS R5,(R4)		
15669	055170	010305		MOV R3,R5	: PAT.CNT,*	7522
15670	055172	062705	000004	ADD #4,R5		
15671	055176	000305		SWAB R5		
15672	055200	006305		ASL R5		
15673	055202	006305		ASL R5		
15674	055204	006305		ASL R5		
15675	055206	006305		ASL R5		
15676	055210	042705	007777	BIC #7777,R5		
15677	055214	042714	170000	BIC #170000,(R4)		
15678	055220	050514		BIS R5,(R4)		
15679	055222	005266	000002	INC 2(SP)	: WRD.CNT	7523
15680	055226	062703	000004	ADD #4,R3	: *,PAT.CNT	7517
15681	055232	020327	000013	CMP R3,#13	: PAT.CNT,*	
15682	055236	003713		BLE 2\$		
15683	055240	005202		INC R2	: COUNT	7515
15684	055242	020227	000077	CMP R2,#77	: COUNT,*	
15685	055246	003705		BLE 1\$		
15686	055250	004767	135266	JSR PC,GD.BLK.XFER		7526
15687	055254	012777	000061	MOV #61,@ML.REG		7527
15688	055262	105777	134462	TSTB @ML.REG+50		7531
15689	055266	100375		BPL 3\$		
15690	055270	104402		TRAP 2		
15691	055272	005003		CLR R3	: IO.CNT	7535
15692	055274	010304		MOV R3,R4	: IO.CNT,*	7536
15693	055276	006304		ASL R4		
15694	055300	005064	010342	CLR IO.BUF(R4)		
15695	055304	005203		INC R3	: IO.CNT	7535
15696	055306	020327	000377	CMP R3,#377	: IO.CNT,*	
15697	055312	003770		BLE 5\$		
15698	055314	152777	000040	BISB #40,@ML.REG+40		7536
15699	055322	016704	134642	MOV ML.DUT,R4		
15700	055326	042704	177770	BIC #177770,R4		
15701	055332	142777	000007	BICB #7,@ML.REG+40		
15702	055340	150477	134374	BISB R4,@ML.REG+40		
15703	055344	004767	135172	JSR PC,GD.BLK.XFER		7539
15704	055350	012777	000071	MOV #71,@ML.REG		7540
15705	055356	105777	134366	TSTB @ML.REG+50		7544
15706	055362	100375		BPL 6\$		
15707	055364	152777	000040	BISB #40,@ML.REG+40		
15708	055372	016704	134572	MOV ML.DUT,R4		

22-Oct-1980 10:47:44 TOPS
 22-Oct-1980 10:45:32 PA:<

Address	OpCode	Op1	Op2	Op3	Op4	Comments	Line#
15710						:ML4	
15711						:	
15712							
15713	055376	042704	177770			BIC #177770,R4	
15714	055402	142777	000007	134330		BICB #7,@ML.REG+40	
15715	055410	150477	134324			BISB R4,@ML.REG+40	
15716	055414	012766	000004	000006		MOV #4,6(SP)	: *,SIZ.EXP 7547
15717	055422	005066	000002			CLR 2(SP)	: WRD.CNT 7548
15718	055426	005001				CLR R1	: COUNT 7550
15719	055430	012703	177777		7\$:	MOV #-1,R3	: *,PAT.CNT 7552
15720	055434	005005			8\$:	CLR R5	: POS.EXP 7554
15721	055436	012716	000001			MOV #1,(SP)	: *,PAT.INC 7555
15722	055442	016604	000002			MOV 2(SP),R4	: WRD.CNT,* 7556
15723	055446	006304				ASL R4	
15724	055450	016466	010342	000004		MOV IO.BUF(R4),4(SP)	: *,TEMP
15725	055456	005002				CLR R2	: CNT 7558
15726	055460	016646	000004		9\$:	MOV 4(SP),-(SP)	: TEMP,* 7561
15727	055464	010546				MOV R5,-(SP)	: POS.EXP,*
15728	055466	016646	000012			MOV 12(SP),-(SP)	: SIZ.EXP,*
15729	055472	005046				CLR -(SP)	
15730	055474	004767	125260			JSR PC,BL\$GT1	
15731	055500	062706	000010			ADD #10,SP	
15732	055504	010304				MOV R3,R4	: PAT.CNT,*
15733	055506	061604				ADD (SP),R4	: PAT.INC,*
15734	055510	020004				CMP R0,R4	
15735	055512	001433				BEQ 10\$	
15736	055514	104455				TRAP 55	:
15737	055516	000131				.WORD 131	: 7565
15738	055520	007534				.WORD ARR.DAT	
15739	055522	000000				.WORD 0	
15740	055524	012746	006040			MOV #WRD.19, -(SP)	:
15741	055530	012746	007030			MOV #FNC.6, -(SP)	: 7566
15742	055534	012746	005760			MOV #WRD.12, -(SP)	
15743	055540	012746	005740			MOV #WRD.10, -(SP)	
15744	055544	012746	006400			MOV #WRD.50, -(SP)	
15745	055550	012746	007216			MOV #FNC.18, -(SP)	
15746	055554	012746	005432			MOV #SIX.FMT, -(SP)	
15747	055560	012746	000007			MOV #7, -(SP)	
15748	055564	010600				MOV SP,R0	: SP,*
15749	055566	104414				TRAP 14	
15750	055570	012766	000001	000030		MOV #1,30(SP)	: *,DODU.FLG 7567
15751	055576	062706	000020			ADD #20,SP	: 7564
15752	055602	062705	000004		10\$:	ADD #4,R5	: *,POS.EXP 7570
15753	055606	005216				INC (SP)	: PAT.INC 7571
15754	055610	005202				INC R2	: CNT 7558
15755	055612	020227	000003			CMP R2,#3	: CNT,*
15756	055616	003720				BLE 9\$	
15757	055620	005266	000002			INC 2(SP)	: WRD.CNT 7574
15758	055624	062703	000004			ADD #4,R3	: *,PAT.CNT 7552
15759	055630	020327	000013			CMP R3,#13	: PAT.CNT,*
15760	055634	003677				BLE 8\$	
15761	055636	005201				INC R1	: COUNT 7550
15762	055640	020127	000077			CMP R1,#77	: COUNT,*
15763	055644	003671				BLE 7\$	
15764	055646	104467				TRAP 67	: 7575

```
15766          :ML4
15767          :
15768
15769 055650 006000          ROR    R0
15770 055652 103606          BLO   4$
15771 055654 026627 000010 000001  CMP   10(SP),#1      ; DODU.FLG,*      7579
15772 055662 001004          BNE   11$
15773 055664 016700 134276  MOV   ML.LUN,R0      ;                      7582
15774 055670 104451          TRAP  51
15775 055672 104444          TRAP  44
15776 055674 062706 000012 11$:  ADD   #12,SP      ;                      7459
15777 055700 000207          RTS   PC
15778
15779          ; Routine Size: 221 words
15780          ; Maximum stack depth per invocation: 19 words
15785
15786
15790
15794 055702          T34::
15795 055702 004767 177102 1$:  JSR   PC,$T34      ;                      7584
15796 055706 104466          TRAP  66
15797 055710 006000          ROR   R0
15798 055712 103773          BLO   1$
15799 055714 000207          RTS   PC
15800
15801          ; Routine Size: 6 words
15802          ; Maximum stack depth per invocation: 0 words
15807
15808
15809 ;          7587 !<BLF/PAGE>
```

22-Oct-1980 10:47:44
22-Oct-1980 10:45:32

TOPS-20 Bliss-16 V2(206)
PA:<NEALE>BL2ML4.BLI.2 (77)

15811 :ML4
15812 :
15813 :
15814 : 7588
15815 : 7589
15816 : 7590
15817 : 7591
15818 : 7592
15819 : 7593
15820 : 7594
15821 : 7595
15822 : 7596
15823 : 7597
15824 : 7598
15825 : 7599
15826 : 7600
15827 : 7601
15828 : 7602
15829 : 7603
15830 : 7604
15831 : 7605
15832 : 7606
15833 : 7607
15834 : 7608
15835 : 7609
15836 : 7610
15837 : 7611
15838 : 7612
15839 : 7613
15840 : 7614
15841 : 7615
15842 : 7616
15843 : 7617
15844 : 7618
15845 : 7619
15846 : 7620
15847 : 7621
15848 : 7622
15849 : 7623
15850 : 7624
15851 : 7625
15852 : 7626
15853 : 7627
15854 : 7628
15855 : 7629
15856 : 7630
15857 : 7631
15858 : 7632
15859 : 7633
15860 : 7634
15861 : 7635
15862 : 7636
15863 : 7637
15864 : 7638
15865 : 7639

BGNTST;

!++

TEST NUMBER: TST 35

TEST NAME: SYNC DATA BUS BIT UNIQUENESS TEST/WRITE PATH

TEST DESCRIPTION:

TEST SYNCHRONOUS DATA BUS FOR
DATA BIT UNIQUENESS BY:

1. LOADING THE FIRST 16 WORDS IN
THE IO BUF WITH A SHIFTING
ZERO IN A FIELD OF ONES PATTERN.
2. VIA MBUS WRITE FUNCTION WRITE
SHIFTING PATTERN THROUGH THE
DATA BUS AND INTO THE GOOD
BLOCK.
3. VIA DAT DM MODE READ THE
GOOD BLOCK AND SAVE ALL GOOD
NIBBLE DATA, IN THEIR PROPER
SEQUENCE, INTO A STACK
STRUCTURE.
4. INTERRIGATE STACK STRUCTURE FOR
SHIFTED DATA PATTERN.

IMPLICIT INPUTS:

PD TEMP
A BIT VECTOR OF 16 BITS WHERE
THE READ PROM DATA IS STORED
AND ACCESSED FROM.

GLOBAL OWN LOCATION TO THIS
TST.

IO BUF
A VECTOR OF 256 WORDS WHERE
DATA FOR MBUS READS AND WRITE
FUNCTION ARE FOUND.

A GLOBAL OWN LOCATION TO
THIS TEST.

STACK

22-Oct-1980 10:47:44 TOPS-20 Bliss-16 V2(206)
22-Oct-1980 10:45:32 PA:<NEALE>BL2ML4.BLI.2 (77)

```

15867 :ML4
15868 :
15869 :
15870 : 7640 ! A VECTOR OF 198 BYTE LOCATIONS
15871 : 7641 ! WHERE GOOD NIBBLE DATA IS STORED
15872 : 7642 ! WHEN STRIPPING AWAY BAD NIBBLE
15873 : 7643 ! LOCATIONS OF A BLOCK.
15874 : 7644 !
15875 : 7645 !
15876 : 7646 !--
15877 : 7647
15878 : 7648 local
15879 : 7649 SAV_NIB, !STORES THE SHIFTED BIT PATTERN
15880 : 7650 DODU_FLG, !DROP UNIT FLAG
15881 : 7651 NIB_BIT, !NIBBLE PATTERN
15882 : 7652 TST_PAT, !TEST PATTERN
15883 : 7653 ALL_ONES_1, !HOW MANY '17' NIBBLE PATTERN POSITION TO READ
15884 : 7654 ALL_ONES_2, !HOW MANY '17' NIBBLE PATTERN POSITION TO READ
15885 : 7655 STK_PTR, !STACK POINTER
15886 : 7656 COUNT; !COUNTER
15887 : 7657
15888 : 7658 BGNSUB;
15889 : 7659 CLR_MBUS;
15890 : 7660 DODU_FLG = ZERO;
15891 : 7661 TST_PAT = ONE; !ONE IN A FIELD OF ZEROES
15892 : 7662
15893 : 7663 incr CNT from 0 to 15 do !WRITE 16 WORDS WITH SHIFTING 0 IN FIELD OF 1'S.
15894 : 7664 begin
15895 : 7665 IO_BUF [.CNT] = not .TST_PAT;
15896 : 7666 TST_PAT = .TST_PAT^ONE;
15897 : 7667 end;
15898 : 7668
15899 : 7669 GD_BLK_XFER (); !SET UP A GOOD BLOCK XFERR
15900 : 7670 MLC51 = write; !WRITE SHIFTING PATTERN THROUGH SYNC BUS
15901 : 7671
15902 : 7672 do !DELAY UNTIL XFER TO COMPLETE
15903 : 7673 0
15904 : 7674 until .DRY IS_SET;
15905 : 7675
15906 : 7676 incr CNT from 0 to 8 do !ZERO OUT THE NIBBLE OFFSET COUNTERS
15907 : 7677 STK_OFF [.CNT] = ZEROES;
15908 : 7678
15909 : 7679 CLR_MBUS;
15910 : 7680 STK_PTR = -1; !RESET THE STACK POINTER
15911 : 7681 DAT_DM_XFER (); !SET UP A DATA DIAG XFERR AT THE GOOD BLOCK
15912 : 7682 MLC51 = read; !DO A READ FUNCTION
15913 : 7683 DELAY (ONE_US);
15914 : 7684
15915 : 7685 incr CNT from 0 to 21 do !LOAD THE STACK WITH ALL GOOD NIBBLE DATA
15916 : 7686 begin
15917 : 7687 PD_TEMP = .MLPD; !GET THE FROM DATA
15918 : 7688 DAT_CLK = ONE; !CLOCK OUT THE DATA WORD
15919 : 7689 DELAY (ONE_US);
15920 : 7690 RD_LNG_WRD; !READ THE DATA DIAG REGISTERS
15921 : 7691

```

```

15923 :ML4
15924 :
15925 :
15926 : 7692      incr NIB_PTR from 0 to 8 do
15927 : 7693      begin
15928 : 7694      STK_PTR = .STK_PTR + 1;
15929 : 7695
15930 : 7696      if .PD_TEMP [.NIB_PTR] IS_SET
15931 : 7697      then
15932 : 7698      STK_OFF [.NIB_PTR] = .STK_OFF [.NIB_PTR] + 9
15933 : 7699      else
15934 : 7700      LOAD_STACK (.STK_PTR, .NIB_PTR);
15935 : 7701
15936 : 7702      end;
15937 : 7703
15938 : 7704      end;
15939 : 7705
15940 : 7706      STK_PTR = -1;
15941 : 7707      NIB_BIT = ONE;
15942 : 7708      ALL_ONES_1 = ZERO;
15943 : 7709      ALL_ONES_2 = 3;
15944 : 7710
15945 : 7711      incr BY_FOUR_WRDS from 0 to 3 do
15946 : 7712      begin
15947 : 7713
15948 : 7714      incr BY_ONE_WRD from 0 to 3 do
15949 : 7715      begin
15950 : 7716      COUNT = ZERO;
15951 : 7717
15952 : 7718      until .COUNT eql .ALL_ONES_1 do
15953 : 7719      begin
15954 : 7720      COUNT = .COUNT + 1;
15955 : 7721      STK_PTR = .STK_PTR + 1;
15956 : 7722
15957 : 7723      if (.stack [.STK_PTR]) neq %'000017'
15958 : 7724      then
15959 : 7725      begin
15960 : 7726      ERRDF (90, SYNC, 0);
15961 : 7727      PRINTB (SIX_FMT, WRD_23, WRD_39, PHR_4, WRD_12, FNC_5, WRD_19);
15962 : 7728      PRINTB (FMT_5, ONES, .stack [.STK_PTR], .STR_PTR);
15963 : 7729      DODU_FLG = ONE;
15964 : 7730      end;
15965 : 7731
15966 : 7732      end;
15967 : 7733
15968 : 7734      STK_PTR = .STK_PTR + 1;
15969 : 7735      SAV_NIB = ( not .NIB_BIT) and (%'000017');
15970 : 7736
15971 : 7737      if (.stack [.STK_PTR]) neq (.SAV_NIB)
15972 : 7738      then
15973 : 7739      begin
15974 : 7740      ERRDF (91, SYNC, 0);
15975 : 7741      PRINTB (SIX_FMT, WRD_23, WRD_39, PHR_4, WRD_12, FNC_5, WRD_19);
15976 : 7742      PRINTB (FMT_15, .STK_PTR);
15977 : 7743      PRINTB (FMT_5, .SAV_NIB, .stack [.STK_PTR]);

```

22-Oct-1980 10:47:44 TOPS-20 Bliss-16 V2(206)
 22-Oct-1980 10:45:32 PA:<NEALE>BL2ML4.BLI.2 (77)

```

!LOOK AT 9 NIBBLES
!INCREMENT THE STACK POINTER
!SEE IF THIS IS A GOOD NIBBLE
!INCREMENT NIBBLE OFF SET IF BAD
!ELSE LOAD THE STACK WITH GOOD NIBBLE DATA
!RESET THE STACK POINTER
!SHIFTING NIBBLE PAT OF 1 IN FIELD OF 0'S
!READ NO '17' NIBBLE PATTERN ON FIRST PASS
!READ THREE '17' NIBBLE PATTERN ON FIRST PASS
!READ 4 GROUPS OF 4 WORDS
!READ 4 GROUPS OF 1 WORD
!CLEAR COUNT
!READ X NUMBER OF '17' NIBBLE PAT
!INCREMENT COUNT
!INCREMENT STACK POINTER
!COMPARE STACK WITH '17'
!ERROR AND SET DODU_FLG IF NEQ
!INCREMENT THE STACK POINTER
!GENERATE THE SHIFTED BIT
!COMPARE STACK TO SHIFTED BIT
!ERROR AND SET DODU_FLG IF NEQ

```

22-Oct-1980 10:47:44 TOPS-20 Bliss-16 V2(206)
22-Oct-1980 10:45:32 PA:<NEALE>BL2ML4.BLI.2 (77)

```

15979 :ML4
15980 :
15981 :
15982 :       7744           DODU_FLG = ONE;
15983 :       7745           end;
15984 :
15985 :       7746           COUNT = ZEROES;
15986 :       7747           !CLEAR COUNT
15987 :       7748           until .COUNT eql .ALL_ONES_2 do
15988 :       7749           !READ X NUMBER OF '17' NIBBLE PAT
15989 :       7750           begin
15990 :       7751           COUNT = .COUNT + 1;
15991 :       7752           !INCREMENT COUNT
15992 :       7753           STK_PTR = .STK_PTR + 1;
15993 :       7754           !INCREMENT STACK POINTER
15994 :       7755           if (.stack [.STK_PTR]) neq %o'000017'
15995 :       7756           !COMPARE STACK POINTER WITH '17'
15996 :       7757           then
15997 :       7758           begin
15998 :       7759           !ERROR AND SET DODU_FLG IF SET
15999 :       7760           ERRDF (92, SYNC, 0);
16000 :       7761           PRINTB (SIX_FMT, WRD_23, WRD_39, PHR_4, WRD_12, FNC_5, WRD_19);
16001 :       7762           PRINTB (FMT_5, ONES, .stack [.STK_PTR], .STK_PTR);
16002 :       7763           DODU_FLG = ONE;
16003 :       7764           end;
16004 :       7765           end;
16005 :       7766           NIB_BIT = .NIB_BIT^ONE;
16006 :       7767           !SHIFT THE SHIFTED NIBBLE BIT
16007 :       7768           end;
16008 :       7769           NIB_BIT = ONE;
16009 :       7770           !RESET THE SHIFTED NIBBLE BIT
16010 :       7771           ALL_ONES_1 = .ALL_ONES_1 + 1;
16011 :       7772           !READ ONE MORE '17' PATTERN
16012 :       7773           ALL_ONES_2 = .ALL_ONES_2 - 1;
16013 :       7774           !READ ONE LESS '17' PATTERN
16014 :       7775           end;
16015 :       7776           ENDSUB;
16016 :       7777           if .DODU_FLG IS_SET
16017 :       7778           !DROP THIS UNIT IF DODU_FLG SET
16018 :       7779           then
16019 :       7780           begin
16020 :       7781           DODU (.ML_LUN);
16021 :       7782           DOCLN;
16022 :       7783           end;
16023 :       7784           ENDTST;
16024 :
16028 055716 004167 126136      $T35: JSR   R1,$SAVE5      ;           7586
16029 055722 162706 000016      SUB   #16,SP
16030 055726 104402      1$: TRAP 2           ;           7656
16031 055730 152777 000040 134002  BISB  #40,@ML.REG+40 ;           7658
16032 055736 016705 134226      MOV   ML.DUT,R5

```


Address	Instruction	Label	Comments	Time	Page
16034		:ML4		22-Oct-1980 10:47:44	TOPS
16035		:		22-Oct-1980 10:45:32	PA:<
16036					
16037	055742 042705 177770		BIC #177770,R5		
16038	055746 142777 000007 133764		BICB #7,@ML.REG+40		
16039	055754 150577 133760		BISB R5,@ML.REG+40		
16040	055760 005016		CLR (SP)	: DODU.FLG	7660
16041	055762 012766 000001 000010		MOV #1,10(SP)	: *,TST.PAT	7661
16042	055770 005004		CLR R4	: CNT	7663
16043	055772 010405	2\$:	MOV R4,R5	: CNT,*	7665
16044	055774 006305		ASL R5		
16045	055776 016665 000010 010342		MOV 10(SP),IO.BUF(R5)	: TST.PAT,*	
16046	056004 005165 010342		COM IO.BUF(R5)		
16047	056010 006366 000010		ASL 10(SP)	: TST.PAT	7666
16048	056014 005204		INC R4	: CNT	7663
16049	056016 020427 000017		CMP R4,#17	: CNT,*	
16050	056022 003763		BLE 2\$		
16051	056024 004767 134512		JSR PC,GD.BLK.XFER		7669
16052	056030 012777 000061 133642		MOV #61,@ML.REG		7670
16053	056036 105777 133706	3\$:	TSTB @ML.REG+50		7674
16054	056042 100375		BPL 3\$		
16055	056044 005005		CLR R5	: CNT	7676
16056	056046 105065 011342	4\$:	CLRB STK.OFF(R5)	: *(CNT)	7677
16057	056052 005205		INC R5	: CNT	7676
16058	056054 020527 000010		CMP R5,#10	: CNT,*	
16059	056060 003772		BLE 4\$		
16060	056062 152777 000040 133650		BISB #40,@ML.REG+40		7677
16061	056070 016705 134074		MOV ML.DUT,R5		
16062	056074 042705 177770		BIC #177770,R5		
16063	056100 142777 000007 133632		BICB #7,@ML.REG+40		
16064	056106 150577 133626		BISB R5,@ML.REG+40		
16065	056112 012702 177777		MOV #-1,R2	: *,STK.PTR	7680
16066	056116 004767 134504		JSR PC,DAT.DM.XFER		7681
16067	056122 012777 000071 133550		MOV #71,@ML.REG		7682
16068	056130 012704 000001		MOV #1,R4	: *,\$\$TMP2	7683
16069	056134 001411	5\$:	BEQ 8\$		
16070	056136 016705 123754		MOV LSDLY,R5	: *,\$\$TMP1	
16071	056142 001404		BEQ 7\$		
16072	056144 005066 000014	6\$:	CLR 14(SP)	: \$\$TMP	
16073	056150 005305		DEC R5	: \$\$TMP1	
16074	056152 001374		BNE 6\$		
16075	056154 005304	7\$:	DEC R4	: \$\$TMP2	
16076	056156 000766		BR 5\$		
16077	056160 005003	8\$:	CLR R3	: CNT	7685
16078	056162 017767 133742 133472	9\$:	MOV @ML.REG+230,PD.TEMP		7687
16079	056170 152777 000020 133622		BISB #20,@ML.REG+120		7688
16080	056176 012704 000001		MOV #1,R4	: *,\$\$TMP2	7689
16081	056202 001411	10\$:	BEQ 13\$		
16082	056204 016705 123706		MOV LSDLY,R5	: *,\$\$TMP1	
16083	056210 001404		BEQ 12\$		
16084	056212 005066 000014	11\$:	CLR 14(SP)	: \$\$TMP	
16085	056216 005305		DEC R5	: \$\$TMP1	
16086	056220 001374		BNE 11\$		
16087	056222 005304	12\$:	DEC R4	: \$\$TMP2	
16088	056224 000766		BR 10\$		

Address	Op1	Op2	Op3	Op4	Op5	Instruction	Comments	Seq
16090					:ML4			
16091					:			
16092								
16093	056226	017767	133636	131460	13\$:	MOV @ML.REG+170,D1.TEMP		
16094	056234	017767	133640	131454		MOV @ML.REG+200,D2.TEMP		
16095	056242	017767	133612	131450		MOV @ML.REG+160,E2.TEMP		
16096	056250	005005				CLR R5	: NIB.PTR	7692
16097	056252	005202			14\$:	INC R2	: STK.PTR	7694
16098	056254	010504				MOV R5,R4	: NIB.PTR,*	7696
16099	056256	006204				ASR R4		
16100	056260	006204				ASR R4		
16101	056262	006204				ASR R4		
16102	056264	062704	011662			ADD #PD.TEMP,R4		
16103	056270	010446				MOV R4,-(SP)		
16104	056272	010546				MOV R5,-(SP)	: NIB.PTR,*	
16105	056274	042716	177770			BIC #177770,(SP)		
16106	056300	012746	000001			MOV #1,-(SP)		
16107	056304	005046				CLR -(SP)		
16108	056306	004767	124570			JSR PC,BL\$GT2		
16109	056312	062706	000010			ADD #10,SP		
16110	056316	005300				DEC R0		
16111	056320	001010				BNE 15\$		
16112	056322	005004				CLR R4	:	7698
16113	056324	156504	011342			BISB STK.OFF(R5),R4	: *(NIB.PTR),*	
16114	056330	062704	000011			ADD #11,R4		
16115	056334	110465	011342			MOVB R4,STK.OFF(R5)	: *,*(NIB.PTR)	
16116	056340	000405				BR 16\$:	7696
16117	056342	010246			15\$:	MOV R2,-(SP)	: STK.PTR,*	7700
16118	056344	010546				MOV R5,-(SP)	: NIB.PTR,*	
16119	056346	004767	133620			JSR PC,LOAD.STACK		
16120	056352	022626				CMP (SP)+,(SP)+		
16121	056354	005205			16\$:	INC R5	: NIB.PTR	7692
16122	056356	020527	000010			CMP R5,#10	: NIB.PTR,*	
16123	056362	003733				BLE 14\$		
16124	056364	005203				INC R3	: CNT	7685
16125	056366	020327	000025			CMP R3,#25	: CNT,*	
16126	056372	003673				BLE 9\$		
16127	056374	012702	177777			MOV #-1,R2	: *,STK.PTR	7706
16128	056400	012766	000001	000002		MOV #1,2(SP)	: *,NIB.BIT	7707
16129	056406	005066	000006			CLR 6(SP)	: ALL.ONES.1	7708
16130	056412	012766	000003	000004		MOV #3,4(SP)	: *,ALL.ONES.2	7709
16131	056420	005004				CLR R4	: BY.FOUR.WRDS	7711
16132	056422	005005			17\$:	CLR R5	: BY.ONE.WRD	7714
16133	056424	005001			18\$:	CLR R1	: COUNT	7716
16134	056426	020166	000006		19\$:	CMP R1,6(SP)	: COUNT,ALL.ONES.1	7718
16135	056432	001456				BEQ 20\$		
16136	056434	005201				INC R1	: COUNT	7720
16137	056436	005202				INC R2	: STK.PTR	7721
16138	056440	126227	011354	000017		CMPB STACK(R2),#17	: *(STK.PTR),*	7723
16139	056446	001767				BEQ 19\$		
16140	056450	104455				TRAP 55	:	7726
16141	056452	000132				.WORD 132		
16142	056454	007500				.WORD SYNC		
16143	056456	000000				.WORD 0		
16144	056460	012746	006040			MOV #WRD.19,-(SP)	:	7727

```
16146      ;ML4
16147      ;
16148
16149 056464 012746 007020      MOV      #FNC.5,-(SP)
16150 056470 012746 005760      MOV      #WRD.12,-(SP)
16151 056474 012746 006630      MOV      #PHR.4,-(SP)
16152 056500 012746 006250      MOV      #WRD.39,-(SP)
16153 056504 012746 006076      MOV      #WRD.23,-(SP)
16154 056510 012746 005432      MOV      #SIX.FMT,-(SP)
16155 056514 012746 000007      MOV      #7,-(SP)
16156 056520 010600      MOV      SP,R0      ; SP,*
16157 056522 104414      TRAP     14
16158 056524 010216      MOV      R2,(SP)      ; STK.PTR,*      7728
16159 056526 005046      CLR      -(SP)
16160 056530 116216 011354      MOV      STACK(R2),(SP) ; *(STK.PTR),*
16161 056534 012746 177777      MOV      #-1,-(SP)
16162 056540 012746 004366      MOV      #FMT.5,-(SP)
16163 056544 012746 000004      MOV      #4,-(SP)
16164 056550 010600      MOV      SP,R0      ; SP,*
16165 056552 104414      TRAP     14
16166 056554 012766 000001 000030      MOV      #1,30(SP)      ; *,DODU.FLG      7729
16167 056562 062706 000030      ADD      #30,SP      ;      7725
16168 056566 000717      BR       19$      ;      7718
16169 056570 005202      INC      R2      ; STK.PTR      7734
16170 056572 012766 000017 000012 20$:      MOV      #17,12(SP)      ; *,SAV.NIB      7735
16171 056600 046666 000002 000012      BIC      2(SP),12(SP)      ; NIB.BIT,SAV.NIB
16172 056606 005003      CLR      R3      ;      7737
16173 056610 156203 011354      BISB     STACK(R2),R3      ; *(STK.PTR),*
16174 056614 020366 000012      CMP      R3,12(SP)      ; *,SAV.NIB
16175 056620 001455      BEQ     21$
16176 056622 104455      TRAP     55      ;      7740
16177 056624 000133      .WORD   133
16178 056626 007500      .WORD   SYNC
16179 056630 000000      .WORD   0
16180 056632 012746 006040      MOV      #WRD.19,-(SP)      ;      7741
16181 056636 012746 007020      MOV      #FNC.5,-(SP)
16182 056642 012746 005760      MOV      #WRD.12,-(SP)
16183 056646 012746 006630      MOV      #PHR.4,-(SP)
16184 056652 012746 006250      MOV      #WRD.39,-(SP)
16185 056656 012746 006076      MOV      #WRD.23,-(SP)
16186 056662 012746 005432      MOV      #SIX.FMT,-(SP)
16187 056666 012746 000007      MOV      #7,-(SP)
16188 056672 010600      MOV      SP,R0      ; SP,*
16189 056674 104414      TRAP     14
16190 056676 010216      MOV      R2,(SP)      ; STK.PTR,*      7742
16191 056700 012746 005106      MOV      #FMT.15,-(SP)
16192 056704 012746 000002      MOV      #2,-(SP)
16193 056710 010600      MOV      SP,R0      ; SP,*
16194 056712 104414      TRAP     14
16195 056714 005016      CLR      (SP)      ;      7743
16196 056716 116216 011354      MOV      STACK(R2),(SP) ; *(STK.PTR),*
16197 056722 016646 000036      MOV      36(SP),-(SP)      ; SAV.NIB,*
16198 056726 012746 004366      MOV      #FMT.5,-(SP)
16199 056732 012746 000003      MOV      #3,-(SP)
16200 056736 010600      MOV      SP,R0      ; SP,*
```

Address	OpCode	Operand 1	Operand 2	Operand 3	Operand 4	Instruction	Comments	Line No.
16202						TRAP 14		
16203						MOV #1,32(SP)	; *,DODU.FLG	7744
16204						ADD #32,SP		7739
16205	056740	104414	000001	000032		CLR R1	; COUNT	7747
16206	056742	012766	000032			CMP R1,4(SP)	; COUNT,ALL.ONES.2	7749
16207	056750	062706				BEQ 23\$		
16208	056754	005001				INC R1	; COUNT	7751
16209	056756	020166	000004			INC R2	; STK.PTR	7752
16210	056762	001456				CMPB STACK(R2),#17	; *(STK.PTR),*	7754
16211	056764	005201				BEQ 22\$		
16212	056766	005202				TRAP 55		7757
16213	056770	126227	011354	000017		.WORD 134		
16214	056776	001767				.WORD SYNC		
16215	057000	104455				.WORD 0		
16216	057002	000134				MOV #WRD.19,-(SP)		7758
16217	057004	007500				MOV #FNC.5,-(SP)		
16218	057006	000000				MOV #WRD.12,-(SP)		
16219	057010	012746	006040			MOV #PHR.4,-(SP)		
16220	057014	012746	007020			MOV #WRD.39,-(SP)		
16221	057020	012746	005760			MOV #WRD.23,-(SP)		
16222	057024	012746	006630			MOV #SIX.FMT,-(SP)		
16223	057024	012746	006630			MOV #7,-(SP)		
16224	057030	012746	006250			MOV SP,RO	; SP,*	
16225	057034	012746	006076			TRAP 14		
16226	057040	012746	005432			MOV R2,(SP)	; STK.PTR,*	7759
16227	057044	012746	000007			CLR -(SP)		
16228	057050	010600				MOVB STACK(R2),(SP)	; *(STK.PTR),*	
16229	057052	104414				MOV #-1,-(SP)		
16230	057054	010216				MOV #FMT.5,-(SP)		
16231	057056	005046				MOV #4,-(SP)		
16232	057060	116216	011354			MOV SP,RO	; SP,*	
16233	057064	012746	177777			TRAP 14		
16234	057070	012746	004366			MOV #1,30(SP)	; *,DODU.FLG	7760
16235	057074	012746	000004			ADD #30,SP		7756
16236	057100	010600				BR 22\$		7749
16237	057102	104414				ASL 2(SP)	; NIB.BIT	7765
16238	057104	012766	000001	000030		INC R5	; BY.ONE.WRD	7714
16239	057112	062706	000030			CMP R5,#3	; BY.ONE.WRD,*	
16240	057116	000717				BGT 24\$		
16241	057120	006366	000002			JMP 18\$		
16242	057124	005205				MOV #1,2(SP)	; *,NIB.BIT	7768
16243	057126	020527	000003			INC 6(SP)	; ALL.ONES.1	7769
16244	057132	003002				DEC 4(SP)	; ALL.ONES.2	7770
16245	057134	000167	177264			INC R4	; BY.FOUR.WRDS	7711
16246	057140	012766	000001	000002		CMP R4,#3	; BY.FOUR.WRDS,*	
16247	057146	005266	000006			BGT 25\$		
16248	057152	005366	000004			JMP 17\$		
16249	057156	005204				TRAP 67		7771
16250	057160	020427	000003			ROR R0		
16251	057164	003002				BHIS 26\$		
16252	057166	000167	177230			JMP 1\$		
16253	057172	104467				CMP (SP),#1	; DODU.FLG,*	7775
16254	057174	006000						
16255	057176	103002						
16256	057200	000167	176522					
16257	057204	021627	000001					

22-Oct-1980 10:47:44 TOPS
22-Oct-1980 10:45:32 PA:<

```
16258 ;ML4 22-Oct-1980 10:47:44 TOPS
16259 ; 22-Oct-1980 10:45:32 PA:<
16260
16261 057210 001004 BNE 27$
16262 057212 016700 132750 MOV ML.LUN,RO ; 7778
16263 057216 104451 TRAP 51
16264 057220 104444 TRAP 44
16265 057222 062706 000016 27$: ADD #16,SP ; 7586
16266 057226 000207 RTS PC
16267
16268 ; Routine Size: 357 words
16269 ; Maximum stack depth per invocation: 26 words
16274
16275
16279
16283 057230 T35::
16284 057230 004767 176462 1$: JSR PC,$T35 ; 7780
16285 057234 104466 TRAP 66
16286 057236 006000 ROR RO
16287 057240 103773 BLO 1$
16288 057242 000207 RTS PC
16289
16290 ; Routine Size: 6 words
16291 ; Maximum stack depth per invocation: 0 words
16296
16297
16298 ; 7783 !<BLF/PAGE>
```

22-Oct-1980 10:47:44 TOPS-20 Bliss-16 V2(206)
 22-Oct-1980 10:45:32 PA:<NEALE>BL2ML4.BLI.2 (78)

16300 :ML4
 16301 :
 16302 :
 16303 :
 16304 :
 16305 :
 16306 :
 16307 :
 16308 :
 16309 :
 16310 :
 16311 :
 16312 :
 16313 :
 16314 :
 16315 :
 16316 :
 16317 :
 16318 :
 16319 :
 16320 :
 16321 :
 16322 :
 16323 :
 16324 :
 16325 :
 16326 :
 16327 :
 16328 :
 16329 :
 16330 :
 16331 :
 16332 :
 16333 :
 16334 :
 16335 :
 16336 :
 16337 :
 16338 :
 16339 :
 16340 :
 16341 :
 16342 :
 16343 :
 16344 :
 16345 :
 16346 :
 16347 :
 16348 :
 16349 :
 16350 :
 16351 :
 16352 :
 16353 :
 16354 :

```

7784 :
7785 : BGNTST;
7786 :
7787 : ++
7788 : TEST NUMBER: TST 36
7789 :
7790 : TEST NAME: SYNC DATA BUS BIT UNIQUENESS TEST/READ PATH
7791 :
7792 : TEST DESCRIPTION:
7793 : TEST SYNCHRONOUS DATA BUS READ
7794 : PATH FOR DATA BIT UNIQUENESS BY:
7795 :
7796 : 1. LOADING THE FIRST 16 WORDS IN THE IO_BUF WITH A SHIFTING
7797 : ZERO IN A FIELD OF ONES PATTERN.
7798 :
7799 : 2. VIA MBUS WRITE FUNCTION WRITE SHIFTING PATTERN INTO THE GOOD BLOCK.
7800 :
7801 : 3. CLEAR THE IO_BUF.
7802 :
7803 : 4. VIA MBUS READ FUNCTION READ THE SHIFTING PATTERN THROUGH THE
7804 : READ PATH.
7805 :
7806 : 5. INTERIGATE THE IO_BUF FOR THE SHIFTING PATTERN.
7807 :
7808 : IMPLICIT INPUTS:
7809 : IO_BUF
7810 : A VECTOR OF 256 WORDS WHERE DATA FOR MBUS READS AND WRITE FUNCTION ARD FOUND.
7811 : --
7812 :
7813 : local
7814 : DODU_FLG, !DROP UNIT FLAG
7815 : TST_PAT; !TEST PATTERN
7816 :
7817 : CLR MBUS;
7818 : DODU_FLG = ZERO;
7819 : TST_PAT = ONE; !ONE IN A FIELD OF ZEROES
7820 :
7821 : incr CNT from 0 to 15 do !WRITE 16 WORDS WITH SHIFTED 0 IN A FIELD OF 1'S
7822 : begin
7823 : IO_BUF [.CNT] = not .TST_PAT;
7824 : TST_PAT = .TST_PAT^ONE;
7825 : end;
7826 :
7827 : GD_BLK_XFER (); !SET UP A GOOD BLOCK XFERR
7828 : MLCS1 = write; !WRITE SHIFTING PATTERN
7829 :
7830 : do !DELAY UNTIL XFER TO COMPLETE
7831 : 0
7832 : until .DRY IS_SET;
7833 :
7834 : BGNSUB;
7835 :

```

```

16356 :ML4
16357 :
16358 :
16359 :      7836 incr CNT from 0 to 15 do
16360 :      7837      IO_BUF [.CNT] = ZEROES;
16361 :      7838
16362 :      7839 CLR MBUS;
16363 :      7840 GD_BLK_XFER ();
16364 :      7841 MLCS1 = read;
16365 :      7842
16366 :      7843 do
16367 :      7844      0
16368 :      7845 until .DRY IS_SET;
16369 :      7846
16370 :      7847 TST_PAT = ONE;
16371 :      7848
16372 :      7849 incr CNT from 0 to 15 do
16373 :      7850      begin
16374 :      7851
16375 :      7852      if .IO_BUF [.CNT] neq ( not .TST_PAT)
16376 :      7853      then
16377 :      7854          begin
16378 :      7855              ERRDF (93, SYNC, 0);
16379 :      7856              PRINTB (FIV_FMT, WRD_23, FNC_6, WRD_19, WRD_39, PHR_4);
16380 :      7857              PRINTB (FMT_2, ( not .TST_PAT), .IO_BUF [.CNT], ( not .TST_PAT xor .IO_BUF [.CNT]));
16381 :      7858              DODU_FLG = ONE;
16382 :      7859          end;
16383 :      7860
16384 :      7861          TST_PAT = .TST_PAT^ONE;
16385 :      7862          end;
16386 :      7863
16387 :      7864 ENDSUB;
16388 :      7865
16389 :      7866 if .DODU_FLG IS_SET
16390 :      7867 then
16391 :      7868          begin
16392 :      7869              DODU (.ML_LUN);
16393 :      7870              DOCLN;
16394 :      7871          end;
16395 :      7872
16396 :      7873 ENDTST;

```

```

22-Oct-1980 10:47:44 TOPS-20 Bliss-16 V2(206)
22-Oct-1980 10:45:32 PA:<NEALE>BL2ML4.BLI.2 (78)

!CLEAR OUT THE IO_BUF

!SET UP A GOOD BLOCK XFERR
!READ SHIFTING PATTERN THROUGH SYNC BUS

!DELAY UNTIL XFER TO COMPLETE

!SHIFTING PATTERN

!READ IO_BUF FOR SHIFTING 0 IN FIELD OF 1'S

!COMPARE IO_BUF TO SHIFTED PAT

!ERROR AND SET DODU_FLG IF NEQ

!SHIFT THE PATTERN AND REPEAT

!DROP THIS UNIT IF DODU_FLG IS_SET

```

```

16404 057244 004167 124610 $T36: JSR R1,$SAVE5 ; 7782
16405 057250 152777 000040 132462 BISB #40,@ML.REG+40 ; 7815
16406 057256 016705 132706 MOV ML.DUT,R5
16407 057262 042705 177770 BIC #177770,R5
16408 057266 142777 000007 132444 BICB #7,@ML.REG+40
16409 057274 150577 132440 BISB R5,@ML.REG+40

```

Address	OpCode	Operand 1	Operand 2	Operand 3	Label	Comment	Date/Time	Page
16411					:ML4		22-Oct-1980 10:47:44	TOPS
16412					:		22-Oct-1980 10:45:32	PA:<
16413								
16414	057300	005005				CLR R5		7818
16415	057302	012704	000001			MOV #1,R4		7819
16416	057306	005000				CLR R0		7821
16417	057310	010001			1\$:	MOV R0,R1		7823
16418	057312	006301				ASL R1		
16419	057314	010461	010342			MOV R4,IO.BUF(R1)		
16420	057320	005161	010342			COM IO.BUF(R1)		
16421	057324	006304				ASL R4		7824
16422	057326	005200				INC R0		7821
16423	057330	020027	000017			CMP R0,#17		
16424	057334	003765				BLE 1\$		
16425	057336	004767	133200			JSR PC,GD.BLK.XFER		7827
16426	057342	012777	000061	132330		MOV #61,@ML.REG		7828
16427	057350	105777	132374		2\$:	TSTB @ML.REG+50		7832
16428	057354	100375				BPL 2\$		
16429	057356	104402			3\$:	TRAP 2		
16430	057360	005000				CLR R0		7836
16431	057362	010001			4\$:	MOV R0,R1		7837
16432	057364	006301				ASL R1		
16433	057366	005061	010342			CLR IO.BUF(R1)		
16434	057372	005200				INC R0		7836
16435	057374	020027	000017			CMP R0,#17		
16436	057400	003770				BLE 4\$		
16437	057402	152777	000040	132330		BISB #40,@ML.REG+40		7837
16438	057410	016703	132554			MOV ML.DUT,R3		
16439	057414	042703	177770			BIC #177770,R3		
16440	057420	142777	000007	132312		BICB #7,@ML.REG+40		
16441	057426	150377	132306			BISB R3,@ML.REG+40		
16442	057432	004767	133104			JSR PC,GD.BLK.XFER		7840
16443	057436	012777	000071	132234		MOV #71,@ML.REG		7841
16444	057444	105777	132300		5\$:	TSTB @ML.REG+50		7845
16445	057450	100375				BPL 5\$		
16446	057452	012704	000001			MOV #1,R4		7847
16447	057456	005002				CLR R2		7849
16448	057460	010201			6\$:	MOV R2,R1		7852
16449	057462	006301				ASL R1		
16450	057464	012703	010342			MOV #IO.BUF,R3		
16451	057470	060103				ADD R1,R3		
16452	057472	010401				MOV R4,R1		
16453	057474	005101				COM R1		
16454	057476	021301				CMP (R3),R1		
16455	057500	001447				BEQ 7\$		
16456	057502	104455				TRAP 55		7855
16457	057504	000135				.WORD 135		
16458	057506	007500				.WORD SYNC		
16459	057510	000000				.WORD 0		
16460	057512	012746	006630			MOV #PHR.4,-(SP)		7856
16461	057516	012746	006250			MOV #WRD.39,-(SP)		
16462	057522	012746	006040			MOV #WRD.19,-(SP)		
16463	057526	012746	007030			MOV #FNC.6,-(SP)		
16464	057532	012746	006076			MOV #WRD.23,-(SP)		
16465	057536	012746	005414			MOV #FIV.FMT,-(SP)		


```

16467      ;ML4
16468      ;
16469
16470 057542 012746 000006      MOV      #6,-(SP)
16471 057546 010600      MOV      SP,R0      ; SP,*
16472 057550 104414      TRAP     14
16473 057552 011316      MOV      (R3),(SP)      ;
16474 057554 010146      MOV      R1,-(SP)      7857
16475 057556 046616 000002      BIC      2(SP),(SP)
16476 057562 040166 000002      BIC      R1,2(SP)
16477 057566 052616      BIS      (SP)+,(SP)
16478 057570 011346      MOV      (R3),-(SP)
16479 057572 010146      MOV      R1,-(SP)
16480 057574 012746 004224      MOV      #FMT.2,-(SP)
16481 057600 012746 000004      MOV      #4,-(SP)
16482 057604 010600      MOV      SP,R0      ; SP,*
16483 057606 104414      TRAP     14
16484 057610 012705 000001      MOV      #1,R5      ; *,DODU.FLG      7858
16485 057614 062706 000026      ADD      #26,SP      ;
16486 057620 006304      7$: ASL      R4      ; TST.PAT      7854
16487 057622 005202      INC      R2      ; CNT      7861
16488 057624 020227 000017      CMP      R2,#17      ; CNT,*      7849
16489 057630 003713      BLE      6$
16490 057632 104467      TRAP     67      ;
16491 057634 006000      ROR      R0      7862
16492 057636 103647      BLO      3$
16493 057640 005305      DEC      R5      ; DODU.FLG      7866
16494 057642 001004      BNE      8$
16495 057644 016700 132316      MOV      ML.LUN,R0      ;
16496 057650 104451      TRAP     51      ;
16497 057652 104444      TRAP     44      ;
16498 057654 000207      8$: RTS      PC      ;
16499
16500      ; Routine Size: 133 words
16501      ; Maximum stack depth per invocation: 17 words
16506
16507
16511
16515 057656      T36::
16516 057656 004767 177362      1$: JSR      PC,$T36      ;
16517 057662 104466      TRAP     66
16518 057664 006000      ROR      R0
16519 057666 103773      BLO      1$
16520 057670 000207      RTS      PC
16525 ;ML4
16526 ;
16527
16528 :      7875 :
16529 :      7876 :
16530 :      7877 BGNTST;
16531 :      7878
16532 :      7879 !++
16533 :      7880 ! TEST NUMBER: TST 37
16534 :      7881 !
16535 :      7882 ! TEST NAME: ARRAY ADDRESS MUX TEST
16536 :      7883 !
16537 :      7884 ! TEST DESCRIPTION:

```

```

16538 : 7885 :
16539 : 7886 : TEST FOR UNIQUE MOS RAM ROW
16540 : 7887 : AND COLUMN ADDRESSING BY:
16541 : 7888 :
16542 : 7889 : 1. FIRST FINDING A ERROR FREE
16543 : 7890 : 16K OR 64K CHUNK OF MEMORY.
16544 : 7891 : THIS REPRESENTS ONE ROW OF
16545 : 7892 : EITHER 16K OR 64K MOS RAMS.
16546 : 7893 :
16547 : 7894 : 2. WRITE A BACKGROUND OF ALL
16548 : 7895 : ONES INTO THE GOOD CHUNK
16549 : 7896 :
16550 : 7897 : 3. WRITE ZEROES INTO THE FIRST
16551 : 7898 : BLOCK OF THE GOOD CHUNK.
16552 : 7899 :
16553 : 7900 : 4. READ REMAINING BLOCKS IN
16554 : 7901 : GOOD CHUNK FOR ONES.
16555 : 7902 :
16556 : 7903 :
16557 : 7904 : IMPLICIT INPUTS:
16558 : 7905 :
16559 : 7906 : IO_BUF
16560 : 7907 :
16561 : 7908 : A VECTOR OF 256 WORDS
16562 : 7909 : WHERE DATA FOR MBUS
16563 : 7910 : READ AND WRITE TRANSFERS
16564 : 7911 : CAN BE FOUND.
16565 : 7912 :
16566 : 7913 :
16567 : 7914 : Local
16568 : 7915 : DSA_ADRS, !DSA ADRS COUNTER
16569 : 7916 : FND_GD_CHK; !FOUND GOOD 16K/64K CHUNK FLAG
16570 : 7917 :
16571 : 7918 : DSA_ADRS = ZEROES;
16572 : 7919 : DSA_ADRS = .DSA_ADRS - .RAS_INC; !REST DSA COUNT
16573 : 7920 : IO_BUF = ONES; !LOAD FIRST IO_BUF WORD WITH ONES
16574 : 7921 : BAI = ONE; !SET ON FIRST IO BUF WORD
16575 : 7922 : ECC_DIS = ONE; !DISABLE ECC
16576 : 7923 :
16577 : 7924 : do !DO UNTIL FOUND GOOD CHUNK OR LBT
16578 : 7925 : begin
16579 : 7926 : DSA_ADRS = .DSA_ADRS + .RAS_INC; !INCREMENT DSA ADRS COUNTER

```

```

16581 :ML4
16582 :
16583 :
16584 : 7927 MLWC = .W_C SIZE;
16585 : 7928 MLBA = IO_BUF;
16586 : 7929 MLDA = .DSA_ADRS;
16587 : 7930 ML_FUNC = write;
16588 : 7931
16589 : 7932 do
16590 : 7933 0
16591 : 7934 until .DRY IS_SET;
16592 : 7935
16593 : 7936 if .SC IS_NOT_SET
16594 : 7937 then
16595 : 7938 begin
16596 : 7939 MLWC = .W_C SIZE;
16597 : 7940 MLBA = IO_BUF;
16598 : 7941 MLDA = .DSA_ADRS;
16599 : 7942 ML_FUNC = WRT_CHK;
16600 : 7943
16601 : 7944 do
16602 : 7945 0
16603 : 7946 until .DRY IS_SET;
16604 : 7947
16605 : 7948 if .SC IS_NOT_SET
16606 : 7949 then
16607 : 7950 FND_GD_CHK = ONE
16608 : 7951 else
16609 : 7952 begin
16610 : 7953 CLR_MBUS;
16611 : 7954 BAI = ONE;
16612 : 7955 ECC_DIS = ONE;
16613 : 7956 end;
16614 : 7957
16615 : 7958 end
16616 : 7959 else
16617 : 7960 begin
16618 : 7961 CLR_MBUS;
16619 : 7962 BAI = ONE;
16620 : 7963 ECC_DIS = ONE;
16621 : 7964 end;
16622 : 7965
16623 : 7966 end
16624 : 7967 until (.FND_GD_CHK IS_SET ) or (.LBT IS_SET );
16625 : 7968
16626 : 7969 if .LBT IS_SET
16627 : 7970 then
16628 : 7971 begin
16629 : 7972 ERRDF (111, INTER, 0);
16630 : 7973 PRINTB (FIV_FMT, FNC_13, FNC_17, WRD_52, WRD_60, WRD_56);
16631 : 7974 PRINTB (THR_FMT, WRD_14, PHR_10, FNC_15);
16632 : 7975 DODU (.ML_LON);
16633 : 7976 DOCLN;
16634 : 7977 end
16635 : 7978 else

```

```

22-Oct-1980 10:47:44 TOPS-20 Bliss-16 V2(206)
22-Oct-1980 10:45:32 PA:<NEALE>BL2ML4.BLI.2 (79)

!16K OR 64K WORDS
!LOAD UBUS ADRS
!LOAD DSA ADRS
!DO A WRITE FUNCTION

!DELAY UNTIL XFER TO COMPLETE

!DID XFERR CAUSE AN SC

!XFERR WAS OK
!LOAD WORD COUNT
!LOAD UBUS ADRS
!LOAD DSA ADRS
!DO A WRITE CHECK FUNCTION

!DELAY UNTIL XFER TO COMPLETE

!IS THIS CHUNCK GOOD
!YES SET FLG
!NO, CLR AND DO NEXT CHUNCK

!THIS CHUNCK IS BAD TRY NEXT CHUNCK
!CLR AND TRY AT NEXT CHUNCK

!REPEAT UNTIL FOUND GOOD CHUNCK OR AT LBT
!IF AT LBT THEN ERROR AND DROP UNIT

!A GOOD CHUNCK WAS FOUND CONTINUE TEST

```

22-Oct-1980 10:47:44 TOPS-20 Bliss-16 V2(206)
 22-Oct-1980 10:45:32 PA:<NEALE>BL2ML4.BLI.2 (79)

```

16637 :ML4
16638 :
16639 :
16640 : 7979 begin
16641 : 7980 CLR_MBUS;
16642 : 7981 BAI = ONE;
16643 : 7982 ECC DIS = ONE;
16644 : 7983 IO_BUF = ZEROES;
16645 : 7984 MLDA = .DSA_ADRS;
16646 : 7985 MLWC = not 255;
16647 : 7986 MLBA = IO_BUF;
16648 : 7987 ML_FUNC = write;
16649 : 7988
16650 : 7989 do
16651 : 7990 0
16652 : 7991 until .DRY IS_SET;
16653 : 7992
16654 : 7993 CLR_MBUS;
16655 : 7994 BAI = ONE;
16656 : 7995 IO_BUF = ONES;
16657 : 7996 ECC DIS = ONE;
16658 : 7997 MLDA = .DSA_ADRS + 1;
16659 : 7998 MLBA = IO_BUF;
16660 : 7999 MLWC = .W_C_SIZE + 256;
16661 : 8000 ML_FUNC = WRT_CHK;
16662 : 8001
16663 : 8002 do
16664 : 8003 0
16665 : 8004 until .DRY IS_SET;
16666 : 8005
16667 : 8006 if .WCE IS_SET
16668 : 8007 then
16669 : 8008 begin
16670 : 8009 ERRDF (112, ASYNC, 0);
16671 : 8010 PRINTB (FOR_FMT, FNC 17, WRD_50, WRD_60, WRD_14);
16672 : 8011 PRINTB (FMT_9, (.MLDA - 1));
16673 : 8012 DODU (.ML_LUN);
16674 : 8013 DOCLN;
16675 : 8014 end;
16676 : 8015
16677 : 8016 end;
16678 : 8017
16679 : 8018 ENDTST;
16683 :
16687 057672 004167 124110 $T37: JSR R1,$SAVE2
16688 057676 005001 CLR R1
16689 057700 166701 131762 SUB RAS.INC,R1
16690 057704 012767 177777 130430 MOV #-1,IO.BUF

```

: 7873
 : DSA.ADRS 7918
 : *.DSA.ADRS 7919
 : 7920

Address	Op Code	Op 2	Op 3	Op 4	Label	Instruction	Comments	Line #
16692								
16693					:ML4			
16694					:			
16695	057712	152777	000010	132020		BISB #10,@ML.REG+40		7921
16696	057720	152777	000002	132072		BISB #2,@ML.REG+120		7922
16697	057726	066701	131734		1\$:	ADD RAS.INC,R1	* ,DSA.ADRS	7926
16698	057732	016777	131726	131750		MOV W.C.SIZE,@ML.REG+10		7927
16699	057740	012777	010342	131752		MOV #IO.BUF,@ML.REG+20		7928
16700	057746	010177	131756			MOV R1,@ML.REG+30	DSA.ADRS,*	7929
16701	057752	142777	000077	131720		BICB #77,@ML.REG		7930
16702	057760	152777	000061	131712		BISB #61,@ML.REG		
16703	057766	105777	131756		2\$:	TSTB @ML.REG+50		7934
16704	057772	100375				BPL 2\$		
16705	057774	032777	100000	131676		BIT #100000,@ML.REG		7936
16706	060002	001030				BNE 4\$		
16707	060004	016777	131654	131676		MOV W.C.SIZE,@ML.REG+10		7939
16708	060012	012777	010342	131700		MOV #IO.BUF,@ML.REG+20		7940
16709	060020	010177	131704			MOV R1,@ML.REG+30	DSA.ADRS,*	7941
16710	060024	142777	000077	131646		BICB #77,@ML.REG		7942
16711	060032	152777	000051	131640		BISB #51,@ML.REG		
16712	060040	105777	131704		3\$:	TSTB @ML.REG+50		7946
16713	060044	100375				BPL 3\$		
16714	060046	032777	100000	131624		BIT #100000,@ML.REG		7948
16715	060054	001003				BNE 4\$		
16716	060056	012700	000001			MOV #1,R0	* ,FND.GD.CHK	7950
16717	060062	000422				BP 5\$		7948
16718	060064	152777	000040	131646	4\$:	BISB #40,@ML.REG+40		7960
16719	060072	016702	132072			MOV ML.DUT,R2		
16720	060076	042702	177770			BIC #177770,R2		
16721	060102	142777	000007	131630		BICB #7,@ML.REG+40		
16722	060110	150277	131624			BISB R2,@ML.REG+40		
16723	060114	152777	000010	131616		BISB #10,@ML.REG+40		7962
16724	060122	152777	000002	131670		BISB #2,@ML.REG+120		7963
16725	060130	020027	000001		5\$:	CMP R0,#1	FND.GD.CHK,*	7967
16726	060134	001404				BEQ 6\$		
16727	060136	032777	002000	131604		BIT #2000,@ML.REG+50		
16728	060144	001670				BEQ 1\$		
16729	060146	032777	002000	131574	6\$:	BIT #2000,@ML.REG+50		7969
16730	060154	001447				BEQ 7\$		
16731	060156	104455				TRAP 55		7972
16732	060160	000157				.WORD 157		
16733	060162	007622				.WORD INTER		
16734	060164	000000				.WORD 0		
16735	060166	012746	006454			MOV #WRD.56,-(SP)		7973
16736	060172	012746	006510			MOV #WRD.60,-(SP)		
16737	060176	012746	006420			MOV #WRD.52,-(SP)		
16738	060202	012746	007206			MOV #FNC.17,-(SP)		
16739	060206	012746	007136			MOV #FNC.13,-(SP)		
16740	060212	012746	005414			MOV #FIV.FMT,-(SP)		
16741	060216	012746	000006			MOV #6,-(SP)		
16742	060222	010600				MOV SP,R0	SP,*	
16743	060224	104414				TRAP 14		
16744	060226	012716	007162			MOV #FNC.15,(SP)		7974
16745	060232	012746	006740			MOV #PHR.10,-(SP)		
16746	060236	012746	005774			MOV #WRD.14,-(SP)		

16748									
16749						:ML4			
16750						:			
16751	060242	012746	005366						
16752	060246	012746	000004						
16753	060252	010600							
16754	060254	104414							
16755	060256	016700	131704						7975
16756	060262	104451							
16757	060264	104444							
16758	060266	062706	000026						7971
16759	060272	000207							7969
16760	060274	152777	000040	131436	7\$:				7979
16761	060302	016702	131662						
16762	060306	042702	177770						
16763	060312	142777	000007	131420					
16764	060320	150277	131414						
16765	060324	152777	000010	131406					7981
16766	060332	152777	000002	131460					7982
16767	060340	005067	127776						7983
16768	060344	010177	131360						7984
16769	060350	012777	177400	131332					7985
16770	060356	012777	010342	131334					7986
16771	060364	142777	000077	131306					7987
16772	060372	152777	000061	131300					
16773	060400	105777	131344		8\$:				7991
16774	060404	100375							
16775	060406	152777	000040	131324					
16776	060414	016702	131550						
16777	060420	042702	177770						
16778	060424	142777	000007	131306					
16779	060432	150277	131302						
16780	060436	152777	000010	131274					7994
16781	060444	012767	177777	127670					7995
16782	060452	152777	000002	131340					7996
16783	060460	010102							7997
16784	060462	005202							
16785	060464	010277	131240						
16786	060470	012777	010342	131222					7998
16787	060476	016702	131162						7999
16788	060502	062702	000400						
16789	060506	010277	131176						
16790	060512	142777	000077	131160					8000
16791	060520	152777	000051	131152					
16792	060526	105777	131216		9\$:				8004
16793	060532	100375							
16794	060534	032777	040000	131176					8006
16795	060542	001441							
16796	060544	104455							8009
16797	060546	000160							
16798	060550	007444							
16799	060552	000000							
16800	060554	012746	005774						8010
16801	060560	012746	006510						
16802	060564	012746	006400						

22-Oct-1980 10:47:44 TOPS
22-Oct-1980 10:45:32 PA:<

```
16804 ;ML4
16805 ;
16806 ;
16807 060570 012746 007206      MOV    #FNC.17,-(SP)
16808 060574 012746 005400      MOV    #FOR.FMT,-(SP)
16809 060600 012746 000005      MOV    #5,-(SP)
16810 060604 010600              MOV    SP,R0                ; SP,*
16811 060606 104414              TRAP   14
16812 060610 017716 131114      MOV    @ML.REG+30,(SP)      ;
16813 060614 005316              DEC    (SP)
16814 060616 012746 004602      MOV    #FMT.9,-(SP)
16815 060622 012746 000002      MOV    #2,-(SP)
16816 060626 010600              MOV    SP,R0                ; SP,*
16817 060630 104414              TRAP   14
16818 060632 016700 131330      MOV    ML.LUN,R0           ;
16819 060636 104451              TRAP   51
16820 060640 104444              TRAP   44
16821 060642 062706 000020      ADD    #20,SP              ;
16822 060646 000207      10$: RTS    PC              ;
16823
16824 ; Routine Size: 247 words
16825 ; Maximum stack depth per invocation: 14 words
16830
16831
16835
16839 060650      T37::
16840 060650 004767 177016      1$: JSR    PC,$T37          ;
16841 060654 104466              TRAP   66
16842 060656 006000              ROR    R0
16843 060660 103773              BLO    1$
16844 060662 000207              RTS    PC
16845
16846 ; Routine Size: 6 words
16847 ; Maximum stack depth per invocation: 0 words
16852
16853
16854 ;      8019 !<BLF/PAGE>
```

22-Oct-1980 10:47:44
22-Oct-1980 10:45:32

TOPS-20 Bliss-16 V2(206)
PA:<NEALE>BL2ML4.BLI.2 (80)

16856 :ML4
16857 :
16858 :
16859 :
16860 :
16861 :
16862 :
16863 :
16864 :
16865 :
16866 :
16867 :
16868 :
16869 :
16870 :
16871 :
16872 :
16873 :
16874 :
16875 :
16876 :
16877 :
16878 :
16879 :
16880 :
16881 :
16882 :
16883 :
16884 :
16885 :
16886 :
16887 :
16888 :
16889 :
16890 :
16891 :
16892 :
16893 :
16894 :
16895 :
16896 :
16897 :
16898 :
16899 :
16900 :
16901 :
16902 :
16903 :
16904 :
16905 :
16906 :
16907 :
16908 :
16909 :
16910 :

```

8020 !
8021 BGNTST;
8022
8023 !++
8024 TEST NUMBER: TST 38
8025
8026 TEST NAME: NIBBLE OFFSET TEST
8027
8028 TEST DESCRIPTION:
8029 TEST NIBBLE OFFSET COUNTERS TO OFFSET GOOD NIBBLE DATA A MAX
8030 OF 14 WORDS ON DETECTION OF ALL BAD NIBBLES BY:
8031
8032 1. LOADING FIRST 2 1/4 WORDS OF THE IO_BUF WITH ZEROES AND THE
8033 REMAINING OF BUFFER WITH ONES.
8034
8035 2. VIA DAT_DM MODE WRITE THE GOOD BLOCK WITH BACKGROUND
8036 ON ONES.
8037
8038 3. VIA PROM R/W MODE FORCE ALL ARRAY NIBBLES BAD.
8039
8040 4. VIA A MBUS WRITE FUNCTION LOAD IO_BUF INTO THE GOOD BLOCK.
8041
8042 5. VIA DAT_DM MODE READ FIRST 15 ARRAY WORDS FOR ZEROES AND THE
8043 REMAINING WORDS FOR ONES.
8044
8045 IMPLICIT INPUTS:
8046 PD TEMP
8047 A BIT VECTOR OF 16 BITS WHERE THE READ PROM DATA IS STORED AND
8048 ACCESSED FROM.
8049
8050 IO_BUF
8051 A VECTOR OF 256 WORDS WHERE DATA FOR MBUS READS AND WRITE
8052 FUNCTIONS ARE FOUND.
8053 --
8054
8055 local
8056 DODU_FLG, !DROP UNIT FLAG
8057 TST_PAT, !TEST PATTERN
8058 START, !STARTING WORD
8059 FINISH, !ENDING WORD
8060 ERR_FLG; !ERROR FLAG
8061
8062 BGNSUB;
8063 CLR MBUS;
8064 DODU_FLG = ZERO;
8065
8066 incr WD_CNT from 0 to 255 do !LOAD IO_BUF WITH ONES
8067 IO_BUF [.WD_CNT] = ONES;
8068
8069 IO_BUF [0] = ZEROES; !LOAD FIRST 2 1/4 WORDS WITH ZEROES
8070 IO_BUF [1] = ZEROES;
8071 IO_BUF [2] = %o'177760';

```


22-Oct-1980 10:47:44 TOPS-20 Bliss-16 V2(206)
 22-Oct-1980 10:45:32 PA:<NEALE>BL2ML4.BLI.2 (80)

```

16912 :ML4
16913 :
16914 :
16915 :      8072 MLD1 = ONES;
16916 :      8073 MLD2 = ONES;
16917 :      8074 MLE2 = ONES;
16918 :      8075 DAT_DM_XFER ();
16919 :      8076 MLC51 = write;
16920 :      8077
16921 :      8078 incr WD_CNT from 0 to 127 do
16922 :      8079      begin
16923 :      8080      DELAY (ONE_US);
16924 :      8081      DAT_CLK = ONE;
16925 :      8082      end;
16926 :      8083
16927 :      8084 CLR_MBUS;
16928 :      8085 WRT_PD (ONES, 19);
16929 :      8086 PROM_RW = ONE;
16930 :      8087 GD_BLK_XFER ();
16931 :      8088 MLC51 = write;
16932 :      8089
16933 :      8090
16934 :      8091 do
16935 :      8092      0
16936 :      8093 until .DRY IS_SET;
16937 :      8094
16938 :      8095 CLR_MBUS;
16939 :      8096 START = ZERO;
16940 :      8097 FINISH = 14;
16941 :      8098 TST_PAT = ZEROES;
16942 :      8099 DAT_DM_XFER ();
16943 :      8100 MLC51 = read;
16944 :      8101 DELAY (ONE_US);
16945 :      8102
16946 :      8103 incr TWICE from 0 to 1 do
16947 :      8104      begin
16948 :      8105
16949 :      8106      incr WRD_CNT from .START to .FINISH do
16950 :      8107      begin
16951 :      8108      PD_TEMP = .MLPD;
16952 :      8109      DAT_CLK = ONE;
16953 :      8110      DELAY (ONE_US);
16954 :      8111      RD_LNG_WRD;
16955 :      8112
16956 :      8113      incr NIB_PTR from 0 to 8 do
16957 :      8114
16958 :      8115      if .PD_TEMP [.NIB_PTR] IS_NOT_SET
16959 :      8116      then
16960 :      8117      begin
16961 :      8118      TST_LNG_WRD (.NIB_PTR, .TST_PAT, ERR_FLG);
16962 :      8119
16963 :      8120      if .ERR_FLG IS_SET
16964 :      8121      then
16965 :      8122      begin
16966 :      8123      ERRDF (94, ARR_DAT, 0);
  
```

```

!LOAD DATA DIAG REGISTERS WITH ONES

!SET UP A DATA DIAG MODE XFERR
!DO A WRITE XFERR

!LOAD BLOCK WITH BG PAT

!FORCE ALL NIBBLES BAD
!SET PROM READ WRITE
!SET UP A GOOD BLOCK XFERR
!WRITE ZEROES IN FIRST BLOCK WORD LOCATION
!OFFSETTING 14 NIBBLES WITH ZEROES ALSO.

!DELAY UNTIL XFER TO COMPLETE

!START AT THE FIRST BLOCK WORD
!END AT THE 14'TH BLOCK WORD
!TEST FOR ZEROES IN FIRST 14 WORDS
!SET UP A DATA DIAG MODE XFERR
!DO A READ FUNCTION

!READ WORDS 0-14 FOR 0'S AND 15-126 FOR 1'S

!READ BLOCK WORDS FORM START TO FINISH

!GET PROM DATA
!CLOCK OUT THE DATA WORD

!READ THE DATA DIAG REGISTERS

!LOOK AT 9 NIBBLES

!FIND GOOD NIBBLES

!COMPARE NIBBLE WITH TST PAT

!SEE IF COMPARE FOUND AN ERROR

!ERROR AND SET DODU_FLG IF ERROR FLG SET
  
```

16968 :ML4

22-Oct-1980 10:47:44
22-Oct-1980 10:45:32

TOPS-20 Bliss-16 V2(206)
PA:<NEALE>BL2ML4.BLI.2 (80)

16969 :

16970 :

16971 : 8124

16972 : 8125

16973 : 8126

16974 : 8127

16975 : 8128

16976 : 8129

16977 : 8130

16978 : 8131

16979 : 8132

16980 : 8133

16981 : 8134

16982 : 8135

16983 : 8136

16984 : 8137

16985 : 8138

16986 : 8139

16987 : 8140

16988 : 8141

16989 : 8142

16990 : 8143

16991 : 8144

16992 : 8145

16993 : 8146

16994 : 8147

16998 :

8124

8125

8126

8127

8128

8129

8130

8131

8132

8133

8134

8135

8136

8137

8138

8139

8140

8141

8142

8143

8144

8145

8146

8147

PRINTB (THR_FMT, WRD_41, WRD_46, WRD_10);
PRINTB (FMT_6, .NIB_PTR);
DODU_FLG = ONE;
end;

end;

end;

TST PAT = not .TST_PAT;
START = 15;
FINISH = 126;
end

!NOW READ FOR ONES
!START A 15
!END AT 126

ENDSUB;

if .DODU_FLG IS_SET
then
begin
DODU (.ML_LUN);
DOCLN;
end;

!DROP THIS UNIT IF DODU_FLG SET

ENDTST;

17002 060664 004167 123170
17003 060670 162706 000012
17004 060674 104402
17005 060676 152777 000040 131034
17006 060704 016704 131260
17007 060710 042704 177770
17008 060714 142777 000007 131016
17009 060722 150477 131012
17010 060726 005066 000004
17011 060732 005002
17012 060734 010203
17013 060736 006303
17014 060740 012763 177777 010342
17015 060746 005202
17016 060750 020227 000377
17017 060754 003767
17018 060756 005067 127360
17019 060762 005067 127356
17020 060766 012767 177760 127352
17021 060774 012777 177777 131066

\$T38:

1\$:

2\$:

JSR R1,\$SAVE5
SUB #12,SP
TRAP 2
BISB #40,@ML.REG+40
MOV ML.DUT,R4
BIC #177770,R4
BICB #7,@ML.REG+40
BISB R4,@ML.REG+40
CLR 4(SP)
CLR R2
MOV R2,R3
ASL R3
MOV #-1,IO.BUF(R3)
INC R2
CMP R2,#377
BLE 2\$
CLR IO.BUF
CLR IO.BUF+2
MOV #-20,IO.BUF+4
MOV #-1,@ML.REG+170

:
:
:
:
:
: DODU.FLG
: WD.CNT
: WD.CNT,*
:
: WD.CNT
: WD.CNT,*
:
:
:
:

8018
8060
8062
8064
8065
8067
8066
8069
8070
8071
8072

Address	Op Code	Operand 1	Operand 2	Operand 3	Instruction	Comments	Line
17023							
17024							
17025							
17026	061002	012777	177777	131070	MOV	#-1,@ML.REG+200	8073
17027	061010	012777	177777	131042	MOV	#-1,@ML.REG+160	8074
17028	061016	004767	131604		JSR	PC,DAT.DM.XFER	8075
17029	061022	012777	000061	130650	MOV	#61,@ML.REG	8076
17030	061030	005001			CLR	R1	8078
17031	061032	012702	000001		MOV	#1,R2	8080
17032	061036	001411			BEQ	7\$	
17033	061040	016703	121052		MOV	L\$DLY,R3	
17034	061044	001404			BEQ	6\$	
17035	061046	005066	000010		CLR	10(SP)	
17036	061052	005303			DEC	R3	
17037	061054	001374			BNE	5\$	
17038	061056	005302			DEC	R2	
17039	061060	000766			BR	4\$	
17040	061062	152777	000020	130730	BISB	#20,@ML.REG+120	8081
17041	061070	005201			INC	R1	8078
17042	061072	020127	000177		CMP	R1,#177	
17043	061076	003755			BLE	3\$	
17044	061100	152777	000040	130632	BISB	#40,@ML.REG+40	8082
17045	061106	016704	131056		MOV	ML.DUT,R4	
17046	061112	042704	177770		BIC	#177770,R4	
17047	061116	142777	000007	130614	BICB	#7,@ML.REG+40	
17048	061124	150477	130610		BISB	R4,@ML.REG+40	
17049	061130	012746	177777		MOV	#-1,-(SP)	8085
17050	061134	012746	000023		MOV	#23,-(SP)	
17051	061140	004767	134566		JSR	PC,WRT.PD	
17052	061144	152777	000100	130646	BISB	#100,@ML.REG+120	8086
17053	061152	004767	131364		JSR	PC,GD.BLK.XFER	8087
17054	061156	012777	000061	130514	MOV	#61,@ML.REG	8088
17055	061164	105777	130560		TSTB	@ML.REG+50	8093
17056	061170	100375			BPL	8\$	
17057	061172	152777	000040	130540	BISB	#40,@ML.REG+40	
17058	061200	016704	130764		MOV	ML.DUT,R4	
17059	061204	042704	177770		BIC	#177770,R4	
17060	061210	142777	000007	130522	BICB	#7,@ML.REG+40	
17061	061216	150477	130516		BISB	R4,@ML.REG+40	
17062	061222	005066	000006		CLR	6(SP)	8096
17063	061226	012766	000016	000004	MOV	#16,4(SP)	8097
17064	061234	005005			CLR	R5	8098
17065	061236	004767	131364		JSR	PC,DAT.DM.XFER	8099
17066	061242	012777	000071	130430	MOV	#71,@ML.REG	8100
17067	061250	012702	000001		MOV	#1,R2	8101
17068	061254	001411			BEQ	12\$	
17069	061256	016703	120634		MOV	L\$DLY,R3	
17070	061262	001404			BEQ	11\$	
17071	061264	005066	000014		CLR	14(SP)	
17072	061270	005303			DEC	R3	
17073	061272	001374			BNE	10\$	
17074	061274	005302			DEC	R2	
17075	061276	000766			BR	9\$	
17076	061300	005004			CLR	R4	8103
17077	061302	016601	000006		MOV	6(SP),R1	8106

22-Oct-1980 10:47:44 TOPS
22-Oct-1980 10:45:32 PA:<

17079						:ML4					22-Oct-1980 10:47:44	TOPS
17080						:					22-Oct-1980 10:45:32	PA:<
17081												
17082	061306	005301					DEC	R1		: WRD.CNT		
17083	061310	000534					BR	22\$				
17084	061312	017767	130612	130342	14\$:		MOV	@ML.REG+230,PD.TEMP				8108
17085	061320	152777	000020	130472			BISB	#20,@ML.REG+120				8109
17086	061326	012702	000001				MOV	#1,R2		: *,\$\$TMP2		8110
17087	061332	001411			15\$:		BEQ	18\$				
17088	061334	016703	120556				MOV	L\$DLY,R3		: *,\$\$TMP1		
17089	061340	001404					BEQ	17\$				
17090	061342	005066	000014		16\$:		CLR	14(SP)		: \$\$TMP		
17091	061346	005303					DEC	R3		: \$\$TMP1		
17092	061350	001374					BNE	16\$				
17093	061352	005302			17\$:		DEC	R2		: \$\$TMP2		
17094	061354	000766					BR	15\$				
17095	061356	017767	130506	126330	18\$:		MOV	@ML.REG+170,D1.TEMP				
17096	061364	017767	130510	126324			MOV	@ML.REG+200,D2.TEMP				
17097	061372	017767	130462	126320			MOV	@ML.REG+160,E2.TEMP				
17098	061400	005002					CLR	R2		: NIB.PTR		8113
17099	061402	010203			19\$:		MOV	R2,R3		: NIB.PTR,*		8115
17100	061404	006203					ASR	R3				
17101	061406	006203					ASR	R3				
17102	061410	006203					ASR	R3				
17103	061412	062703	011662				ADD	#PD.TEMP,R3				
17104	061416	010346					MOV	R3,-(SP)				
17105	061420	010246					MOV	R2,-(SP)		: NIB.PTR,*		
17106	061422	042716	177770				BIC	#177770,(SP)				
17107	061426	012746	000001				MOV	#1,-(SP)				
17108	061432	005046					CLR	-(SP)				
17109	061434	004767	121442				JSR	PC,BL\$GT2				
17110	061440	062706	000010				ADD	#10,SP				
17111	061444	005700					TST	R0				
17112	061446	001051					BNE	21\$				
17113	061450	010246					MOV	R2,-(SP)		: NIB.PTR,*		8118
17114	061452	010546					MOV	R5,-(SP)		: TST.PAT,*		
17115	061454	012746	000020				MOV	#20,-(SP)				
17116	061460	060616					ADD	SP,(SP)		: ERR.FLG,*		
17117	061462	004767	131172				JSR	PC,TST.LNG.WRD				
17118	061466	026627	000020	000001			CMP	20(SP),#1		: ERR.FLG,*		8120
17119	061474	001034					BNE	20\$				
17120	061476	104455					TRAP	55				8123
17121	061500	000136					.WORD	136				
17122	061502	007534					.WORD	ARR.DAT				
17123	061504	000000					.WORD	0				
17124	061506	012746	005740				MOV	#WRD.10,-(SP)				8124
17125	061512	012746	006336				MOV	#WRD.46,-(SP)				
17126	061516	012746	006264				MOV	#WRD.41,-(SP)				
17127	061522	012746	005366				MOV	#THR.FMT,-(SP)				
17128	061526	012746	000004				MOV	#4,-(SP)				
17129	061532	010600					MOV	SP,R0		: SP,*		
17130	061534	104414					TRAP	14				
17131	061536	010216					MOV	R2,(SP)		: NIB.PTR,*		8125
17132	061540	012746	004470				MOV	#FMT.6,-(SP)				
17133	061544	012746	000002				MOV	#2,-(SP)				

17135	061550	010600			MOV	SP,R0		; SP,*	
17136	061552	104414			TRAP	14			
17137	061554	012766	000001	000034	MOV	#1,34(SP)		; *,DODU.FLG	8126
17138	061562	062706	000016		ADD	#16,SP			8122
17139	061566	062706	000000		ADD	#6,SP			8117
17140	061572	005202			20\$: INC	R2		; NIB.PTR	8113
17141	061574	020227	000010		21\$: CMP	R2,#10		; NIB.PTR,*	
17142	061600	003700			BLE	19\$			
17143	061602	005201			22\$: INC	R1		; WRD.CNT	8106
17144	061604	020166	000004		CMP	R1,4(SP)		; WRD.CNT,FINISH	
17145	061610	003640			BLE	14\$			
17146	061612	005105			COM	R5		; TST.PAT	8133
17147	061614	012766	000017	000006	MOV	#17,6(SP)		; *,START	8134
17148	061622	012766	000176	000004	MOV	#176,4(SP)		; *,FINISH	8135
17149	061630	005204			INC	R4		; TWICE	8103
17150	061632	020427	000001		CMP	R4,#1		; TWICE,*	
17151	061636	003621			BLE	13\$			
17152	061640	022626			CMP	(SP)+,(SP)+			8060
17153	061642	104467			TRAP	67			8136
17154	061644	006000			ROR	R0			
17155	061646	103002			BHIS	23\$			
17156	061650	000167	177020		JMP	1\$			
17157	061654	026627	000004	000001	23\$: CMP	4(SP),#1		; DODU.FLG,*	8140
17158	061662	001004			BNE	24\$			
17159	061664	016700	130276		MOV	ML.LUN,R0			8143
17160	061670	104451			TRAP	51			
17161	061672	104444			TRAP	44			
17162	061674	062706	000012		24\$: ADD	#12,SP			8018
17163	061700	000207			RTS	PC			
17164									
17165									
17166									
17171									
17172									
17176									
17180	061702				T38::				
17181	061702	004767	176756		1\$: JSR	PC,\$T38			8145
17182	061706	104466			TRAP	66			
17183	061710	006000			ROR	R0			
17184	061712	103773			BLO	1\$			
17185	061714	000207			RTS	PC			

; Routine Size: 263 words
; Maximum stack depth per invocation: 23 words

22-Oct-1980 10:47:44
 22-Oct-1980 10:45:32

TOPS-20 Bliss-16 V2(206)
 PA:<NEALE>BL2ML4.BLI.2 (81)

17191 :ML4
 17192 :
 17193 :
 17194 :
 17195 :
 17196 :
 17197 :
 17198 :
 17199 :
 17200 :
 17201 :
 17202 :
 17203 :
 17204 :
 17205 :
 17206 :
 17207 :
 17208 :
 17209 :
 17210 :
 17211 :
 17212 :
 17213 :
 17214 :
 17215 :
 17216 :
 17217 :
 17218 :
 17219 :
 17220 :
 17221 :
 17222 :
 17223 :
 17224 :
 17225 :
 17226 :
 17227 :
 17228 :
 17229 :
 17230 :
 17231 :
 17232 :
 17233 :
 17234 :
 17235 :
 17236 :
 17237 :
 17238 :
 17239 :
 17240 :
 17241 :
 17242 :
 17243 :
 17244 :
 17245 :

8149
 8150
 8151
 8152
 8153
 8154
 8155
 8156
 8157
 8158
 8159
 8160
 8161
 8162
 8163
 8164
 8165
 8166
 8167
 8168
 8169
 8170
 8171
 8172
 8173
 8174
 8175
 8176
 8177
 8178
 8179
 8180
 8181
 8182
 8183
 8184
 8185
 8186
 8187
 8188
 8189
 8190
 8191
 8192
 8193
 8194
 8195
 8196
 8197
 8198
 8199
 8200

```

!
BGNTST;
!++
TEST NUMBER: TST 39
TEST NAME: CS1 FUNCTION ABORT TEST
TEST DESCRIPTION:
TEST CS1 FUNCTION ABORTS ON DETECTION OF CLASS 'A' & 'B' ERRORS BY:
1. VIA MBUS WRITE FUNCTION LOAD THE GOOD BLOCK WITH BACKGROUND
   PATTERN OF ONES.
2. CLEAR THE IO_BUF
3. DO A MBUS READ FUNCTION. WHILE THE READ IS IN PROGRESS WRITE TO
   MLDA (CLASS 'A' ERROR) READ THE IO_BUF FOR ONES.
4. CLEAR THE IO_BUF
5. VIA PROM R/W MODE FORCE UV ERROR TO THE UV ADRS ERROR PROM
   (CLASS 'B' ERROR). DO A MBUS READ.
6. READ IO_BUF FOR ZEROES

IMPLICIT INPUTS:
IO_BUF
A VECTOR OF 256 WORDS WHERE DATA FOR MBUS READS AND
WRITE FUNCTION ARE FOUND.
--

CLR_MBUS;
BAI = ONE; !SET ON FIRST IO_BUF ADRS
IO_BUF = ONES; !LOAD FIRST IO_BUF ADRS
GD_BLK_XFER (); !SET UP A GOOD_BLOCK XFERR
MLCS1 = write; !WRITE BACKGROUND PATTERN

do !DELAY UNTIL XFER TO COMPLETE
  0
until .DRY IS_SET;

incr TWICE from 0 to 1 do !FORCE CLASS 'A' AND CLASS 'B' ERRORS
  begin
  BGNSUB;
  CLR_MBUS;

  incr CNT from 0 to 255 do
    IO_BUF [.CNT] = ZEROES; !CLEAR OUT IO_BUF

  if .TWICE eql 1 !IF 2ND PASS THEN FORCE 'B' ERROR
  then
  
```

22-Oct-1980 10:47:44 TOPS-20 Bliss-16 V2(206)
22-Oct-1980 10:45:32 PA:<NEALE>BL2ML4.BLI.2 (81)

```

17247 :ML4
17248 :
17249 :
17250 :      8201      begin
17251 :      8202      PROM_RW = ONE;
17252 :      8203      MLPD = %o'777';
17253 :      8204      end;
17254 :      8205
17255 :      8206      GD_BLK_XFER ();
17256 :      8207      MLCS1 = read;
17257 :      8208
17258 :      8209      if .TWICE eql 0 then MLDA = ONES;
17259 :      8210
17260 :      8211      do
17261 :      8212      0
17262 :      8213      until .DRY IS_SET;
17263 :      8214
17264 :      8215      if .TWICE eql 0
17265 :      8216      then
17266 :      8217      begin
17267 :      8218
17268 :      8219      incr WRD_CNT from 0 to 64 do
17269 :      8220      begin
17270 :      8221
17271 :      8222      if .IO_BUF [.WRD_CNT] neq ONES
17272 :      8223      then
17273 :      8224      begin
17274 :      8225      ERRDF (95, SYNC, 0);
17275 :      8226      PRINTB (FOR_FMT, PHR_8, FNC_13, WRD_19, WRD_10);
17276 :      8227      exitloop;
17277 :      8228      end;
17278 :      8229
17279 :      8230      end;
17280 :      8231
17281 :      8232      end
17282 :      8233      else
17283 :      8234      begin
17284 :      8235
17285 :      8236      incr WRD_CNT from 0 to 64 do
17286 :      8237      begin
17287 :      8238
17288 :      8239      if .IO_BUF [.WRD_CNT] neq ZEROES
17289 :      8240      then
17290 :      8241      begin
17291 :      8242      ERRDF (96, SYNC, 0);
17292 :      8243      PRINTB (FOR_FMT, PHR_9, FNC_13, WRD_19, WRD_10);
17293 :      8244      exitloop;
17294 :      8245      end;
17295 :      8246
17296 :      8247      end;
17297 :      8248
17298 :      8249      end;
17299 :      8250
17300 :      8251      if .SC IS_NOT_SET
17301 :      8252      then

```

```

!SET UP A GOOD BLOCK XFERR
!DO A READ FUNCTION
!IF FIRST PASS THEN FORCE AN 'A' ERROR
!DELAY UNTIL XFER TO COMPLETE
!SEE WHICH PASS WE'RE ON
!CLASS 'A' ERROR
!SEE IF XFERR WAY ALLOWED TO CONTINUE
!READ IO_BUF FOR BG PAT
!ERROR AND EXIT LOOP IF ZEROES
!CLASS 'B' ERROR
!SEE IF XFERR WAS ABORTED
!READ IO_BUF CLEARED DATA
!ERROR IF ONES AND EXIT LOOP
!SEE IF SC BIT SET

```

17303 :ML4

22-Oct-1980 10:47:44

TOPS-20 Bliss-16 V2(206)

17304 :

22-Oct-1980 10:45:32

PA:<NEALE>BL2ML4.BLI.2 (81)

17305

17306 : 8253

begin

!ERROR IF NOT SET

17307 : 8254

ERRDF (104, SYNC, 0);

17308 : 8255

PRINTB (FIV_FMT, WRD_59, PHR_1, WRD_11, WRD_19, FNC_13)

17309 : 8256

end;

17310 : 8257

17311 : 8258

ENDSUB;

17312 : 8259

end;

17313 : 8260

17314 : 8261

ENDTST;

17318

17322 061716 004167 122100 \$T39: JSR R1,\$SAVE3 : 8147

17323 061722 152777 000040 130010 BISB #40,@ML.REG+40 : 8150

17324 061730 016703 130234 MOV ML.DUT,R3 :

17325 061734 042703 177770 BIC #177770,R3 :

17326 061740 142777 000007 127772 BICB #7,@ML.REG+40 :

17327 061746 150377 127766 BISB R3,@ML.REG+40 :

17328 061752 152777 000010 127760 BISB #10,@ML.REG+40 : 8182

17329 061760 012767 177777 126354 MOV #-1,IO.BUF : 8183

17330 061766 004767 130550 JSR PC,GD.BLK.XFER : 8184

17331 061772 012777 000061 127700 MOV #61,@ML.REG : 8185

17332 062000 105777 127744 1\$: TSTB @ML.REG+50 : 8189

17333 062004 100375 BPL 1\$:

17334 062006 005003 CLR R3 : TWICE 8191

17335 062010 104402 2\$: TRAP 2 : 8192

17336 062012 152777 000040 127720 BISB #40,@ML.REG+40 : 8193

17337 062020 016702 130144 MOV ML.DUT,R2 :

17338 062024 042702 177770 BIC #177770,R2 :

17339 062030 142777 000007 127702 BICB #7,@ML.REG+40 :

17340 062036 150277 127676 BISB R2,@ML.REG+40 :

17341 062042 005000 CLR R0 : CNT 8196

17342 062044 010001 3\$: MOV R0,R1 : CNT,* 8197

17343 062046 006301 ASL R1 :

17344 062050 005061 010342 CLR IO.BUF(R1) :

17345 062054 005200 INC R0 : CNT 8196

17346 062056 020027 000377 CMP R0,#377 : CNT,*

17347 062062 003770 BLE 3\$:

17348 062064 020327 000001 CMP R3,#1 : TWICE,* 8199

17349 062070 001006 BNE 4\$:

17350 062072 152777 000100 127720 BISB #100,@ML.REG+120 : 8202

17351 062100 012777 000777 130022 MOV #777,@ML.REG+230 : 8203

17352 062106 004767 130430 4\$: JSR PC,GD.BLK.XFER : 8206

17353 062112 012777 000071 127560 MOV #71,@ML.REG : 8207

17354 062120 005002 CLR R2 : 8209

17355 062122 005703 TST R3 : TWICE

17356 062124 001004 BNE 5\$

17358												
17359												
17360												
17361	062126	005202										
17362	062130	012777	177777	127572								
17363	062136	105777	127606		5\$:	INC	R2					
17364	062142	100375				MOV	#-1,@ML.REG+30					
17365	062144	006002				TSTB	@ML.REG+50					8213
17366	062146	103037				BPL	5\$					
17367	062150	005002				ROR	R2					8215
17368	062152	010201				BCC	8\$					
17369	062154	006301				CLR	R2					8219
17370	062156	026127	010342	177777	6\$:	MOV	R2,R1					8222
17371	062164	001423				ASL	R1					
17372	062166	104455				CMP	IO.BUF(R1),#-1					
17373	062170	000137				BEQ	7\$					
17374	062172	007500				TRAP	55					8225
17375	062174	000000				.WORD	137					
17376	062176	012746	005740			.WORD	SYNC					
17377	062202	012746	006040			.WORD	0					8226
17378	062206	012746	007136			MOV	#WRD.10,-(SP)					
17379	062212	012746	006714			MOV	#WRD.19,-(SP)					
17380	062216	012746	005400			MOV	#FNC.13,-(SP)					
17381	062222	012746	000005			MOV	#PHR.8,-(SP)					
17382	062226	010600				MOV	#FOR.FMT,-(SP)					
17383	062230	104414				MOV	#5,-(SP)					
17384	062232	000435				MOV	SP,R0					
17385	062234	005202				TRAP	14					
17386	062236	020227	000100		7\$:	BR	10\$					8227
17387	062242	003743				INC	R2					8219
17388	062244	000437				CMP	R2,#100					
17389	062246	005002				BLE	6\$					
17390	062250	010201				BR	12\$					8215
17391	062252	006301				CLR	R2					8236
17392	062254	005761	010342		8\$:	MOV	R2,R1					8239
17393	062260	001425			9\$:	ASL	R1					
17394	062262	104455				TST	IO.BUF(R1)					
17395	062264	000140				BEQ	11\$					
17396	062266	007500				TRAP	55					8242
17397	062270	000000				.WORD	140					
17398	062272	012746	005740			.WORD	SYNC					
17399	062276	012746	006040			.WORD	0					8243
17400	062302	012746	007136			MOV	#WRD.10,-(SP)					
17401	062306	012746	006726			MOV	#WRD.19,-(SP)					
17402	062312	012746	005400			MOV	#FNC.13,-(SP)					
17403	062316	012746	000005			MOV	#PHR.9,-(SP)					
17404	062322	010600				MOV	#FOR.FMT,-(SP)					
17405	062324	104414				MOV	#5,-(SP)					
17406	062326	062706	000014			MOV	SP,R0					
17407	062332	000404			10\$:	TRAP	14					
17408	062334	005202				ADD	#14,SP					8244
17409	062336	020227	000100			BR	12\$					
17410	062342	003742			11\$:	INC	R2					8236
17411	062344	032777	100000	127326	12\$:	CMP	R2,#100					
17412	062352	001026				BLE	9\$					
						BIT	#100000,@ML.REG					8251
						BNE	13\$					

```

17414 ;ML4
17415 ;
17416 ;
17417 062354 104455 TRAP 55 ;
17418 062356 000150 .WORD 150 ;
17419 062360 007500 .WORD SYNC ;
17420 062362 000000 .WORD 0 ;
17421 062364 012746 007136 MOV #FNC.13,-(SP) ;
17422 062370 012746 006040 MOV #WRD.19,-(SP) ;
17423 062374 012746 005750 MOV #WRD.11,-(SP) ;
17424 062400 012746 006542 MOV #PHR.1,-(SP) ;
17425 062404 012746 006504 MOV #WRD.59,-(SP) ;
17426 062410 012746 005414 MOV #FIV.FMT,-(SP) ;
17427 062414 012746 000006 MOV #6,-(SP) ;
17428 062420 010600 MOV SP,R0 ; SP,*
17429 062422 104414 TRAP 14 ;
17430 062424 062706 000016 ADD #16,SP ;
17431 062430 104467 13$: TRAP 67 ;
17432 062432 006000 ROR R0 ;
17433 062434 103002 BHIS 15$ ;
17434 062436 000167 177346 14$: JMP 2$ ;
17435 062442 005203 15$: INC R3 ; TWICE
17436 062444 020327 000001 CMP R3,#1 ; TWICE,*
17437 062450 003772 BLE 14$ ;
17438 062452 000207 RTS PC ;
17439 ;
17440 ; Routine Size: 175 words
17441 ; Maximum stack depth per invocation: 11 words
17446 ;
17447 ;
17451 ;
17455 062454 T39::
17456 062454 004767 177236 1$: JSR PC,$T39 ;
17457 062460 104466 TRAP 66 ;
17458 062462 006000 ROR R0 ;
17459 062464 103773 BLO 1$ ;
17460 062466 000207 RTS PC ;

```

22-Oct-1980 10:47:44 TOPS
22-Oct-1980 10:45:32 PA:<

8254
8255
8253
8256
8191
8147
8259

22-Oct-1980 10:47:44
22-Oct-1980 10:45:32

TOPS-20 Bliss-16 V2(206)
PA:<NEALE>BL2ML4.BLI.2 (82)

17466 :ML4
17467 :
17468 :
17469 :
17470 :
17471 :
17472 :
17473 :
17474 :
17475 :
17476 :
17477 :
17478 :
17479 :
17480 :
17481 :
17482 :
17483 :
17484 :
17485 :
17486 :
17487 :
17488 :
17489 :
17490 :
17491 :
17492 :
17493 :
17494 :
17495 :
17496 :
17497 :
17498 :
17499 :
17500 :
17501 :
17502 :
17503 :
17504 :
17505 :
17506 :
17507 :
17508 :
17509 :
17510 :
17511 :
17512 :
17513 :
17514 :
17515 :
17516 :
17517 :
17518 :
17519 :
17520 :

8263 !
8264 !
8265 !
8266 !
8267 !
8268 !
8269 !
8270 !
8271 !
8272 !
8273 !
8274 !
8275 !
8276 !
8277 !
8278 !
8279 !
8280 !
8281 !
8282 !
8283 !
8284 !
8285 !
8286 !
8287 !
8288 !
8289 !
8290 !
8291 !
8292 !
8293 !
8294 !
8295 !
8296 !
8297 !
8298 !
8299 !
8300 !
8301 !
8302 !
8303 !
8304 !
8305 !
8306 !
8307 !
8308 !
8309 !
8310 !
8311 !
8312 !
8313 !
8314 !

! BGNTST;

! ++

! TEST NUMBER: TST 40

! TEST NAME: LAST BLOCK TRANSFER TEST

! TEST DESCRIPTION:

! TEST THE LAST BLOCK INDICATOR BIT
! FOR SETTING/NOT SETTING AND THE
! DSA REGISTER FOR INCREMENTING BY:

! 1. DOING MBUS TRANSFERS AT EACH
! BLOCK FROM BLOCK ZERO TO
! LAST BLOCK -1 AND TEST LBT
! CLEAR AND DSA REGISTER TO BE
! INCREMENTED.

! 2. DO A MBUS TRANSFER AT
! THE LAST BLOCK.
! TEST LBT TO BE SET AND
! TEST FOR CLEAR.
! TEST DSA REG TO BE INCREMENTED.

! IMPLICIT INPUTS:

! IO BUF
! A VECTOR OF 256 WORDS WHERE
! DATA FOR MBUS READS AND WRITE
! FUNCTION ARE FOUND.

! A GLOBAL OWN LOCATION TO THIS TEST.

! --

! local

! DODU_FLG;

! DROP UNIT FLAG

! DODU_FLG = ZERO;

! incr DSA_CNT from 0 to .LST_BLK - 1 do

! DO XFERRS UP TO THE LAST BLOCK

! begin

! BGNSUB;

! CLR_MBUS;

! ECC_DIS = ONE;

! MLWC = not 255;

! MLBA = IO BUF;

! MLDA = .DSA_CNT;

! MLCS1 = write;

! DISABLE ECC

! LOAD WORD COUNT

! LOAD UBUS ADRS

! LOAD DSA WITH DSA CNT

! DO A WRITE FUNCTION

22-Oct-1980 10:47:44 TOPS-20 Bliss-16 V2(206)
22-Oct-1980 10:45:32 PA:<NEALE>BL2ML4.BLI.2 (82)

```

17522 :ML4
17523 :
17524 :
17525 :      8315
17526 :      8316      do
17527 :      8317      0
17528 :      8318      until .DRY IS_SET;
17529 :      8319
17530 :      8320      if .LBT IS_SET
17531 :      8321      then
17532 :      8322      begin
17533 :      8323      ERRDF (97, ASYNC, 0);
17534 :      8324      PRINTB (THR_FMT, WRD_27, PHR_5, WRD_29);
17535 :      8325      PRINTB (FMT_7, .DSA_CNT);
17536 :      8326      DODU_FLG = ONE;
17537 :      8327      end;
17538 :      8328
17539 :      8329      if .MLDA neq .DSA_CNT + 1
17540 :      8330      then
17541 :      8331      begin
17542 :      8332      ERRDF (98, ASYNC, 0);
17543 :      8333      PRINTB (THR_FMT, REG_6, WRD_31, WRD_14);
17544 :      8334      PRINTB (FMT_7, .DSA_CNT);
17545 :      8335      DODU_FLG = ONE;
17546 :      8336      end;
17547 :      8337
17548 :      8338      ENDSUB;
17549 :      8339      end;
17550 :      8340
17551 :      8341      BGNSUB;
17552 :      8342      CLR_MBUS;
17553 :      8343      ECC_DIS = ONE;
17554 :      8344      LAST_BLK_XFER ();
17555 :      8345      MLCST = write;
17556 :      8346
17557 :      8347      do
17558 :      8348      0
17559 :      8349      until .DRY IS_SET;
17560 :      8350
17561 :      8351      if .MLDA neq .LST_BLK + 1
17562 :      8352      then
17563 :      8353      begin
17564 :      8354      ERRDF (101, ASYNC, 0);
17565 :      8355      PRINTB (THR_FMT, REG_6, WRD_31, WRD_14);
17566 :      8356      PRINTB (FMT_7, .LST_BLK);
17567 :      8357      DODU_FLG = ONE;
17568 :      8358      end;
17569 :      8359
17570 :      8360      if .LBT IS_SET
17571 :      8361      then
17572 :      8362      begin
17573 :      8363      MLDA = ONES;
17574 :      8364
17575 :      8365      if .LBT IS_SET
17576 :      8366      then

```

!DELAY UNTIL XFER TO COMPLETE

!SEE IF THE LAST BLOCK XFERR BIT SET

!ERROR AND SET DODU_FLG IF SET

!SEE IF THE DSA REG INCREMENTED

!ERROR AND SET DODU_FLG IF NOT

!DISABLE ECC

!SET UP A LAST BLOCK XFERR

!DO A WRITE FUNCTION

!DELAY UNTIL XFER TO COMPLETE

!SEE IF DSA REGISTER INCREMENTED

!ERROR AND SET DODU_FLG IF NOT

!SEE IF LBT BIT SET

!IF SET THEN TRY TO CLEAR IT

!SEE IF BIT CLEARED

22-Oct-1980 10:47:44 TOPS-20 Bliss-16 V2(206)
22-Oct-1980 10:45:32 PA:<NEALE>BL2ML4.BLI.2 (82)

```

17578 ;ML4
17579 ;
17580 ;
17581 ;      8367      begin
17582 ;      8368      ERRDF (99, ASYNC, 0);
17583 ;      8369      PRINTB (FIV_FMT, WRD_27, PHR_2, WRD_11, WRD_17, REG_6);
17584 ;      8370      DODU_FLG = ONE;
17585 ;      8371      end;
17586 ;      8372
17587 ;      8373      end
17588 ;      8374      else
17589 ;      8375      begin
17590 ;      8376      ERRDF (100, ASYNC, 0);
17591 ;      8377      PRINTB (FOR_FMT, WRD_27, PHR_1, WRD_11, WRD_27);
17592 ;      8378      DODU_FLG = ONE;
17593 ;      8379      end;
17594 ;      8380
17595 ;      8381      ENDSUB;
17596 ;      8382
17597 ;      8383      if .DODU_FLG IS_SET
17598 ;      8384      then
17599 ;      8385      begin
17600 ;      8386      DODU (.ML_LUN);
17601 ;      8387      DOCLN;
17602 ;      8388      end;
17603 ;      8389
17604 ;      8390      ENDTST;
17608 ;
17612 062470 004167 121344      $T40: JSR      R1,$SAVE4
17613 062474 005001              CLR      R1
17614 062476 016704 125632      MOV      LST.BLK,R4
17615 062502 005002              CLR      R2
17616 062504 000541              BR       6$
17617 062506 010203      1$:      MOV      R2,R3
17618 062510 005203              INC      R3
17619 062512 104402      2$:      TRAP     2
17620 062514 152777 000040 127216      BISB     #40,@ML.REG+40
17621 062522 016700 127442      MOV      ML.DUT,R0
17622 062526 042700 177770      BIC      #177770,R0
17623 062532 142777 000007 127200      BICB     #7,@ML.REG+40
17624 062540 150077 127174      BISB     R0,@ML.REG+40
17625 062544 152777 000002 127246      BISB     #2,@ML.REG+120
17626 062552 012777 177400 127130      MOV      #-400,@ML.REG+10
17627 062560 012777 010342 127132      MOV      #IO.BUF,@ML.REG+20
17628 062566 010277 127136      MOV      R2,@ML.REG+30
17629 062572 012777 000061 127100      MOV      #61,@ML.REG
17630 062600 105777 127144      3$:      TSTB     @ML.REG+50
17631 062604 100375      BPL      3$

```

8261
8304
8306
8329
8307
8308
8310
8311
8312
8313
8314
8318

```

17633          ;ML4
17634          ;
17635
17636 062606 032777 002000 127134 BIT #2000,@ML.REG+50 ;
17637 062614 001433 BEQ 4$ ;
17638 062616 104455 TRAP 55 ;
17639 062620 000141 .WORD 141 ;
17640 062622 007444 .WORD ASYNC ;
17641 062624 000000 .WORD 0 ;
17642 062626 012746 006142 MOV #WRD.29,-(SP) ;
17643 062632 012746 006646 MOV #PHR.5,-(SP) ;
17644 062636 012746 006134 MOV #WRD.27,-(SP) ;
17645 062642 012746 005366 MOV #THR.FMT,-(SP) ;
17646 062646 012746 000004 MOV #4,-(SP) ;
17647 062652 010600 MOV SP,R0 ; SP,*
17648 062654 104414 TRAP 14 ;
17649 062656 010216 MOV R2,(SP) ; DSA.CNT,*
17650 062660 012746 004520 MOV #FMT.7,-(SP) ;
17651 062664 012746 000002 MOV #2,-(SP) ;
17652 062670 010600 MOV SP,R0 ; SP,*
17653 062672 104414 TRAP 14 ;
17654 062674 012701 000001 MOV #1,R1 ; *,DODU.FLG
17655 062700 062706 000016 ADD #16,SP ;
17656 062704 027703 127020 4$: CMP @ML.REG+30,R3 ;
17657 062710 001433 BEQ 5$ ;
17658 062712 104455 TRAP 55 ;
17659 062714 000142 .WORD 142 ;
17660 062716 007444 .WORD ASYNC ;
17661 062720 000000 .WORD 0 ;
17662 062722 012746 005774 MOV #WRD.14,-(SP) ;
17663 062726 012746 006164 MOV #WRD.31,-(SP) ;
17664 062732 012746 007342 MOV #REG.6,-(SP) ;
17665 062736 012746 005366 MOV #THR.FMT,-(SP) ;
17666 062742 012746 000004 MOV #4,-(SP) ;
17667 062746 010600 MOV SP,R0 ; SP,*
17668 062750 104414 TRAP 14 ;
17669 062752 010216 MOV R2,(SP) ; DSA.CNT,*
17670 062754 012746 004520 MOV #FMT.7,-(SP) ;
17671 062760 012746 000002 MOV #2,-(SP) ;
17672 062764 010600 MOV SP,R0 ; SP,*
17673 062766 104414 TRAP 14 ;
17674 062770 012701 000001 MOV #1,R1 ; *,DODU.FLG
17675 062774 062706 000016 ADD #16,SP ;
17676 063000 104467 5$: TRAP 67 ;
17677 063002 006000 ROR R0 ;
17678 063004 103642 BLO 2$ ;
17679 063006 005202 INC R2 ; DSA.CNT
17680 063010 020204 6$: CMP R2,R4 ; DSA.CNT,*
17681 063012 002635 BLT 1$ ;
17682 063014 104402 7$: TRAP 2 ;
17683 063016 152777 000040 126714 BISB #40,@ML.REG+40 ;
17684 063024 016704 127140 MOV ML.DUT,R4 ;
17685 063030 042704 177770 BIC #177770,R4 ;
17686 063034 142777 000007 126676 BICB #7,@ML.REG+40 ;
17687 063042 150477 126672 BISB R4,@ML.REG+40 ;

```

22-Oct-1980 10:47:44 TOPS
22-Oct-1980 10:45:32 PA:<

8320

8323

8324

8325

8326

8322

8329

8332

8333

8334

8335

8331

8336

8306

8339

8341

Address	Hex	Hex	Hex	Hex	Label	Code	Comment	Address	Time	Time	Address
17689					:ML4				22-Oct-1980 10:47:44		TOPS
17690					:				22-Oct-1980 10:45:32		PA:<
17691					:						
17692	063046	152777	000002	126744		BISB	#2,@ML.REG+120				8343
17693	063054	004767	127514			JSR	PC, LAST.BLK.XFER				8344
17694	063060	012777	000061	126612		MOV	#61,@ML.REG				8345
17695	063066	105777	126656		8\$:	TSTB	@ML.REG+50				8349
17696	063072	100375				BPL	8\$				
17697	063074	016702	125234			MOV	LST.BLK,R2				8351
17698	063100	005202				INC	R2				
17699	063102	027702	126622			CMP	@ML.REG+30,R2				
17700	063106	001434				BEQ	9\$				
17701	063110	104455				TRAP	55				8354
17702	063112	000145				.WORD	145				
17703	063114	007444				.WORD	ASYNC				
17704	063116	000000				.WORD	0				
17705	063120	012746	005774			MOV	#WRD.14,-(SP)				8355
17706	063124	012746	006164			MOV	#WRD.31,-(SP)				
17707	063130	012746	007342			MOV	#REG.6,-(SP)				
17708	063134	012746	005366			MOV	#THR.FMT,-(SP)				
17709	063140	012746	000004			MOV	#4,-(SP)				
17710	063144	010600				MOV	SP,R0				; SP,*
17711	063146	104414				TRAP	14				
17712	063150	016716	125160			MOV	LST.BLK,(SP)				8356
17713	063154	012746	004520			MOV	#FMT.7,-(SP)				
17714	063160	012746	000002			MOV	#2,-(SP)				
17715	063164	010600				MOV	SP,R0				; SP,*
17716	063166	104414				TRAP	14				
17717	063170	012701	000001			MOV	#1,R1				; *,DODU.FLG
17718	063174	062706	000016			ADD	#16,SP				8353
17719	063200	032777	002000	126542	9\$:	BIT	#2000,@ML.REG+50				8360
17720	063206	001440				BEQ	10\$				
17721	063210	012777	177777	126512		MOV	#-1,@ML.REG+30				8363
17722	063216	032777	002000	126524		BIT	#2000,@ML.REG+50				8365
17723	063224	001457				BEQ	11\$				
17724	063226	104455				TRAP	55				8368
17725	063230	000143				.WORD	143				
17726	063232	007444				.WORD	ASYNC				
17727	063234	000000				.WORD	0				
17728	063236	012746	007342			MOV	#REG.6,-(SP)				8369
17729	063242	012746	006022			MOV	#WRD.17,-(SP)				
17730	063246	012746	005750			MOV	#WRD.11,-(SP)				
17731	063252	012746	006560			MOV	#PHR.2,-(SP)				
17732	063256	012746	006134			MOV	#WRD.27,-(SP)				
17733	063262	012746	005414			MOV	#FIV.FMT,-(SP)				
17734	063266	012746	000006			MOV	#6,-(SP)				
17735	063272	010600				MOV	SP,R0				; SP,*
17736	063274	104414				TRAP	14				
17737	063276	012701	000001			MOV	#1,R1				; *,DODU.FLG
17738	063302	062706	000016			ADD	#16,SP				8370
17739	063306	000426				BR	11\$				8367
17740	063310	104455			10\$:	TRAP	55				8360
17741	063312	000144				.WORD	144				8376
17742	063314	007444				.WORD	ASYNC				
17743	063316	000000				.WORD	0				

22-Oct-1980 10:47:44 TOPS
22-Oct-1980 10:45:32 PA:<

```
17745      ;ML4
17746      ;
17747
17748 063320 012746 006134      MOV      #WRD.27,-(SP)      ;
17749 063324 012746 005750      MOV      #WRD.11,-(SP)
17750 063330 012746 006542      MOV      #PHR.1,-(SP)
17751 063334 012746 006134      MOV      #WRD.27,-(SP)
17752 063340 012746 005400      MOV      #FOR.FMT,-(SP)
17753 063344 012746 000005      MOV      #5,-(SP)
17754 063350 010600      MOV      SP,R0      ; SP,*
17755 063352 104414      TRAP     14
17756 063354 012701 000001      MOV      #1,R1      ; *,DODU.FLG      8378
17757 063360 062706 000014      ADD      #14,SP      ;
17758 063364 104467      11$:    TRAP     67      ;
17759 063366 006000      ROR      R0
17760 063370 103611      BLO      7$
17761 063372 005301      DEC      R1      ; DODU.FLG      8383
17762 063374 001004      BNE      12$
17763 063376 016700 126564      MOV      ML.LUN,R0      ;
17764 063402 104451      TRAP     51      ;
17765 063404 104444      TRAP     44
17766 063406 000207      12$:    RTS      PC      ;
17767
17768      ; Routine Size: 232 words
17769      ; Maximum stack depth per invocation: 12 words
17774
17775
17779
17783 063410      T40::
17784 063410 004767 177054      1$:    JSR      PC,$T40      ;
17785 063414 104466      TRAP     66
17786 063416 006000      ROR      R0
17787 063420 103773      BLO      1$
17788 063422 000207      RTS      PC
17789
17790      ; Routine Size: 6 words
17791      ; Maximum stack depth per invocation: 0 words
17796
17797
17798 ;      8391 !<BLF/PAGE>
```


22-Oct-1980 10:47:44
 22-Oct-1980 10:45:32

TOPS-20 Bliss-16 V2(206)
 PA:<NEALE>BL2ML4.BLI.2 (83)

17800 :ML4
 17801 :
 17802 :
 17803 :
 17804 :
 17805 :
 17806 :
 17807 :
 17808 :
 17809 :
 17810 :
 17811 :
 17812 :
 17813 :
 17814 :
 17815 :
 17816 :
 17817 :
 17818 :
 17819 :
 17820 :
 17821 :
 17822 :
 17823 :
 17824 :
 17825 :
 17826 :
 17827 :
 17828 :
 17829 :
 17830 :
 17831 :
 17832 :
 17833 :
 17834 :
 17835 :
 17836 :
 17837 :
 17838 :
 17839 :
 17840 :
 17841 :
 17842 :
 17843 :
 17844 :
 17845 :
 17846 :
 17847 :
 17848 :
 17849 :
 17850 :
 17851 :
 17852 :
 17853 :
 17854 :

8392
 8393
 8394
 8395
 8396
 8397
 8398
 8399
 8400
 8401
 8402
 8403
 8404
 8405
 8406
 8407
 8408
 8409
 8410
 8411
 8412
 8413
 8414
 8415
 8416
 8417
 8418
 8419
 8420
 8421
 8422
 8423
 8424
 8425
 8426
 8427
 8428
 8429
 8430
 8431
 8432
 8433
 8434
 8435
 8436
 8437
 8438
 8439
 8440
 8441
 8442
 8443

! BGNTST;

!++

! TEST NUMBER: TST 41

! TEST NAME: INVALID ADRS TEST

! TEST DESCRIPTION:

! TEST THE DETECTION OF ILLEGAL DSA
 ! ADDRESSES BY:

- ! 1. DOING A MBUS WRITE FUNCTION
 ! AT ALL POSSIBLE ILLEGAL DSA
 ! ADDRESSES AND TEST THE
 ! IAE BIT SET.

! IMPLICIT INPUTS:

! IO_BUF
 ! A VECTOR OF 256 WORDS WHERE
 ! DATA FOR MBUS READS AND WRITE
 ! FUNCTION ARE FOUND.

! A GLOBAL OWN LOCATION TO
 ! THIS TEST.

!--

! local

! IAE_CNT;

! INVLID ADRS COUNT

! IAE_CNT = .LST_BLK;

! START AT LAST BLOCK + 1

! do

! TEST FOR ALL INVALID ADDRESSES

! begin

! IAE_CNT = .IAE_CNT + 1;

! INCREMENT IAE_CNT

! BGNSUB;

! CLR MBUS;

! MLDA = .IAE_CNT;

! LOAD DSA

! MLWC = not 255;

! LOAD WORD COUNT

! MLBA = IO_BUF;

! LOAD UBUS ADRS

! MLCS1 = write;

! DO A WRITE FUNCTION

! if .IAE IS_NOT_SET

! SET IF IAE SET

! then

! begin

! ERROR IF NOT SET

! ERRDF (102, ASYNC, 0);

! PRINTB (FIV_FMT, WRD_30, PHR_1, WRD_11, WRD_30, WRD_10);

22-Oct-1980 10:47:44 TOPS
22-Oct-1980 10:45:32 PA:<

17911 ;ML4
17912 ;
17913 ;
17914 ; Routine Size: 62 words
17915 ; Maximum stack depth per invocation: 8 words
17920
17921
17925

17929 063620
17930 063620 004767 177600
17931 063624 104466
17932 063626 006000
17933 063630 103773
17934 063632 000207

T41::
1\$: JSR PC,\$T41 ;
TRAP 66
ROR R0
BLO 1\$
RTS PC

8448

17935
17936 ; Routine Size: 6 words
17937 ; Maximum stack depth per invocation: 0 words
17942

17943
17944 ; 8451 !<BLF/PAGE>

22-Oct-1980 10:47:44
 22-Oct-1980 10:45:32

TOPS-20 Bliss-16 V2(206)
 PA:<NEALE>BL2ML4.BLI.2 (84)

17946 ;ML4

17947 ;

17948

17949 ;

17950 ;

17951 ;

17952 ;

17953 ;

17954 ;

17955 ;

17956 ;

17957 ;

17958 ;

17959 ;

17960 ;

17961 ;

17962 ;

17963 ;

17964 ;

17965 ;

17966 ;

17967 ;

17968 ;

17969 ;

17970 ;

17971 ;

17972 ;

17973 ;

17974 ;

17975 ;

17976 ;

17977 ;

17978 ;

17979 ;

17980 ;

17981 ;

17982 ;

17983 ;

17984 ;

17985 ;

17986 ;

17987 ;

17991

17995 063634

152777

000040

126076

\$T42:

BISB

#40,@ML.REG+40

;

8453

17996 063642

016700

126322

MOV

ML.DUT,R0

17997 063646

042700

177770

BIC

#177770,R0

17998 063652

142777

000007

126060

BICB

#7,@ML.REG+40

17999 063660

150077

126054

BISB

R0,@ML.REG+40

8452

!

8453

BGNTST;

8454

8455

!++

8456

TEST NUMBER: TST 42

8457

8458

TEST NAME: ADRS OVERFLOW BIT TEST

8459

8460

TEST DESCRIPTION:

8461

TEST THE DETECTION OF ADDRESS OVERFLOWS BY:

8462

8463

1. STARTING AT THE LAST BLOCK DO A TWO BLOCK TRANSFER.

8464

8465

2. READ THE AOE BIT SET.

8466

8467

IMPLICIT INPUTS:

8468

IO_BUF

8469

A VECTOR OF 256 WORDS WHERE DATA FOR MBUS READS AND WRITE

8470

FUNCTIONS ARE FOUND.

8471

!--

8472

8473

CLR MBUS;

8474

MLWC = not 511;

8475

MLBA = IO_BUF;

8476

MLDA = .LST_BLK;

8477

MLCS1 = write;

8478

8479

do

8480

0

8481

until .DRY IS_SET;

8482

8483

if .AOE IS_NOT_SET

8484

then

8485

begin

8486

ERRDF (103, SYNC, 0);

8487

PRINTB (FOR_FMT, WRD_26, PHR_1, WRD_11, FNC_19);

8488

end;

8489

8490

ENDTST;

!DO TWO BLOCK XFERR
 !LOAD UBUS ADRS
 !START AT LAST BLOCK
 !DO A WRITE FUNCTION

!DELAY UNTIL XFER TO COMPLETE

!SEE IF AOE SET

!ERROR IF NOT SET

22-Oct-1980 10:47:44 TOPS
22-Oct-1980 10:45:32 PA:<

```

18001      ;ML4
18002      ;
18003
18004 063664 012777 177000 126016      MOV      #-1000,@ML.REG+10      ;
18005 063672 012777 010342 126020      MOV      #IO.BUF,@ML.REG+20      ;
18006 063700 016777 124430 126022      MOV      LST.BLK,@ML.REG+30      ;
18007 063706 012777 000061 125764      MOV      #61,@ML.REG      ;
18008 063714 105777 126030      1$: TSTB   @ML.REG+50      ;
18009 063720 100375      BPL      1$      ;
18010 063722 032777 001000 126030      BIT      #1000,@ML.REG+60      ;
18011 063730 001024      BNE      2$      ;
18012 063732 104455      TRAP     55      ;
18013 063734 000147      .WORD   147      ;
18014 063736 007500      .WORD   SYNC      ;
18015 063740 000000      .WORD   0      ;
18016 063742 012746 007230      MOV      #FNC.19,-(SP)      ;
18017 063746 012746 005750      MOV      #WRD.11,-(SP)      ;
18018 063752 012746 006542      MOV      #PHR.1,-(SP)      ;
18019 063756 012746 006126      MOV      #WRD.26,-(SP)      ;
18020 063762 012746 005400      MOV      #FOR.FMT,-(SP)      ;
18021 063766 012746 000005      MOV      #5,-(SP)      ;
18022 063772 010600      MOV      SP,R0      ; SP,*
18023 063774 104414      TRAP     14      ;
18024 063776 062706 000014      ADD      #14,SP      ;
18025 064002 000207      2$: RTS   PC      ;
18026
18027      ; Routine Size: 52 words
18028      ; Maximum stack depth per invocation: 6 words
18033
18034
18038
18042 064004      T42::
18043 064004 004767 177624      1$: JSR   PC,$T42      ;
18044 064010 104466      TRAP     66      ;
18045 064012 006000      ROR      R0      ;
18046 064014 103773      BLO      1$      ;
18047 064016 000207      RTS      PC      ;

```

22-Oct-1980 10:47:44 TOPS-20 Bliss-16 V2(206)
22-Oct-1980 10:45:32 PA:<NEALE>BL2ML4.BLI.2 (85)

```

18053 :ML4
18054 :
18055 :
18056 :      8492 !
18057 :      8493 BGNTST;
18058 :      8494
18059 :      8495 !++
18060 :      8496 TEST NUMBER: TST 43
18061 :      8497
18062 :      8498 TEST NAME: SYNC BUS PARITY TEST
18063 :      8499
18064 :      8500 TEST DESCRIPTION:
18065 :      8501 TEST ABILITY OF SYNC BUS TO DETECT AND GENERATE
18066 :      8502 GOOD PARITY BY:
18067 :      8503
18068 :      8504 1. VIA MBUS WRITE FUNCTION WRITE ALTERNATING ONES AND ZEROES TO THE
18069 :      8505 DEVICE AND READ THE DPAR BIT CLEARED.
18070 :      8506
18071 :      8507 2. REPEAT WITH SHIFTED DATA
18072 :      8508
18073 :      8509 3. VIA MBUS READ FUNCTION READ THE ALTERNATING PATTERN AND MDPE CLEARED.
18074 :      8510
18075 :      8511 IMPLICIT INPUTS:
18076 :      8512 IO_BUF
18077 :      8513 A VECTOR OF 256 WORDS WHERE DATA FOR MBUS READS AND WRITES
18078 :      8514 FUNCTIONS ARE FOUND.
18079 :      8515 !--
18080 :      8516
18081 :      8517 if .PAR_DIS IS_SET !SEE IF PARITY IS DISABLED
18082 :      8518 then
18083 :      8519 begin !PRINT MESSAGE AND EXIT TST IF YES
18084 :      8520 PRINTB (THR_FMT, FNC_3, WRD_7, WRD_37);
18085 :      8521 EXIT_TST;
18086 :      8522 end;
18087 :      8523
18088 :      8524 CLR_MBUS;
18089 :      8525 BAI = ONE; !SET ON FIRST IO_BUF ADRS
18090 :      8526 IO_BUF = %o'125252'; !ALTERNATE 1, 0 PATTERN
18091 :      8527
18092 :      8528 incr TWICE from 0 to 1 do !REPEAT LOOP TWICE
18093 :      8529 begin
18094 :      8530 BGNSUB;
18095 :      8531 GD_BLK_XFER (); !SET UP A GOOD BLOCK XFERR
18096 :      8532 MLCS1 = write; !DO A WRITE FUNCTION
18097 :      8533
18098 :      8534 do !DELAY UNTIL XFER TO COMPLETE
18099 :      8535 0
18100 :      8536 until .DRY IS_SET;
18101 :      8537
18102 :      8538 if .DPAR IS_SET !SEE IF DPAR GOT SET
18103 :      8539 then
18104 :      8540 begin !ERROR IF SET
18105 :      8541 ERRDF (105, SYNC, 0);
18106 :      8542 PRINTB (FOR_FMT, WRD_23, WRD_6, WRD_7, WRD_9);
18107 :      8543 end;

```

22-Oct-1980 10:47:44 TOPS-20 Bliss-16 V2(206)
22-Oct-1980 10:45:32 PA:<NEALE>BL2ML4.BLI.2 (85)

```

18109 ;ML4
18110 ;
18111 ;
18112 ;      8544
18113 ;      8545      IO_BUF = .IO_BUF^ONE;
18114 ;      8546      MLER = ZEROES;
18115 ;      8547      ENDSUB;
18116 ;      8548      end;
18117 ;      8549
18118 ;      8550      incr TWICE from 0 to 1 do
18119 ;      8551      begin
18120 ;      8552      BGNSUB;
18121 ;      8553      CLR_MBUS;
18122 ;      8554      BAI = ONE;
18123 ;      8555      GD_BLK_XFER ();
18124 ;      8556      MLCS1 = read;
18125 ;      8557
18126 ;      8558      do
18127 ;      8559      0
18128 ;      8560      until .DRY IS_SET;
18129 ;      8561
18130 ;      8562      if .MDPE IS_SET
18131 ;      8563      then
18132 ;      8564      begin
18133 ;      8565      ERRDF (106, SYNC, 0);
18134 ;      8566      PRINTB (FOR_FMT, WRD_23, WRD_6, WRD_7, WRD_8);
18135 ;      8567      end;
18136 ;      8568
18137 ;      8569      ENDSUB;
18138 ;      8570      IO_BUF = .IO_BUF^ONE;
18139 ;      8571      GD_BLK_XFER ();
18140 ;      8572      MLCS1 = write;
18141 ;      8573
18142 ;      8574      do
18143 ;      8575      0
18144 ;      8576      until .DRY IS_SET;
18145 ;      8577
18146 ;      8578      end;
18147 ;      8579
18148 ;      8580      ENDTST;

```

```

!SHIFT THE IO_BUF & REPEAT
!CLEAR OUT ERROR REG & REPEAT

!REPEATE LOOP TWICE

!SET ON FIRST IO_BUF ADRS
!SET UP A GOOD BLOCK XFERR
!DO A READ XFERR

!DELAY UNTIL XFER TO COMPLETE

!SEE IF READ GENERATED BAD PARITY
!ERROR IF MDPE SET

!MAKE DATA PATTERN HAVE ONE LESS ONE
!SET UP A GOOD BLK XFER
!CHANGE THE BACKGROUND IN MEMORY

!DELAY UNTIL XFER TO COMPLETE

```

18156	064020	010146		\$T43:	MOV	R1, -(SP)	:	8490
18157	064022	026727	124302	000001	CMP	PAR.DIS, #1	:	8517
18158	064030	001021			BNE	1\$:	
18159	064032	012746	006232		MOV	#WRD.37, -(SP)	:	8520
18160	064036	012746	005676		MOV	#WRD.7, -(SP)		
18161	064042	012746	006774		MOV	#FNC.3, -(SP)		
18162	064046	012746	005366		MOV	#THR.FMT, -(SP)		

18220				:ML4					22-Oct-1980 10:47:44	TOPS
18221				:					22-Oct-1980 10:45:32	PA:<
18222										
18223	064346	105777	125376	6\$:	TSTB	@ML.REG+50	:			8560
18224	064352	100375			BPL	6\$:			
18225	064354	032777	000400	125356	BIT	#400,@ML.REG+40	:			8562
18226	064362	001424			BEQ	7\$:			
18227	064364	104455			TRAP	55	:			8565
18228	064366	000152			.WORD	152				
18229	064370	007500			.WORD	SYNC				
18230	064372	000000			.WORD	0				
18231	064374	012746	005712		MOV	#WRD.8,-(SP)	:			8566
18232	064400	012746	005676		MOV	#WRD.7,-(SP)				
18233	064404	012746	005670		MOV	#WRD.6,-(SP)				
18234	064410	012746	006076		MOV	#WRD.23,-(SP)				
18235	064414	012746	005400		MOV	#FOR.FMT,-(SP)				
18236	064420	012746	000005		MOV	#5,-(SP)				
18237	064424	010600			MOV	SP,RO	:	SP,*		
18238	064426	104414			TRAP	14				
18239	064430	062706	000014		ADD	#14,SP	:			8564
18240	064434	104467		7\$:	TRAP	67	:			8567
18241	064436	006000			ROR	RO				
18242	064440	103715			BLO	5\$				
18243	064442	006367	123674		ASL	IO.BUF	:			8570
18244	064446	004767	126070		JSR	PC,GD.BLK.XFER	:			8571
18245	064452	012777	000061	125220	MOV	#61,@ML.REG	:			8572
18246	064460	105777	125264		MOV	@ML.REG+50	:			8576
18247	064464	100375		8\$:	BPL	8\$:			
18248	064466	005201			INC	R1	:	TWICE		8550
18249	064470	020127	000001		CMP	R1,#1	:	TWICE,*		
18250	064474	003677			BLE	5\$				
18251	064476	012601		9\$:	MOV	(SP)+,R1	:			8490
18252	064500	000207			RTS	PC				
18257										
18258										
18262										
18266	064502			T43::						
18267	064502	004767	177312	1\$:	JSR	PC,\$T43	:			8578
18268	064506	104466			TRAP	66				
18269	064510	006000			ROR	RO				
18270	064512	103773			BLO	1\$				
18271	064514	000207			RTS	PC				

22-Oct-1980 10:47:44
 22-Oct-1980 10:45:32

TOPS-20 Bliss-16 V2(206)
 PA:<NEALE>BL2ML4.BLI.2 (86)

```

18277 :ML4
18278 :
18279 :
18280 :      8582 !
18281 :      8583 !
18282 :      8584 ! BGNTST;
18283 :      8585 !
18284 :      8586 ! ++
18285 :      8587 ! TEST NUMBER: TST 44
18286 :      8588 !
18287 :      8589 ! TEST NAME: WRITE READ ML11
18288 :      8590 !
18289 :      8591 ! TEST DESCRIPTION:
18290 :      8592 !
18291 :      8593 ! PROVIDES A MBUS READ/WRITE
18292 :      8594 ! DATA TRANSFER TROUBLE SHOOTING
18293 :      8595 ! LOOP BY:
18294 :      8596 !
18295 :      8597 ! 1. LOAD APPROPRIATE RH REGISTERS.
18296 :      8598 ! DO A WRITE FUNCTION.
18297 :      8599 !
18298 :      8600 ! 2. LOAD APPROPRIATE RH REGISTERS
18299 :      8601 ! DO A WRITE CHECK FUNCTION.
18300 :      8602 !
18301 :      8603 ! 3. COMPIMENT DATA AND
18302 :      8604 ! REPEAT.
18303 :      8605 !
18304 :      8606 ! IMPLICIT INPUTS:
18305 :      8607 !
18306 :      8608 ! IO_BUF
18307 :      8609 ! A VECTOR OF 256 WORDS WHERE
18308 :      8610 ! DATA FOR MBUS READ AND WRITE
18309 :      8611 ! FUNCTIONS IS FOUND.
18310 :      8612 !
18311 :      8613 ! A GLOBAL OWN LOCATION TO THIS
18312 :      8614 ! TEST.
18313 :      8615 !
18314 :      8616 !
18315 :      8617 ! --
18316 :      8618 !
18317 :      8619 ! local
18318 :      8620 ! TST_PAT;
18319 :      8621 !
18320 :      8622 ! TST_PAT = ONES;
18321 :      8623 !
18322 :      8624 ! incr TWICE from 0 to 1 do
18323 :      8625 ! begin
18324 :      8626 ! CLR_MBUS;
18325 :      8627 ! BAI = ONE;
18326 :      8628 ! IO_BUF = TST_PAT;
18327 :      8629 ! GD_BLK_XFER (T);
18328 :      8630 ! MLC$1 = write;
18329 :      8631 !
18330 :      8632 ! do
18331 :      8633 ! 0
  
```

```

!TEST PATTERN

!WRITE READ 1'S AND 0'S ON MBUS

!SET ON FIRST IO_BUF ADRS
!LOAD FIRST IO_BUF ADRS
!SET UP A GOOD_BLOCK XFERR
!DO A WRITE FUNCTION

!DELAY UNTIL XFER TO COMPLETE
  
```

22-Oct-1980 10:47:44 TOPS-20 Bliss-16 V2(206)
22-Oct-1980 10:45:32 PA:<NEALE>BL2ML4.BLI.2 (86)

```

18333 :ML4
18334 :
18335 :
18336 :      8634      until .DRY IS_SET;
18337 :      8635
18338 :      8636      CLR_MBUS;
18339 :      8637      BAI = ONE;
18340 :      8638      GD_BLK_XFER ();
18341 :      8639      MLCS1 = WRT_CHK;
18342 :      8640
18343 :      8641      do
18344 :      8642          0
18345 :      8643      until .DRY IS_SET;
18346 :      8644
18347 :      8645      if .WCE IS_SET
18348 :      8646      then
18349 :      8647          begin
18350 :      8648              ERRDF (109, TRBLE_LOOP, 0);
18351 :      8649              PRINTB (SIX_FMT, FNC_4, WRD_10, WRD_12, FNC_5, FNC_6, FNC_3);
18352 :      8650              end;
18353 :      8651
18354 :      8652      TST_PAT = not .TST_PAT;
18355 :      8653      end;
18356 :      8654
18357 :      8655      ENDTST;

```

```

!SET ON FIRST IO BUF ADRS
!SET UP A GOOD BLOCK XFERR
!DO A WRITE CHECK FUNCTION

!DELAY UNTIL XFER TO COMPLETE

!SEE IF WRITE CHECK ERROR SET
!ERROR IF SET

!COMPLIMENT TST_PAT AND REPEAT

```

```

18365 064516 004167 117264      $T44: JSR      R1,$SAVE2      :
18366 064522 012701 177777      MOV      #-1,R1      : *,TST.PAT
18367 064526 005002      CLR      R2      : TWICE
18368 064530 152777 000040 125202 1$: BISB    #40,@ML.REG+40 :
18369 064536 016700 125426      MOV      ML.DUT,R0 :
18370 064542 042700 177770      BIC      #177770,R0 :
18371 064546 142777 000007 125164 B!CB    #7,@ML.REG+40 :
18372 064554 150077 125160      BISB    R0,@ML.REG+40 :
18373 064560 152777 000010 125152 BISB    #10,@ML.REG+40 :
18374 064566 010167 123550      MOV      R1,IO.BUF : TST.PAT,*
18375 064572 004767 125744      JSR      PC,GD.BLK.XFER :
18376 064576 012777 000061 125074 MOV      #61,@ML.REG :
18377 064604 105777 125140      2$: TSTB  @ML.REG+50 :
18378 064610 100375      BPL      2$ :
18379 064612 152777 000040 125120 BISB    #40,@ML.REG+40 :
18380 064620 016700 125344      MOV      ML.DUT,R0 :
18381 064624 042700 177770      BIC      #177770,R0 :
18382 064630 142777 000007 125102 B!CB    #7,@ML.REG+40 :
18383 064636 150077 125076      BISB    R0,@ML.REG+40 :
18384 064642 152777 00001C 125070 BISB    #10,@ML.REG+40 :
18385 064650 004767 125666      JSR      PC,GD.BLK.XFER :
18386 064654 012777 000051 125016 MOV      #51,@ML.REG :

```

```

8580
8622
8624
8625

8627
8628
8629
8630
8634

8637
8638
8639

```

22-Oct-1980 10:47:44 TOPS
 22-Oct-1980 10:45:32 PA:<

```

18388      ;ML4
18389      ;
18390
18391 064662 105777 125062      3$:  TSTB  @ML.REG+50      ;
18392 064666 100375           BPL  3$      ;
18393 064670 032777 040000 125042 BIT  #40000,@ML.REG+40 ;
18394 064676 001430           BEQ  4$      ;
18395 064700 104455           TRAP 55      ;
18396 064702 000155           .WORD 155      ;
18397 064704 007662           .WORD TRBLE.LOOP ;
18398 064706 000000           .WORD 0      ;
18399 064710 012746 006774      MOV  #FNC.3,-(SP) ;
18400 064714 012746 007030      MOV  #FNC.6,-(SP) ;
18401 064720 012746 007020      MOV  #FNC.5,-(SP) ;
18402 064724 012746 005760      MOV  #WRD.12,-(SP) ;
18403 064730 012746 005740      MOV  #WRD.10,-(SP) ;
18404 064734 012746 007002      MOV  #FNC.4,-(SP) ;
18405 064740 012746 005432      MOV  #SIX.FMT,-(SP) ;
18406 064744 012746 000007      MOV  #7,-(SP) ;
18407 064750 010600           MOV  SP,R0      ; SP,*
18408 064752 104414           TRAP 14      ;
18409 064754 062706 000020      ADD  #20,SP      ;
18410 064760 005101           4$:  COM  R1      ; TST.PAT
18411 064762 005202           INC  R2      ; TWICE
18412 064764 020227 000001      CMP  R2,#1      ; TWICE,*
18413 064770 003657           BLE  1$      ;
18414 064772 000207           RTS  PC      ;
18415
18416      ; Routine Size: 87 words
18417      ; Maximum stack depth per invocation: 11 words
18422
18423
18427
18431 064774           T44::
18432 064774 004767 177516      1$:  JSR  PC,$T44 ;
18433 065000 104466           TRAP 66      ;
18434 065002 006000           ROR  R0      ;
18435 065004 103773           BLO  1$      ;
18436 065006 000207           RTS  PC      ;
  
```

8643

8645

8648

8649

8647

8652

8624

8580

8653

22-Oct-1980 10:47:44
 22-Oct-1980 10:45:32

TOPS-20 Bliss-16 V2(206)
 PA:<NEALE>BL2ML4.BLI.2 (87)

```

18442 :ML4
18443 :
18444 :
18445 :      8657  !
18446 :      8658  !
18447 :      8659  ! BGNTST;
18448 :      8660  !
18449 :      8661  ! ++
18450 :      8662  ! TEST NUMBER: TST 45
18451 :      8663  !
18452 :      8664  ! TEST NAME: PROM DATA TEST
18453 :      8665  !
18454 :      8666  ! TEST DESCRIPTION:
18455 :      8667  !
18456 :      8668  !     VERIFY THAT CHECK SUM VALUES FOR
18457 :      8669  !     ALL PRESENT MEMORY ARRAY UV PROM
18458 :      8670  !     LOCATIONS ARE ERROR FREE BY:
18459 :      8671  !
18460 :      8672  !     1. DOING MBUS TRANSFERS
18461 :      8673  !         AT ALL PRESENT BLOCKS
18462 :      8674  !         AND TESTING THE UNS BIT
18463 :      8675  !         TO BE CLEARED.
18464 :      8676  !
18465 :      8677  ! IMPLICIT INPUTS:
18466 :      8678  !
18467 :      8679  !     IO_BUF
18468 :      8680  !
18469 :      8681  !     A VECTOR OF 256 WORDS WHERE
18470 :      8682  !     DATA FOR MBUS READ AND WRITE
18471 :      8683  !     FUNCTIONS IS FOUND.
18472 :      8684  !
18473 :      8685  !
18474 :      8686  !     CHIP_SIZ
18475 :      8687  !     INDICATED THE SIZE OF THE
18476 :      8688  !     ARRAY MODULES MOS RAMS.
18477 :      8689  !
18478 :      8690  !
18479 :      8691  ! --
18480 :      8692  !
18481 :      8693  ! incr DSA_CNT from 0 to .LST_BLK do
18482 :      8694  !   begin
18483 :      8695  !   BGNSUB;
18484 :      8696  !   CLR_MBUS;
18485 :      8697  !   ECC_DIS = ONE;
18486 :      8698  !   MLWC = not 255;
18487 :      8699  !   MLBA = IO_BUF;
18488 :      8700  !   MLDA = .DSA_CNT;
18489 :      8701  !   MLCS1 = write;
18490 :      8702  !
18491 :      8703  !   do
18492 :      8704  !     0
18493 :      8705  !   until .DRY IS_SET;
18494 :      8706  !
18495 :      8707  !   if .UNS IS_SET
18496 :      8708  !   then
  
```

!WRITE TO ALL PRESENT BLK'S AND CHECK UNS BIT

!DISABLE ECC
 !LOAD WORD COUNT
 !LOAD UBUS ADRS
 !LOAD DSA
 !DO A WRITE FUNCTION

!DELAY UNTIL XFER TO COMPLETE

!SEE IF XFERR CAUSED AN UNS ERROR

18498
18499 : 8709
18500 : 8710
18501 : 8711
18502 : 8712
18503 : 8713
18504 : 8714
18505 : 8715
18506 : 8716
18507 : 8717
18508 : 8718
18509 : 8719
18510 : 8720
18511 : 8721
18512 : 8722
18513 : 8723
18514 : 8724
18515 : 8725
18516 : 8726
18517 : 8727
18518 : 8728
18519 : 8729
18523

```
begin
ERRDF (107, ARR_DAT, 0);
PRINTB (TWO_FMT, WRD_35, PHR_4);
PRINTB (FMT_9, .DSA_CNT);

if .CHIP_SIZ eql 64
then
begin
PRINTB (FMT_8, ((.DSA_CNT<11, 4>) + 1)); !64K MOS RAM
end
else
begin
PRINTB (FMT_8, ((.DSA_CNT<9, 4>) + 1)); !16K MOS RAM
end
end;

ENDSUB;
end;

ENDTST;
```

18527 065010 004167 116772
18528 065014 016702 123314
18529 065020 005001
18530 065022 000536
18531 065024 104402
18532 065026 152777 000040 124704
18533 065034 016700 125130
18534 065040 042700 177770
18535 065044 142777 000007 124666
18536 065052 150077 124662
18537 065056 152777 000002 124734
18538 065064 012777 177400 124616
18539 065072 012777 010342 124620
18540 065100 010177 124624
18541 065104 012777 000061 124566
18542 065112 105777 124632
18543 065116 100375
18544 065120 032777 040000 124632
18545 065126 001470
18546 065130 104455
18547 065132 000153
18548 065134 007534
18549 065136 000000
18550
18551 065140 012746 006630
18552 065144 012746 006220
18553 065150 012746 005356
18554 065154 012746 000003
18555 065160 010600
18556 065162 104414
18557 065164 010116
18558 065166 012746 004602
18559 065172 012746 000002
18560 065176 010600

```
$T45: JSR R1,$SAVE2
MOV LST.BLK,R2
CLR R1
BR 6$
1$: TRAP 2
BISB #40,@ML.REG+40
MOV ML.DUT,R0
BIC #177770,R0
BICB #7,@ML.REG+40
BISB R0,@ML.REG+40
BISB #2,@ML.REG+120
MOV #-400,@ML.REG+10
MOV #10.BUF,@ML.REG+20
MOV R1,@ML.REG+30
MOV #61,@ML.REG
2$: TSTB @ML.REG+50
BPL 2$
BIT #40000,@ML.REG+60
BEQ 5$
TRAP 5$
.WORD 153
.WORD ARR.DAT
.WORD 0

MOV #PHR.4,-(SP)
MOV #WRD.35,-(SP)
MOV #TWO.FMT,-(SP)
MOV #3,-(SP)
MOV SP,R0
TRAP 14
MOV R1,(SP)
MOV #FMT.9,-(SP)
MOV #2,-(SP)
MOV SP,R0
```

8655
8693
8694
8695
8697
8698
8699
8700
8701
8705
8707
8710
8711
8712

18561	065200	104414			TRAP	14			
18562	065202	026727	123124	000100	CMP	CHIP.SIZ,#100	:		8714
18563	065210	001020			BNE	3\$			
18564	065212	010100			MOV	R1,R0	:	DSA.CNT,*	8717
18565	065214	006200			ASR	R0			
18566	065216	006200			ASR	R0			
18567	065220	006200			ASR	R0			
18568	065222	000300			SWAB	R0			
18569	065224	042700	177760		BIC	#177760,R0			
18570	065230	010046			MOV	R0,-(SP)			
18571	065232	005216			INC	(SP)			
18572	065234	012746	004546		MOV	#FMT.8,-(SP)			
18573	065240	012746	000002		MOV	#2,-(SP)			
18574	065244	010600			MOV	SP,R0	:	SP,*	
18575	065246	104414			TRAP	14			
18576	065250	000415			BR	4\$:		8714
18577	065252	010100		3\$:	MOV	R1,R0	:	DSA.CNT,*	8721
18578	065254	006200			ASR	R0			
18579	065256	000300			SWAB	R0			
18580	065260	042700	177760		BIC	#177760,R0			
18581	065264	010046			MOV	R0,-(SP)			
18582	065266	005216			INC	(SP)			
18583	065270	012746	004546		MOV	#FMT.8,-(SP)			
18584	065274	012746	000002		MOV	#2,-(SP)			
18585	065300	010600			MOV	SP,R0	:	SP,*	
18586	065302	104414			TRAP	14			
18587	065304	062706	000022	4\$:	ADD	#22,SP	:		8709
18588	065310	104467		5\$:	TRAP	67	:		8724
18589	065312	006000			ROR	R0			
18590	065314	103643			BLO	1\$			
18591	065316	005201			INC	R1	:	DSA.CNT	8693
18592	065320	020102		6\$:	CMP	R1,R2	:	DSA.CNT,*	
18593	065322	003640			BLE	1\$			
18594	065324	000207			RTS	PC	:		8655
18595									
18596									
18597									
18608									
18612	065326								
18613	065326	004767	177456		T45::				
18614	065332	104466		1\$:	JSR	PC,\$T45	:		8727
18615	065334	006000			TRAP	66			
18616	065336	103773			ROR	R0			
18617	065340	000207			BLO	1\$			
					RTS	PC			

; Routine Size: 103 words
 ; Maximum stack depth per invocation: 12 words

```

18623 ;ML4
18624 ;
18625 ;
18626 ; 8731 ;
18627 ; 8732 ;
18628 ; 8733 BGNCLN;
18629 ; 8734 CLR = ONE;
18630 ; 8735 return;
18631 ; 8736 ENDCLN;
18635 ;
18639 065342 152777 000040 124370 LCLEAN: BISB #40,@ML.REG+40 ;
18640 065350 000207 RTS PC ;
18641 ;
18642 ; Routine Size: 4 words
18643 ; Maximum stack depth per invocation: 0 words
18648 ;
18649 ;
18653 ;
18657 065352 L$CLEAN::
18658 065352 004767 177764 JSR PC,L$CLEAN ;
18659 065356 104412 TRAP 12 ;
18660 065360 000207 RTS PC ;
18661 ;
18662 ; Routine Size: 4 words
18663 ; Maximum stack depth per invocation: 0 words
18668 ;
18669 ;
18670 ; 8737 LASTAD;
18671 ; 8738 BGNSETUP (0);
18672 ; 8739 ENDSETUP;

```

22-Oct-1980 10:47:44
22-Oct-1980 10:45:32

TOPS-20 Bliss-16 V2(206)
PA:<NEALE>BL2ML4.BLI.2 (88)

8734
8729

8735

22-Oct-1980 10:47:44 TOPS
22-Oct-1980 10:45:32 PA:<

```
18677      :ML4
18678      ;
18679
18680
18681 065362 065366      BL$LAS::.WORD  T$FREE
18682 065364 000000      .WORD  <<T$FREE-<BL$LAS+4>>/2>
18683 065366 000000      T$FREE::.WORD  0
18684
18685
18686      065366      L$LAST==          BL$LAS+4
18687      000000      T$PTHV==          0
18688
18689
18693 065370
18694 065370 000207      $END.LINK::
18695      RTS      PC      ;
18696
18697      ; Routine Size: 1 word
18702      ; Maximum stack depth per invocation: 0 words
18703
18704 :      8740 end
18705 :      8741
18706 :      8742 eludom
18710
18711      : OTS external references
18712      .GLOBL BL$GT2, $SAVE5, $SAVE4, $SAVE3
18713      .GLOBL $SAVE2, BL$PU2, BL$GT1
18714
18715
18716
18717
18718
18719
18720      : Size:          11069 code + 1542 data words
18721      : Run Time:       02:15.5
18722      : Elapsed Time:   04:38.4
18723      : Memory Used:    103 pages
18724      : Compilation Complete
18725
18726      000001      .END
```

8736

SYMBOL TABLE

ADR = 000020 G	C\$DU = 000053	EVL = 000004 G	F\$MOD = 000000	I\$SRV = 000041
ARR.DA= 007534	C\$EDIT= 000003	E\$END = 002100	F\$MSG = 000011	I\$SUB = 000041
ARR.IN 010324	C\$ERDF= 000055	E\$LOAD= 000035	F\$PROT= 000021	I\$TST = 000041
ARR.16 010336	C\$ERHR= 000056	E2.TEM= 007720	F\$PWR = 000017	J\$JMP = 000167
ASSEMB= 000010	C\$ERRO= 000060	FIRST. 012520	F\$RPT = 000012	LAST.B 012574
ASYNCR = 007444	C\$ERSF= 000054	FIV.FM= 005414	F\$SEG = 000003	LAU 004152
BIT0 = 000001 G	C\$ERSO= 000057	FMT.1 = 004164	F\$SOFT= 000005	LAUTO 004126
BIT00 = 000001 G	C\$ESCA= 000010	FMT.10= 004634	F\$SRV = 000010	LCLEAN 065342
BIT01 = 000002 G	C\$ESEGE= 000005	FMT.11= 004706	F\$SUB = 000002	LDU 004140
BIT02 = 000004 G	C\$ESUB= 000003	FMT.12= 004742	F\$SW = 000014	LD.LNG 013636
BIT03 = 000010 G	C\$ETST= 000001	FMT.13= 004772	F\$TEST= 000001	LINIT 020600
BIT04 = 000020 G	C\$EXIT= 000032	FMT.14= 005046	GD.BLK 012542	LOAD.S 012172
BIT05 = 000040 G	C\$GETB= 000026	FMT.15= 005106	GOOD.B 010326	LOE = 040000 G
BIT06 = 000100 G	C\$GETW= 000027	FMT.16= 005154	G\$CNT0= 000200	LOT = 000010 G
BIT07 = 000200 G	C\$GMAN= 000043	FMT.17= 005244	G\$DELM= 000372	LRPT 004114
BIT08 = 000400 G	C\$GPHR= 000042	FMT.18= 005300	G\$DISP= 000003	LST.AR 010340
BIT09 = 001000 G	C\$GPLO= 000030	FMT.2 = 004224	G\$EXCP= 000400	LST.BL 010334
BIT1 = 000002 G	C\$GPRI= 000040	FMT.3 = 004312	G\$HILI= 000002	LSACP 002110 G
BIT10 = 002000 G	C\$INIT= 000011	FMT.4 = 004336	G\$LOLI= 000001	LSAPT 002036 G
BIT11 = 004000 G	C\$INLP= 000020	FMT.5 = 004366	G\$NO = 000000	LSAU 004154 G
BIT12 = 010000 G	C\$MANI= 000050	FMT.6 = 004470	G\$OFFS= 000400	LSAUT 002070 G
BIT13 = 020000 G	C\$MEM = 000031	FMT.7 = 004520	G\$OFISI= 000376	LSAUTO 004130 G
BIT14 = 040000 G	C\$MSG = 000023	FMT.8 = 004546	G\$PRMA= 000001	L\$CCP 002106 G
BIT15 = 100000 G	C\$OPEN= 000034	FMT.9 = 004602	G\$PRMD= 000002	L\$CLEA 065352 G
BIT2 = 000004 G	C\$PNTB= 000014	FNC.1 = 006752	G\$PRML= 000000	L\$CO 002032 G
BIT3 = 000010 G	C\$PNTF= 000017	FNC.10= 007074	G\$RADA= 000140	L\$DEPO 002011 G
BIT4 = 000020 G	C\$PNTS= 000016	FNC.11= 007104	G\$RADB= 000000	L\$DESC 002130 G
BIT5 = 000040 G	C\$PNTX= 000015	FNC.12= 007124	G\$RADD= 000040	L\$DESP 002076 G
BIT6 = 000100 G	C\$QIO = 000377	FNC.13= 007136	G\$RADL= 000120	L\$DEVP 002060 G
BIT7 = 000200 G	C\$RDBU= 000007	FNC.14= 007146	G\$RADO= 000020	L\$DISP 002164 G
BIT8 = 000400 G	C\$REFG= 000047	FNC.15= 007162	G\$XFER= 000004	L\$DLY 002116 G
BIT9 = 001000 G	C\$RESE= 000033	FNC.16= 007174	G\$YES = 000010	L\$DTP 002040 G
BL\$DIV 003714 G	C\$REVI= 000003	FNC.17= 007206	HELP = 000000	L\$DTP 002034 G
BL\$G11 002760 G	C\$RFLA= 000021	FNC.18= 007216	HOE = 100000 G	L\$DU 004142 G
BL\$GT2 003102 G	C\$RPT = 000025	FNC.19= 007230	HW.OR. 007722	L\$DUT 002072 G
BL\$LAS 065362 G	C\$SEFG= 000046	FNC.2 = 006766	IBE = 010000 G	L\$DVTY 002122 G
BL\$MOD 003726 G	C\$SPRI= 000041	FNC.21= 007242	IDU = 000040 G	L\$EF 002052 G
BL\$MUL 003470 G	C\$SVEC= 000037	FNC.22= 007254	IER = 020000 G	L\$ENVI 002044 G
BL\$PU1 003244 G	C\$TPRI= 000013	FNC.23= 007266	INTER = 007622	L\$ERRT 002152 G
BL\$PU2 003340 G	DAT.DM 012626	FNC.3 = 006774	IO.BUF 010342	L\$ETP 002102 G
BL\$SHF 003740 G	DFPTBL 002320 G	FNC.4 = 007002	ISR = 000100 G	L\$EXP1 002046 G
BOE = 000400 G	DIAGMC= 000000	FNC.5 = 007020	IXE = 004000 G	L\$EXP4 002064 G
CHIP.S 010332	DIVMOD 003532	FNC.6 = 007030	ISAU = 000041	L\$EXP5 002066 G
C\$AU = 000052	DRIVE. 011674	FNC.7 = 007036	ISAUTO= 000041	L\$SHARD 002344 G
C\$AUTO= 000061	D1.TEM= 007714	FNC.8 = 007046	ISCLN = 000041	L\$HIME 002120 G
C\$BRK = 000022	D2.TEM= 007716	FNC.9 = 007062	ISDU = 000041	L\$HPCP 002016 G
C\$BSEG= 000004	EF.CON= 000036 G	FOR.FM= 005400	ISHRD = 000041	L\$HPTP 002022 G
C\$BSUB= 000002	EF.NEW= 000035 G	F\$AU = 000015	\$INIT= 000041	L\$HW 002320 G
C\$CEFG= 000045	EF.PWR= 000034 G	F\$AUTO= 000020	\$MOD = 000041	L\$ICP 002104 G
C\$CLCK= 000062	EF.RES= 000037 G	F\$BGN = 000040	\$MSG = 000041	L\$INIT 021512 G
C\$CLEA= 000012	EF.STA= 000040 G	F\$CLEA= 000007	\$PROT= 000040	L\$LADP 002026 G
C\$CLOS= 000035	EIG.FM= 005474	F\$DU = 000016	\$PTAB= 000041	L\$LAST= 065366 G
C\$CLP1= 000006	ELV.FM= 005576	F\$END = 000041	\$PWR = 000041	L\$LOAD 002100 G
C\$CVEC= 000036	ERRBLK 002160 G	F\$HARD= 000004	\$RPT = 000041	L\$LUN 002074 G
C\$DCLN= 000044	ERRMSG 002156 G	F\$HW = 000013	\$SEG = 000041	L\$MREV 002050 G
C\$DODU= 000051	ERRNBR 002154 G	F\$INIT= 000006	\$SETU= 000041	L\$NAME 002000 G
C\$DRPT= 000024	ERRTYP 002152 G	F\$JMP = 000050	\$SFT = 000041	L\$PRIO 002042 G

SYMBOL TABLE

LSPROT	002712	G	PRI	=	002000	G	P.ABU	006034	P.ADZ	007020	REG.IN	011676		
LSPRT	002112	G	PRI00	=	000000	G	P.ABV	006040	P.AEA	007030	REG.1	=	007302	
LSREPP	002062	G	PRI01	=	000040	G	P.ABW	006046	P.AEB	007036	REG.10	=	007372	
LSREV	002010	G	PRI02	=	000100	G	P.ABX	006054	P.AEC	007046	REG.11	=	007400	
LSRPT	004116	G	PRI03	=	000140	G	P.ABY	006062	P.AED	007062	REG.12	=	007406	
LSSOFT	002662	G	PRI04	=	000200	G	P.ABZ	006076	P.AEE	007074	REG.13	=	007414	
LSSPC	002056	G	PRI05	=	000240	G	P.ACA	006104	P.AEF	007104	REG.14	=	007422	
LSSPCP	002020	G	PRI06	=	000300	G	P.ACB	006112	P.AEG	007124	REG.15	=	007430	
LSSPTP	002024	G	PRI07	=	000340	G	P.ACC	006126	P.AEH	007136	REG.16	=	007436	
LSSTA	002030	G	PRSN	=	002340	G	P.ACD	006134	P.AEI	007146	REG.2	=	007312	
LSSW	002340	G	PTBL.P	=	010320		P.ACE	006142	P.AEJ	007162	REG.3	=	007320	
LSTEST	002114	G	P.AAA	=	004164		P.ACF	006156	P.AEK	007174	REG.4	=	007326	
LSTIML	002014	G	P.AAB	=	004224		P.ACG	006164	P.AEL	007206	REG.5	=	007334	
LSUNIT	002012	G	P.AAC	=	004312		P.ACH	006200	P.AEM	007216	REG.6	=	007342	
L10000	002336		P.AAD	=	004336		P.ACI	006206	P.AEN	007230	REG.7	=	007350	
L10001	002342		P.AAE	=	004366		P.ACJ	006212	P.AEO	007242	REG.8	=	007356	
L10002	002440		P.AAF	=	004470		P.ACK	006220	P.AEP	007254	REG.9	=	007364	
L10003	002670		P.AAG	=	004520		P.ACL	006226	P.AEQ	007266	RE2	=	004106	
MEM.AR=	007566		P.AAH	=	004546		P.ACM	006232	P.AER	007302	RE3	=	004104	
ML.DUT	012170	G	P.AAI	=	004602		P.ACN	006242	P.AES	007312	RE4	=	004102	
ML.LUN	012166	G	P.AAJ	=	004634		P.ACO	006250	P.AET	007320	RH.ADD	=	012160	G
ML.REG	011700	G	P.AAK	=	004706		P.ACP	006260	P.AEU	007326	RH.TYP	=	012162	G
MSGH1	002440		P.AAL	=	004742		P.ACQ	006264	P.AEV	007334	RH.VEC	=	012164	G
MSGH2	002454		P.AAM	=	004772		P.ACR	006300	P.AEW	007342	SEV.FM=	=	005452	
MSGH3	002502		P.AAN	=	005046		P.ACS	006306	P.AEX	007350	SFPTBL	=	002340	G
MSGH4	002525		P.AAO	=	005106		P.ACT	006316	P.AEY	007356	SIX.FM=	=	005432	
MSGH5	002556		P.AAP	=	005154		P.ACU	006324	P.AEZ	007364	STACK	=	011354	
MSGH6	002610		P.AAQ	=	005244		P.ACV	006336	P.AFA	007372	STK.OF	=	011342	
MSGH7	002634		P.AAR	=	005300		P.ACW	006350	P.AFB	007400	SVCGBL=	=	177777	
MSGS1	002670		P.AAS	=	005350		P.ACX	006360	P.AFC	007406	SVCINS=	=	177777	
NIB.SA	007714		P.AAT	=	005356		P.ACY	006370	P.AFD	007414	SVCSUB=	=	177777	
NIN.FM=	005520		P.AAU	=	005366		P.ACZ	006400	P.AFE	007422	SVCTAG=	=	177777	
ONEFIL=	000001		P.AAV	=	005400		P.ADA	006406	P.AFF	007430	SVCTST=	=	177777	
ONE.FM=	005350		P.AAW	=	005414		P.ADB	006420	P.AFG	007436	SYNC	=	007500	
OP.NUM	010322		P.AAX	=	005432		P.ADC	006426	P.AFH	007444	S\$LSYM=	=	010000	
OSAPTS=	000001		P.AAY	=	005452		P.ADD	006436	P.AFI	007500	TEN.FM=	=	005546	
OSAU	=	000001	P.AAZ	=	005474		P.ADE	006446	P.AFJ	007534	THR.FM=	=	005366	
OSBGNR=	000001		P.ABA	=	005520		P.ADF	006454	P.AFK	007566	TRBLE.=	=	007662	
OSBGNS=	000001		P.ABB	=	005546		P.ADG	006462	P.AFL	007622	TST.LN	=	012660	
OSDU	=	000001	P.ABC	=	005576		P.ADH	006472	P.AFM	007662	TWO.FM=	=	005356	
OSERRT=	000001		P.ABD	=	005630		P.ADI	006504	RAS.IN	011666	T\$ARGC=	=	000003	
OSGNSW=	000001		P.ABE	=	005634		P.ADJ	006510	RD.CS1	014244	T\$CODE=	=	000130	
OSPOIN=	000001		P.ABF	=	005646		P.ADK	006526	RD.DA	014600	T\$ERRN=	=	000000	
OSSETU=	000001		P.ABG	=	005654		P.ADL	006542	RD.DAT	011672	T\$EXCP=	=	000000	
PAR.DI	010330		P.ABH	=	005662		P.ADM	006560	RD.DS	017514	T\$FREE	=	065366	G
PD.TEM	011662		P.ABI	=	005670		P.ADN	006576	RD.D1	016366	T\$GMAN=	=	000000	
PHR.1	=	006542	P.ABJ	=	005676		P.ADO	006630	RD.D2	016704	T\$HILI=	=	000007	
PHR.10=	006740		P.ABK	=	005712		P.ADP	006646	RD.D3	017236	T\$LAST=	=	000000	
PHR.2	=	006560	P.ABL	=	005726		P.ADQ	006660	RD.EE	016242	T\$LOLI=	=	000000	
PHR.3	=	006576	P.ABM	=	005740		P.ADR	006672	RD.EL	016200	T\$LSYM=	=	010000	
PHR.4	=	006630	P.ABN	=	005750		P.ADS	006714	RD.ER	014422	T\$NEST=	=	177777	
PHR.5	=	006646	P.ABO	=	005760		P.ADT	006726	RD.E1	015356	T\$NS0	=	000000	
PHR.6	=	006660	P.ABP	=	005770		P.ADU	006740	RD.E2	015576	T\$NS1	=	000021	
PHR.7	=	006672	P.ABQ	=	005774		P.ADV	006752	RD.MR	014756	T\$PTHV=	=	000000	G
PHR.8	=	006714	P.ABR	=	006006		P.ADW	006766	RD.PA	015150	T\$PTNU=	=	000000	
PHR.9	=	006726	P.ABS	=	006014		P.ADX	006774	RD.PD	016040	T\$SAVL=	=	177777	
PNT	=	001000	P.ABT	=	006022		P.ADY	007002	RD.REG	020210	T\$SEGL=	=	177777	

TSSUBN= 000000	T34	055702 G	WRD.27= 006134	WRD.7 = 005676	\$T16	033770
TSTAGL= 177777	T35	057230 G	WRD.29= 006142	WRD.8 = 005712	\$T17	035134
TSTAGN= 010005	T36	057656 G	WRD.3 = 005646	WRD.9 = 005726	\$T18	036074
TSTEMP= 000000	T37	060650 G	WRD.30= 006156	WRT.CS 014174	\$T19	036470
TSTEST= 000000	T38	061702 G	WRD.31= 006164	WRT.DA 014530	\$T2	021730
TSTSTM= 177777	T39	062454 G	WRD.32= 006200	WRT.DS 017512	\$T20	037136
TSTSTS= 000000	T4	023462 G	WRD.33= 006206	WRT.D1 016302	\$T21	037474
TSSHAR= 010002	T40	063410 G	WRD.34= 006212	WRT.D2 016620	\$T22	040650
TSSHW = 010000	T41	063620 G	WRD.35= 006220	WRT.D3 017136	\$T23	041424
TSSPRO= 010004	T42	064004 G	WRD.36= 006226	WRT.EE 016240	\$T24	042550
TSSSOF= 010003	T43	064502 G	WRD.37= 006232	WRT.EL 016176	\$T25	043326
TSSSW = 010001	T44	064774 G	WRD.38= 006242	WRT.ER 014352	\$T26	044312
T1	T45	065326 G	WRD.39= 006250	WRT.E1 015272	\$T27	045424
T10	T5	024354 G	WRD.4 = 005654	WRT.E2 015500	\$T28	046220
T11	T6	025340 G	WRD.40= 006260	WRT.MR 014706	\$T29	047614
T12	T7	025412 G	WRD.41= 006264	WRT.PA 015064	\$T3	022416
T13	T8	026054 G	WRD.42= 006300	WRT.PD 015732	\$T30	051352
T14	T9	026320 G	WRD.43= 006306	WRT.RE 017622	\$T31	052364
T15	UAM =	000200 G	WRD.44= 006316	WT.DAT 011670	\$T32	053320
T16	WRD.1 =	005630	WRD.45= 006324	W.C.SI 011664	\$T33	053752
T17	WRD.10=	005740	WRD.46= 006336	XOR.LN 013354	\$T34	055010
T18	WRD.11=	005750	WRD.47= 006350	XSALWA= 000000	\$T35	055716
T19	WRD.12=	005760	WRD.48= 006360	XSFALSE= 000040	\$T36	057244
T2	WRD.13=	005770	WRD.49= 006370	XSOFFS= 000400	\$T37	057672
T20	WRD.14=	005774	WRD.5 = 005662	XSTRUE= 000020	\$T38	060664
T21	WRD.15=	006006	WRD.50= 006400	SEND.L 065370 G	\$T39	061716
T22	WRD.16=	006014	WRD.51= 006406	\$PATCH 002720 G	\$T4	023040
T23	WRD.17=	006022	WRD.52= 006420	\$SAVE2 004006 G	\$T40	062470
T24	WRD.18=	006034	WRD.53= 006426	\$SAVE3 004022 G	\$T41	063424
T25	WRD.19=	006040	WRD.54= 006436	\$SAVE4 004040 G	\$T42	063634
T26	WRD.2 =	005634	WRD.55= 006446	\$SAVE5 004060 G	\$T43	064020
T27	WRD.20=	006046	WRD.56= 006454	\$T1	\$T44	064516
T28	WRD.21=	006054	WRD.57= 006462	\$T10	\$T45	065010
T29	WRD.22=	006062	WRD.58= 006472	\$T11	\$T5	023476
T3	WRD.23=	006076	WRD.59= 006504	\$T12	\$T6	024370
T30	WRD.24=	006104	WRD.6 = 005670	\$T13	\$T7	025354
T31	WRD.25=	006112	WRD.60= 006510	\$T14	\$T8	025426
T32	WRD.26=	006126	WRD.61= 006526	\$T15	\$T9	026070
T33						

. ABS. 065372 000
000000 001

ERRORS DETECTED: 0

VIRTUAL MEMORY USED: 32783 WORDS (129 PAGES)

DYNAMIC MEMORY: 21558 WORDS (82 PAGES)

ELAPSED TIME: 00:14:51

ML11,ML11/-SP/CR:SYM=SVC/ML,CZMLAA.DOC,ML2.P11,B16PG1.P11,B16PG2.P11,B16PG3.P11,B16PG4.P11,B16MUL.P11,B16SAV.P11,ML3.P11,ML4.P11

SYMBOL	CROSS REFERENCE	VALUE	REFERENCES
ADR	= 000020	G	#78-1685
ARR.DA	= 007534		#81-1848 166-6293 169-6540 174-6858 181-7241 239-11163 241-11423 247-11769 247-11786
ARR.IN	010324		247-11807 248-11828 248-11845 259-12450 267-12871 312-15405 318-15738 342-17122 368-18548
ARR.16	010336		#73-1433 *155-5686 *155-5706 227-10156 227-10200 227-10202 257-12336 267-12857 267-12866
ASSEMB	= 000010		285-13871 285-13885 288-14036 288-14050
ASYN	= 007444		#74-1444 *155-5676 *155-5687 *155-5707 *155-5708 227-10201
			7-13 7-13
			#81-1846 158-5860 162-6077 166-6275 169-6522 173-6834 174-6897 180-7228 185-7505
			186-7529 186-7549 189-7698 192-7875 193-7896 193-7913 197-8130 198-8149 198-8166
			198-8183 199-8204 199-8239 200-8262 200-8286 204-8505 205-8524 205-8541 205-8558
			206-8579 206-8614 207-8637 207-8661 211-8911 211-8926 211-8943 211-8960 211-8980
			211-9015 211-9037 211-9061 214-9269 214-9285 214-9304 215-9325 215-9343 215-9361
			218-9521 223-9761 223-9778 223-9796 224-9832 224-9865 225-9912 227-10106 227-10122
			227-10139 227-10171 227-10216 229-10342 229-10358 229-10374 230-10394 233-10559 233-10578
			233-10599 236-10782 253-12132 272-13139 279-13567 289-14134 336-16798 352-17640 352-17660
			353-17703 353-17726 353-17742 356-17890
BIT0	= 000001	G	#78-1669
BIT00	= 000001	G	#77-1655
BIT01	= 000002	G	#77-1654
BIT02	= 000004	G	#77-1653
BIT03	= 000010	G	#77-1652
BIT04	= 000020	G	#77-1651
BIT05	= 000040	G	#77-1650
BIT06	= 000100	G	#77-1649
BIT07	= 000200	G	#77-1648
BIT08	= 000400	G	#77-1647
BIT09	= 001000	G	#77-1646
BIT1	= 000002	G	#78-1668
BIT10	= 002000	G	#77-1645
BIT11	= 004000	G	#77-1644
BIT12	= 010000	G	#77-1643
BIT13	= 020000	G	#77-1642
BIT14	= 040000	G	#77-1641
BIT15	= 100000	G	#77-1640
BIT2	= 000004	G	#78-1667
BIT3	= 000010	G	#78-1666
BIT4	= 000020	G	#78-1665
BIT5	= 000040	G	#78-1664
BIT6	= 000100	G	#78-1663
BIT7	= 000200	G	#78-1662
BIT8	= 000400	G	#77-1657
BIT9	= 001000	G	#77-1656
BL\$DIV	003714	G	#40-274
BL\$GT1	002760	G	#12-143 318-15730 370-18713
BL\$GT2	003102	G	#18-153 222-9751 224-9826 225-9906 239-11052 239-11133 246-11719 259-12422 266-12815
			271-13108 277-13440 279-13553 285-13910 288-14074 295-14434 301-14782 312-15383 312-15392
			325-16108 342-17109 370-18712
BL\$LAS	065362	G	#370-18681 370-18682 370-18686
BL\$MOD	003726	G	#41-304
BL\$MUL	003470	G	#37-128
BL\$PU1	003244	G	#24-140
BL\$PU2	003340	G	#31-183 224-9859 312-15426 370-18713

SYMBOL	CROSS REFERENCE	VALUE	REFERENCES
C\$RFLA	=	000021	#7-13
C\$RPT	=	000025	#7-13
C\$SEFG	=	000046	#7-13
C\$SPRI	=	000041	#7-13
C\$SVEC	=	000037	#7-13
C\$TPRI	=	000013	#7-13
DAT.DM		012626	#87-2145 246-11746 270-13046 270-13067 293-14353 294-14393 300-14709 300-14741 311-15340 324-16066 341-17028 341-17065
DFPTBL		002320	G #7-119
DIAGMC	=	000000	7-13 7-13
DIVMOD		003532	#38-196 40-275 41-305
DRIVE.		011674	#74-1456 *155-5682 *155-5702 174-6893 175-6910
D1.TEM	=	007714	#81-1852 *253-12101 *258-12404 265-12746 *266-12800 *271-13094 *277-13426 *279-13539 *294-14416 *301-14768 *311-15362 *325-16093 *342-17095
D2.TEM	=	007716	#81-1853 *253-12102 *258-12405 265-12747 *266-12801 *271-13095 *277-13427 *279-13540 *294-14417 *301-14769 *311-15363 *325-16094 *342-17096
EF.CON	=	000036	G #78-1672
EF.NEW	=	000035	G #78-1673
EF.PWR	=	000034	G #78-1674
EF.RES	=	000037	G #78-1671
EF.STA	=	000040	G #78-1670
EIG.FM	=	005474	#79-1726
ELV.FM	=	005576	#79-1729
ERRBLK		002160	G #7-96
ERRMSG		002156	G #7-96
ERRNBR		002154	G #7-96
ERRTYP		002152	G #7-96
EVL	=	000004	G #78-1683
E\$END	=	002100	#7-13
E\$LOAD	=	000035	#7-13 7-65
E2.TEM	=	007720	#81-1854 *253-12103 *258-12406 265-12748 *266-12802 *271-13096 *277-13428 *279-13541 *294-14418 *301-14770 *311-15364 *325-16095 *342-17097
FIRS1.		012520	#84-1997 197-8124 204-8499 211-8902 232-10547 241-11353 252-12053 252-12075 278-13494
FIV.FM	=	005414	#79-1723 193-7886 193-7903 193-7920 197-8137 198-8156 198-8173 198-8190 199-8211 200-8255 200-8269 200-8298 204-8512 205-8531 205-8548 205-8565 206-8586 206-8626 207-8644 207-8673 211-8918 211-8933 211-8950 211-8970 211-8987 211-9030 211-9044 211-9076 214-9277 214-9292 215-9315 215-9333 215-9351 223-9785 224-9839 225-9876 227-10113 227-10130 227-10149 227-10178 227-10223 229-10349 229-10365 230-10385 230-10401 233-10566 233-10585 233-10606 253-12139 272-13146 301-14802 331-16465 335-16740 348-17426 353-17733 356-17897
FMT.1	=	004164	#78-1697 163-6092
FMT.10	=	004634	#78-1706 239-11105
FMT.11	=	004706	#78-1707
FMT.12	=	004742	#78-1708 233-10571 233-10590 234-10615
FMT.13	=	004772	#78-1709 239-11199
FMT.14	=	005046	#78-1710 290-14150
FMT.15	=	005106	#78-1711 326-16191
FMT.16	=	005154	#78-1712 167-6316 169-6562 174-6873 181-7257
FMT.17	=	005244	#78-1713 156-5731
FMT.18	=	005300	#79-1718
FMT.2	=	004224	#78-1698 162-6040 175-6911 189-7715 241-11442 305-15040 332-16480
FMT.3	=	004312	#78-1699 183-7364

SYMBOL	CROSS REFERENCE	VALUE	REFERENCES
FMT.4	=	004336	#78-1700 218-9532
FMT.5	=	004366	#78-1701 301-14809 326-16162 326-16198 327-16233
FMT.6	=	004470	#78-1702 342-17132
FMT.7	=	004520	#78-1703 226-9929 248-11862 352-17650 352-17670 353-17713
FMT.8	=	004546	#78-1704 368-18572 368-18583
FMT.9	=	004602	#78-1705 227-10183 227-10228 259-12460 337-16814 368-18558
FNC.1	=	006752	#80-1804 189-7701
FNC.10	=	007074	#80-1813 227-10109 227-10125 227-10145 227-10174 227-10219
FNC.11	=	007104	#80-1814 227-10126 229-10345 229-10361 229-10377 230-10397
FNC.12	=	007124	#80-1815 233-10562 233-10581 233-10602
FNC.13	=	007136	#80-1816 277-13464 335-16739 347-17378 347-17400 348-17421
FNC.14	=	007146	#80-1817 253-12134
FNC.15	=	007162	#80-1818 267-12873 287-14017 335-16744
FNC.16	=	007174	#80-1819 272-13142
FNC.17	=	007206	#80-1820 287-14015 289-14138 335-16738 337-16807
FNC.18	=	007216	#80-1821 312-15412 318-15745
FNC.19	=	007230	#80-1822 359-18016
FNC.2	=	006766	#80-1805 193-7882 193-7899 193-7916 223-9799 224-9835
FNC.21	=	007242	#80-1823 239-11098
FNC.22	=	007254	#80-1824 277-13456
FNC.23	=	007266	#80-1825
FNC.3	=	006774	#80-1806 161-6022 162-6081 214-9273 215-9329 215-9347 216-9372 361-18161 366-18399
FNC.4	=	007002	#80-1807 197-8133 198-8186 199-8207 199-8247 200-8265 200-8294 366-18404
FNC.5	=	007020	#80-1808 166-6298 169-6548 204-8508 205-8561 206-8582 206-8622 207-8640 207-8669
			208-8694 211-8966 211-8983 211-9026 211-9040 211-9069 295-14469 296-14495 301-14797
			312-15408 326-16149 326-16181 327-16220 366-18401
FNC.6	=	007030	#80-1809 166-6297 169-6547 211-8914 211-9093 239-11093 305-15023 318-15741 331-16463
			366-18400
FNC.7	=	007036	#80-1810 214-9272 215-9328 215-9346 216-9371
FNC.8	=	007046	#80-1811 218-9523 218-9526 223-9763 223-9780
FNC.9	=	007062	#80-1812
FOR.FM	=	005400	#79-1722 218-9527 223-9767 223-9803 236-10788 236-10803 239-11186 241-11429 289-14140
			337-16808 347-17380 347-17402 354-17752 359-18020 362-18196 363-18235
F\$AU	=	000015	#7-13
F\$AUTO	=	000020	#7-13
F\$BGN	=	000040	#7-13 7-39 8-226 8-274 8-305 8-337
F\$CLEA	=	000007	#7-13
F\$DU	=	000016	#7-13
F\$END	=	000041	#7-13 7-13 7-13 7-13 7-13 7-13 7-13 7-13 7-13 7-13
			7-13 7-13 7-13 7-13 7-13 7-13 7-13 7-13 7-13 7-13
			8-247 8-288 8-337
F\$HARD	=	000004	#7-13 8-226 8-247
F\$HW	=	000013	#7-13 7-119 7-139
F\$INIT	=	000006	#7-13
F\$JMP	=	000050	#7-13
F\$MOD	=	000000	#7-13 7-39 8-337
F\$MESSG	=	000011	#7-13
F\$PROT	=	000021	#7-13 8-305 8-311
F\$PWR	=	000017	#7-13
F\$RPT	=	000012	#7-13
F\$SEG	=	000003	#7-13
F\$SOFT	=	000005	#7-13 8-274 8-288

ML11 SYMBOL	CROSS REFERENCE VALUE	MACRO REFERENCES	ON 23-OCT-80 AT 09:15	PAGE 5	G 14	SEQ 0383
SYMBOL	VALUE	REFERENCES		CREF	V01	
FSSRV	= 000010	#7-13				
FSSUB	= 000002	#7-13				
FSSW	= 000014	#7-13	7-150	7-162		
FSTEST	= 000001	#7-13				
GD.BLK	012542	#85-2046	294-14380	300-14732	304-14987	305-15008
		331-16425	331-16442	341-17053	346-17330	346-17352
		365-18385				
GOOD.B	010326	#73-1434	85-2048	87-2147	*155-5675	*266-12846
GSCNTO	= 000200	#7-13				
GSDLM	= 000372	#7-13				
GSDISP	= 000003	#7-13				
GSEXCP	= 000400	#7-13				
GSHILI	= 000002	#7-13				
GSLOLI	= 000001	#7-13				
GSNO	= 000000	#7-13				
GSOFFS	= 000400	#7-13	8-239	8-240	8-241	8-242
GSOFSI	= 000376	#7-13	8-239	8-240	8-241	8-242
GSPRMA	= 000001	#7-13	8-239			
GSPRMD	= 000002	#7-13	8-240	8-241	8-242	8-244
GSPRML	= 000000	#7-13	8-243	8-245	8-285	
GSRADA	= 000140	#7-13				
GSRADB	= 000000	#7-13				
GSRADD	= 000040	#7-13	8-242			
GSRADL	= 000120	#7-13	8-243	8-245	8-285	
GSRADO	= 000020	#7-13	8-239	8-240	8-241	8-244
GSXFER	= 000004	#7-13				
GSYES	= 000010	#7-13	8-239	8-240	8-241	8-242
HELP	= 000000	#7-4	7-8	7-30	7-48	7-67
		8-228	8-249	8-276	8-291	8-313
HOE	= 100000	G #78-1696				
HW.OR.	007722	G #73-1427	*241-11371	241-11414		
IBE	= 010000	G #78-1693				
IDU	= 000040	G #78-1686				
IER	= 020000	G #78-1694				
INTER	= 007622	#81-1850	161-6017	208-8691	211-9090	239-11091
IO.BUF	010342	#74-1446	84-1999	85-2049	86-2098	87-2148
		285-13884	288-14049	*294-14378	*300-14731	*304-14986
		*317-15694	318-15724	*324-16045	*324-16046	*331-16419
		335-16699	335-16708	*336-16767	336-16770	*336-16781
		*340-17020	*346-17329	*346-17344	347-17370	347-17392
		*362-18201	*363-18243	*365-18374	368-18539	
ISR	= 000100	G #78-1687				
IXE	= 004000	G #78-1692				
ISAU	= 000041	#7-13				
ISAUTO	= 000041	#7-13				
ISCLN	= 000041	#7-13				
ISDU	= 000041	#7-13				
ISHRD	= 000041	#8-226	#8-247			
ISINIT	= 000041	#7-13				
ISMOD	= 000041	#7-13	7-39	#7-39	8-337	#8-337
ISMSG	= 000041	#7-13				
ISPROT	= 000040	#7-13	#8-305			

SYMBOL	CROSS REFERENCE	VALUE	REFERENCES
ISPTAB	=	000041	#7-13
ISPWR	=	000041	#7-13
ISRPT	=	000041	#7-13
ISSEG	=	000041	#7-13
ISSETU	=	000041	#7-13
ISSFT	=	000041	#8-274 #8-288
ISSRV	=	000041	#7-13
ISSUB	=	000041	#7-13
ISTST	=	000041	#7-13
JSJMP	=	000167	#7-13
LAST.B		012574	#86-2095 276-13377 276-13399 278-13512 353-17693
LAU		004152	#46-181 46-198
LAUTO		004126	#46-83 46-100
LCLEAN		065342	#369-18639 369-18658
LDU		004140	#46-131 46-148
LD.LNG		013636	#96-2672 264-12737
LINIT		020600	#154-5613 156-5758
LOAD.S		012172	#81-1861 325-16119
LOE	=	040000 G	#78-1695
LOT	=	000010 G	#78-1684
LRPT		004114	#46-32 46-49
LST.AR		010340	#74-1445 *155-5677 *155-5692 *155-5693 *156-5719 *156-5720 227-10155 227-10199 257-12335 267-12856 267-12865
LST.BL		010334	#74-1443 86-2097 *155-5678 *155-5698 *155-5699 *155-5700 *156-5727 *156-5728 *156-5729 278-13482 351-17614 353-17697 353-17712 356-17874 359-18006 368-18528
LSACP		002110 G	#7-65
LSAPT		002036 G	#7-65
LSAU		004154 G	7-65 #46-198
LSAUT		002070 G	#7-65
LSAUTO		004130 G	7-65 #46-100
LSCCP		002106 G	#7-65
LSCLEA		065352 G	7-65 #369-18657
LSCO		002032 G	#7-65
LSDEPO		002011 G	#7-65
LSDESC		002130 G	7-65 #7-87
LSDESP		002076 G	#7-65
LSDEVP		002060 G	#7-65
LSDISP		002164 G	7-65 #7-103
LSDLY		002116 G	#7-65 158-5831 158-5849 162-6058 192-7864 199-8219 206-8594 211-8995 214-9258 236-10768 239-11031 241-11362 241-11386 241-11399 252-12058 252-12079 253-12094 258-12362 258-12385 258-12397 265-12753 265-12776 265-12789 270-13051 270-13071 271-13087 276-13382 276-13403 277-13419 278-13516 279-13532 285-13890 288-14055 293-14358 294-14397 294-14409 300-14714 300-14745 300-14757 311-15344 311-15355 324-16070 324-16082 341-17033 341-17069 342-17088
LSDTP		002040 G	#7-65
LSDTYP		002034 G	#7-65
LSDU		004142 G	7-65 #46-148
LSDUT		002072 G	#7-65
LSDVTY		002122 G	7-65 #7-80
LSEF		002052 G	#7-65
LSEVI		002044 G	#7-65
LSERRT		002152 G	7-65 #7-96

SYMBOL	CROSS REFERENCE	VALIJE	REFERENCES	CREF	V01							
L\$ETP		002102	G	#7-65								
L\$EXP1		002046	G	#7-65								
L\$EXP4		002064	G	#7-65								
L\$EXP5		002066	G	#7-65								
L\$HARD		002344	G	7-65	8-226	#8-226						
L\$HIME		002120	G	#7-65								
L\$HPCP		002016	G	#7-65								
L\$HPTP		002022	G	#7-65								
L\$HW		002320	G	7-65	7-119	#7-119						
L\$ICP		002104	G	#7-65								
L\$INIT		021512	G	7-65	#156-5758							
L\$LADP		002026	G	#7-65								
L\$LAST	=	065366	G	7-65	#370-18686							
L\$LOAD		002100	G	#7-65								
L\$LUN		002074	G	#7-65								
L\$MREV		002050	G	#7-65								
L\$NAME		002000	G	#7-65								
L\$PRIO		002042	G	#7-65								
L\$PROT		002712	G	7-65	#8-305							
L\$PRT		002112	G	#7-65								
L\$REPP		002062	G	#7-65								
L\$REV		002010	G	#7-65								
L\$RPT		004116	G	7-65	#46-49							
L\$SOFT		002662	G	7-65	8-274	#8-274						
L\$SPC		002056	G	#7-65								
L\$SPCP		002020	G	#7-65								
L\$SPTP		002024	G	#7-65								
L\$STA		002030	G	#7-65								
L\$SW		002340	G	7-65	7-150	#7-150						
L\$TEST		002114	G	#7-65								
L\$TIML		002014	G	#7-65								
L\$UNIT		002012	G	#7-65	154-5607	154-5623	154-5655					
L10000		002336		7-119	#7-139							
L10001		002342		7-150	#7-162							
L10002		002440		8-226	#8-247							
L10003		002670		8-274	#8-288							
MEM.AR	=	007566		#81-1849								
ML.DUT		012170	G	#77-1637	131-4496	132-4522	135-4686	136-4712	139-4890	140-4919	*155-5670	156-5736
				158-5841	161-6005	162-6033	162-6039	162-6047	163-6091	165-6249	169-6500	172-6783
				173-6811	178-7106	185-7495	186-7520	186-7540	189-7680	192-7856	197-8119	199-8231
				200-8278	204-8494	206-8606	207-8653	211-8897	211-9007	211-9053	213-9245	218-9505
				222-9732	224-9815	224-9822	224-9848	224-9855	225-9890	227-10096	227-10161	227-10206
				229-10333	232-10542	236-10750	239-11005	241-11347	241-11377	245-11694	252-12045	252-12070
				257-12343	258-12374	264-12712	265-12765	270-13037	270-13063	276-13367	276-13394	278-13489
				278-13507	285-13876	288-14042	293-14345	294-14374	294-14386	299-14698	300-14726	300-14737
				304-14981	305-14997	310-15319	311-15332	316-15634	317-15699	317-15708	323-16032	324-16061
				330-16406	331-16438	335-16719	336-16761	336-16776	340-17006	341-17045	341-17058	346-17324
				346-17337	351-17621	352-17684	356-17878	358-17996	362-18174	362-18212	365-18369	365-18380
				368-18533								
ML.LUN		012166	G	#77-1636	*154-5621	*154-5622	154-5623	154-5626	*154-5653	*154-5654	154-5655	155-5662
				156-5730	158-5874	163-6104	167-6335	169-6585	175-6918	181-7280	190-7730	193-7931
				201-8314	208-8707	211-9106	216-9384	219-9551	239-11211	241-11453	249-11881	254-12147

ML11
SYMBOL CROSS REFERENCE
SYMBOL VALUE

CREATED BY MACRO ON 23-OCT-80 AT 09:15

PAGE 8
CREF V01

J 14

SEQ 0386

ML.REG 011700 G

REFERENCES	267-12880	272-13150	280-13600	296-14483	302-14832	306-15058	313-15449	319-15773
260-12476	267-12880	272-13150	280-13600	296-14483	302-14832	306-15058	313-15449	319-15773
328-16262	332-16495	336-16755	337-16818	343-17159	354-17763			
#74-1463	84-1998	84-1999	84-2000	85-2047	85-2048	85-2049	85-2050	86-2096
86-2097	86-2098	86-2099	87-2146	87-2147	87-2148	87-2149	98-2797	98-2799
98-2800	98-2802	100-2895	100-2897	100-2898	100-2901	100-2902	101-2954	101-2956
101-2957	101-2959	103-3053	103-3055	103-3056	103-3059	103-3060	104-3112	104-3114
104-3115	104-3117	106-3210	106-3212	106-3213	106-3216	106-3217	107-3269	107-3271
107-3272	107-3274	109-3368	109-3370	109-3371	109-3374	109-3375	110-3420	110-3426
110-3428	110-3429	110-3431	110-3432	112-3521	112-3527	112-3529	112-3530	112-3533
112-3534	112-3539	113-3580	113-3586	113-3588	113-3589	113-3591	113-3592	115-3681
115-3687	115-3689	115-3690	115-3693	115-3694	115-3699	116-3753	*116-3754	116-3760
116-3762	116-3763	116-3768	*116-3769	116-3770	118-3860	*118-3861	118-3867	118-3869
118-3870	118-3873	118-3874	*118-3879	118-3880	119-3934	119-3935	119-3941	119-3943
120-3948	120-3950	120-3951	120-3952	120-3953	122-4046	122-4047	122-4053	122-4055
122-4056	122-4059	122-4060	122-4065	122-4066	125-4197	128-4330	129-4374	129-4380
129-4382	129-4383	129-4385	129-4386	131-4489	131-4490	131-4491	131-4492	131-4495
131-4498	131-4499	131-4500	131-4507	132-4513	132-4514	132-4517	132-4521	132-4524
132-4525	133-4566	133-4572	133-4574	133-4575	133-4577	133-4578	135-4679	135-4680
135-4681	135-4682	135-4685	135-4688	135-4689	135-4690	135-4697	135-4699	136-4704
136-4707	136-4711	136-4714	136-4715	137-4765	*137-4766	137-4772	137-4774	137-4775
137-4780	*137-4781	137-4782	*139-4882	139-4883	139-4884	139-4885	139-4886	139-4889
139-4892	139-4893	139-4894	140-4905	140-4907	140-4908	140-4911	140-4912	*140-4917
140-4918	140-4921	140-4922	143-5063	143-5065	143-5066	143-5069	143-5070	*154-5644
156-5735	156-5738	156-5739	158-5840	158-5843	158-5844	158-5846	158-5856	161-6004
161-6008	161-6009	161-6011	161-6012	162-6053	162-6054	162-6055	162-6070	162-6071
162-6072	165-6248	165-6251	165-6252	167-6315	169-6499	169-6502	169-6503	169-6561
*172-6775	172-6782	172-6785	173-6790	173-6801	173-6802	173-6810	173-6813	173-6814
174-6872	*174-6892	174-6893	178-7105	178-7108	178-7109	181-7256	183-7363	185-7494
185-7497	185-7498	185-7499	185-7500	185-7501	186-7519	186-7522	186-7523	186-7524
186-7525	186-7539	186-7542	186-7543	186-7544	186-7545	189-7679	189-7682	189-7683
189-7687	189-7706	192-7855	192-7858	192-7859	192-7861	192-7871	193-7892	193-7909
197-8118	197-8121	197-8122	197-8125	197-8126	198-8145	198-8162	198-8179	199-8200
199-8226	199-8228	199-8230	199-8233	199-8234	199-8235	200-8275	200-8277	200-8280
200-8281	200-8282	204-8493	204-8496	204-8497	204-8500	204-8501	204-8516	205-8537
205-8554	205-8571	206-8601	206-8603	206-8605	206-8608	206-8609	206-8610	207-8650
207-8652	207-8655	207-8656	207-8657	207-8683	211-8896	211-8899	211-8900	211-8903
211-8904	211-8922	211-8939	211-8956	211-8976	211-9002	211-9004	211-9006	211-9009
211-9010	211-9011	211-9050	211-9052	211-9055	211-9056	211-9057	211-9086	213-9244
213-9247	213-9248	213-9250	214-9255	214-9265	214-9281	214-9300	215-9321	215-9339
215-9357	218-9504	218-9507	218-9508	218-9516	218-9517	222-9731	222-9735	222-9736
222-9737	222-9738	222-9739	223-9774	223-9791	223-9792	224-9814	224-9845	224-9846
224-9860	224-9861	225-9886	225-9887	225-9893	225-9894	227-10095	227-10098	227-10099
227-10100	227-10101	227-10102	227-10118	227-10135	227-10160	227-10163	227-10164	227-10165
227-10166	227-10167	227-10205	227-10208	227-10209	227-10210	227-10211	227-10212	229-10332
229-10335	229-10336	229-10337	229-10338	229-10354	229-10370	230-10390	232-10541	232-10544
232-10545	232-10548	232-10549	232-10551	233-10595	236-10749	236-10752	236-10753	236-10754
236-10761	236-10763	236-10765	236-10778	236-10793	239-11004	239-11010	239-11011	239-11013
239-11028	239-11038	241-11346	241-11349	241-11350	241-11352	241-11354	241-11355	241-11371
241-11372	241-11376	241-11379	241-11380	241-11381	241-11383	241-11393	241-11396	241-11406
245-11693	245-11696	245-11697	246-11747	246-11749	246-11750	246-11751	246-11752	252-12044
252-12047	252-12048	252-12049	252-12050	252-12051	252-12052	252-12054	252-12065	252-12069
252-12072	252-12073	252-12074	252-12076	253-12091	253-12101	253-12102	253-12103	257-12342

REFERENCES

257-12345	257-12346	257-12347	257-12348	257-12349	257-12350	257-12351	258-12356	258-12357
258-12358	258-12369	258-12373	258-12376	258-12377	258-12378	258-12379	258-12380	258-12381
258-12382	258-12393	258-12394	258-12404	258-12405	258-12406	264-12711	264-12714	264-12715
264-12716	264-12717	264-12718	264-12719	265-12746	265-12747	265-12748	265-12749	265-12760
265-12764	265-12767	265-12768	265-12769	265-12770	265-12771	265-12772	265-12773	265-12785
265-12786	266-12800	266-12801	266-12802	270-13036	270-13039	270-13040	270-13043	270-13044
270-13045	270-13047	270-13058	270-13062	270-13065	270-13066	270-13068	271-13083	271-13084
271-13094	271-13095	271-13096	276-13366	276-13369	276-13370	276-13373	276-13374	276-13375
276-13376	276-13378	276-13389	276-13393	276-13396	276-13397	276-13398	276-13400	277-13415
277-13416	277-13426	277-13427	277-13428	278-13484	278-13485	278-13486	278-13488	278-13491
278-13492	278-13493	278-13495	278-13499	278-13506	278-13509	278-13510	278-13511	278-13513
279-13528	279-13529	279-13539	279-13540	279-13541	285-13875	285-13878	285-13879	285-13882
285-13883	285-13884	285-13886	285-13887	285-13897	286-13932	286-13940	286-13941	286-13946
286-13955	286-13956	286-13960	286-13968	286-13969	287-13978	287-13987	287-13988	287-13993
287-13994	287-14001	288-14041	288-14044	288-14045	288-14047	288-14048	288-14049	288-14051
288-14052	288-14062	288-14078	289-14095	289-14097	289-14099	289-14101	289-14103	289-14105
289-14111	289-14113	289-14119	289-14127	293-14344	293-14347	293-14348	293-14350	293-14351
293-14352	293-14354	294-14369	294-14373	294-14376	294-14377	294-14379	294-14381	294-14382
294-14385	294-14388	294-14389	294-14394	294-14405	294-14406	294-14416	294-14417	294-14418
299-14697	299-14700	299-14701	299-14702	300-14707	300-14708	300-14710	300-14721	300-14725
300-14728	300-14729	300-14730	300-14733	300-14734	300-14736	300-14739	300-14740	300-14742
300-14753	300-14754	301-14768	301-14769	301-14770	304-14980	304-14983	304-14984	304-14985
305-14992	305-14993	305-14996	305-14999	305-15000	305-15009	305-15010	310-15318	310-15321
311-15326	311-15328	311-15329	311-15331	311-15334	311-15335	311-15341	311-15351	311-15352
311-15362	311-15363	311-15364	316-15633	316-15636	316-15637	317-15687	317-15688	317-15698
317-15701	317-15702	317-15704	317-15705	317-15707	318-15714	318-15715	323-16031	324-16038
324-16039	324-16052	324-16053	324-16060	324-16063	324-16064	324-16067	324-16078	324-16079
325-16093	325-16094	325-16095	330-16405	330-16408	330-16409	331-16426	331-16427	331-16437
331-16440	331-16441	331-16443	331-16444	335-16695	335-16696	335-16698	335-16699	335-16700
335-16701	335-16702	335-16703	335-16705	335-16707	335-16708	335-16709	335-16710	335-16711
335-16712	335-16714	335-16718	335-16721	335-16722	335-16723	335-16724	335-16727	335-16729
336-16760	336-16763	336-16764	336-16765	336-16766	336-16768	336-16769	336-16770	336-16771
336-16772	336-16773	336-16775	336-16778	336-16779	336-16780	336-16782	336-16785	336-16786
336-16789	336-16790	336-16791	336-16792	336-16794	337-16812	340-17005	340-17008	340-17009
340-17021	341-17026	341-17027	341-17029	341-17040	341-17044	341-17047	341-17048	341-17052
341-17054	341-17055	341-17057	341-17060	341-17061	341-17066	342-17084	342-17085	342-17095
342-17096	342-17097	346-17323	346-17326	346-17327	346-17328	346-17331	346-17332	346-17336
346-17339	346-17340	346-17350	346-17351	346-17353	347-17362	347-17363	347-17411	351-17620
351-17623	351-17624	351-17625	351-17626	351-17627	351-17628	351-17629	351-17630	352-17636
352-17656	352-17683	352-17686	352-17687	353-17692	353-17694	353-17695	353-17699	353-17719
353-17721	353-17722	356-17877	356-17880	356-17881	356-17882	356-17883	356-17884	356-17885
356-17886	358-17995	358-17998	358-17999	359-18004	359-18005	359-18006	359-18007	359-18008
359-18010	362-18173	362-18176	362-18177	362-18178	362-18183	362-18184	362-18186	362-18202
362-18211	362-18214	362-18215	362-18216	362-18218	363-18223	363-18225	363-18245	363-18246
365-18368	365-18371	365-18372	365-18373	365-18376	365-18377	365-18379	365-18382	365-18383
365-18384	365-18386	366-18391	366-18393	368-18532	368-18535	368-18536	368-18537	368-18538
368-18539	368-18540	368-18541	368-18542	368-18544	369-18639			

MSGH1 002440
MSGH2 002454
MSGH3 002502
MSGH4 002525
MSGH5 002556

8-239 #8-256
8-240 #8-257
8-241 #8-258
8-242 #8-259
8-243 #8-260

SYMBOL	CROSS REFERENCE	VALUE	REFERENCES
MSGH6		002610	8-244 #8-261
MSGH7		002634	8-245 #8-262
MSGS1		002670	8-285 #8-297
NIB.SA		007714	#73-1425 81-1852 81-1853 81-1854 82-1887 82-1893 82-1899 82-1905 82-1911 82-1917 82-1927 82-1933 83-1950 90-2302 90-2309 90-2320 91-2332 91-2344 91-2351 91-2362 91-2370 91-2382 92-2394 94-2530 94-2533 94-2541 94-2543 94-2545 94-2552 94-2558 94-2560 94-2566 94-2570 *96-2689 *96-2696 *96-2700 *97-2712 *97-2713 *97-2716 *97-2723 *97-2727 *97-2735 *97-2736 *97-2740 *97-2748 *97-2749
NIN.FM	=	005520	#79-1727
ONEFIL	=	000001	#2-4 5-1176 6-1177 7-34 7-164 8-190 8-353
ONE.FM	=	005350	#79-1719 158-5863
OP.NUM		010322	#73-1431 *155-5674 155-5688 155-5694 155-5709 156-5721 189-7685 189-7713 227-10194 285-13872 288-14037
OSAPTS	=	000001	#7-13 #7-46 7-65
OSAU	=	000001	#7-13 #7-46 7-65
OSBGNR	=	000001	#7-13 #7-46 7-65
OSBGNS	=	000001	#7-13 #7-46 7-65
OSDU	=	000001	#7-13 #7-46 7-65
OSERRT	=	000001	#7-13 #7-46 7-65
OSGNSW	=	000001	#7-13 #7-46 7-65
OSPOIN	=	000001	#7-13 #7-46 7-46 7-65
OSSETU	=	000001	#7-13 #7-46 7-65
PAR.DI		010330	#74-1440 *155-5668 185-7490 361-18157
PD.TEM		011662	#74-1450 *258-12393 259-12416 *265-12785 266-12809 *271-13083 271-13102 *277-13415 277-13434 *279-13528 279-13547 *285-13897 285-13904 *288-14062 288-14068 *294-14405 295-14428 *300-14753 301-14776 *311-15351 311-15374 *324-16078 325-16102 *342-17084 342-17103
PHR.1	=	006542	#80-1794 197-8135 198-8154 200-8267 204-8510 205-8529 207-8642 211-8916 211-8931 211-9042 215-9313 218-9525 223-9765 223-9782 227-10147 227-10221 230-10399 233-10583 236-10786 236-10801 247-11813 248-11834 248-11851 348-17424 354-17750 356-17895 359-18018
PHR.10	=	006740	#80-1803 267-12874 287-14018 335-16745
PHR.2	=	006560	#80-1795 193-7884 198-8171 199-8249 200-8296 205-8546 206-8624 207-8671 211-8948 211-9028 211-9071 223-9801 224-9837 225-9874 227-10111 229-10347 233-10564 353-17731
PHR.3	=	006576	#80-1796 158-5862
PHR.4	=	006630	#80-1797 161-6024 166-6300 169-6550 174-6865 175-6903 241-11425 253-12137 272-13144 277-13458 295-14473 296-14499 326-16151 326-16183 327-16222 331-16460 368-18551
PHR.5	=	006646	#80-1798 193-7901 193-7918 198-8168 198-8188 199-8209 205-8543 205-8563 206-8584 208-8696 211-8945 211-8968 211-8985 211-9095 214-9275 214-9287 214-9290 215-9331 215-9349 227-10128 227-10176 229-10363 229-10379 233-10604 247-11775 247-11792 352-17643
PHR.6	=	006660	#80-1799 198-8151 205-8526 211-8928 214-9306 225-9918
PHR.7	=	006672	#80-1800 225-9914
PHR.8	=	006714	#80-1801 347-17379
PHR.9	=	006726	#80-1802 347-17401
PNT	=	001000	G #78-1690
PRI	=	002000	G #78-1691
PRI00	=	000000	G #78-1682
PRI01	=	000040	G #78-1681
PRI02	=	000100	G #78-1680
PRI03	=	000140	G #78-1679
PRI04	=	000200	G #78-1678
PRI05	=	000240	G #78-1677
PRI06	=	000300	G #78-1676

SYMBOL	CROSS REFERENCE	VALUE	REFERENCES
PRI07	=	000340 G	#78-1675
PRSN		002340 G	#7-160 183-7355 183-7361
PTBL.P		010320	#73-1429 *154-5628 154-5629 154-5631 154-5632 154-5634 *155-5664 155-5665 155-5667 155-5669 155-5671 155-5679
P.AAA		004164	#60-704 78-1697
P.AAB		004224	#61-719 78-1698
P.AAC		004312	#61-737 78-1699
P.AAD		004336	#61-744 78-1700
P.AAE		004366	#61-752 78-1701
P.AAF		004470	#62-778 78-1702
P.AAG		004520	#62-786 78-1703
P.AAH		004546	#62-794 78-1704
P.AAI		004602	#62-804 78-1705
P.AAJ		004634	#62-813 78-1706
P.AAK		004706	#63-831 78-1707
P.AAL		004742	#63-841 78-1708
P.AAM		004772	#63-849 78-1709
P.AAN		005046	#63-864 78-1710
P.AAO		005106	#63-875 78-1711
P.AAP		005154	#64-892 78-1712
P.AAQ		005244	#64-911 78-1713
P.AAR		005300	#64-921 79-1718
P.AAS		005350	#65-939 79-1719
P.AAT		005356	#65-941 79-1720
P.AAU		005366	#65-944 79-1721
P.AAV		005400	#65-948 79-1722
P.AAW		005414	#65-952 79-1723
P.AAX		005432	#65-957 79-1724
P.AAY		005452	#65-963 79-1725
P.AAZ		005474	#65-969 79-1726
P.ABA		005520	#65-976 79-1727
P.ABB		005546	#65-984 79-1728
P.ABL		005576	#66-996 79-1729
P.ABD		005630	#66-1005 79-1730
P.ABE		005634	#66-1007 79-1731
P.ABF		005646	#66-1011 79-1732
P.ABG		005654	#66-1013 79-1733
P.ABH		005662	#66-1015 79-1734
P.ABI		005670	#66-1017 79-1735
P.ABJ		005676	#66-1019 79-1736
P.ABK		005712	#66-1023 79-1737
P.ABL		005726	#66-1027 79-1738
P.ABM		005740	#66-1031 79-1739
P.ABN		005750	#66-1034 79-1740
P.ABO		005760	#66-1037 79-1741
P.ABP		005770	#66-1040 79-1742
P.ABQ		005774	#66-1042 79-1743
P.ABR		006006	#67-1050 79-1744
P.ABS		006014	#67-1052 79-1745
P.ABT		006022	#67-1054 79-1746
P.ABU		006034	#67-1058 79-1747
P.ABV		006040	#67-1060 79-1748

SYMBOL	CROSS REFERENCE	VALUE	REFERENCES
P.ABW		006046	#67-1062 79-1749
P.ABX		006054	#67-1064 79-1750
P.ABY		006062	#67-1066 79-1751
P.ABZ		006076	#67-1070 79-1752
P.ACA		006104	#67-1072 79-1753
P.ACB		006112	#67-1074 79-1754
P.ACC		006126	#67-1078 79-1755
P.ACD		006134	#67-1080 79-1756
P.ACE		006142	#67-1082 79-1757
P.ACF		006156	#67-1086 79-1758
P.ACG		006164	#67-1088 79-1759
P.ACH		006200	#67-1092 79-1760
P.ACI		006206	#67-1094 79-1761
P.ACJ		006212	#67-1096 79-1762
P.ACK		006220	#67-1098 79-1763
P.ACL		006226	#68-1104 79-1764
P.ACM		006232	#68-1106 79-1765
P.ACN		006242	#68-1109 79-1766
P.ACO		006250	#68-1111 79-1767
P.ACP		006260	#68-1114 79-1768
P.ACQ		006264	#68-1116 79-1769
P.ACR		006300	#68-1120 80-1774
P.ACS		006306	#68-1122 80-1775
P.ACT		006316	#68-1125 80-1776
P.ACU		006324	#68-1127 80-1777
P.ACW		006336	#68-1131 80-1778
P.ACX		006350	#68-1135 80-1779
P.ACX		006360	#68-1138 80-1780
P.ACY		006370	#68-1141 80-1781
P.ACZ		006400	#68-1144 80-1782
P.ADA		006406	#68-1146 80-1783
P.ADB		006420	#68-1150 80-1784
P.ADL		006426	#68-1152 80-1785
P.ADD		006436	#68-1155 80-1786
P.ADE		006446	#69-1162 80-1787
P.ADF		006454	#69-1164 80-1788
P.ADG		006462	#69-1166 80-1789
P.ADH		006472	#69-1169 80-1790
P.ADI		006504	#69-1173 80-1791
P.ADJ		006510	#69-1175 80-1792
P.ADK		006526	#69-1180 80-1793
P.ADL		006542	#69-1184 80-1794
P.ADM		006560	#69-1189 80-1795
P.ADN		006576	#69-1194 80-1796
P.ADO		006630	#69-1203 80-1797
P.ADP		006646	#69-1208 80-1798
P.ADQ		006660	#70-1216 80-1799
P.ADR		006672	#70-1220 80-1800
P.ADS		006714	#70-1226 80-1801
P.ADT		006726	#70-1230 80-1802
P.ADU		006740	#70-1234 80-1803
P.ADV		006752	#70-1238 80-1804

SYMBOL	CROSS REFERENCE	VALUE	REFERENCES
P.ADW		006766	#70-1242 80-1805
P.ADX		006774	#70-1244 80-1806
P.ADY		007002	#70-1246 80-1807
P.ADZ		007020	#70-1251 80-1808
P.AEA		007030	#70-1254 80-1809
P.AEB		007036	#70-1256 80-1810
P.AEC		007046	#70-1259 80-1811
P.AED		007062	#70-1263 80-1812
P.AEE		007074	#70-1267 80-1813
P.AEF		007104	#71-1274 80-1814
P.AEG		007124	#71-1280 80-1815
P.AEH		007136	#71-1284 80-1816
P.AEI		007146	#71-1287 80-1817
P.AEJ		007162	#71-1291 80-1818
P.AEK		007174	#71-1295 80-1819
P.AEL		007206	#71-1299 80-1820
P.AEM		007216	#71-1302 80-1821
P.AEN		007230	#71-1306 80-1822
P.AEO		007242	#71-1310 80-1823
P.AEP		007254	#71-1314 80-1824
P.AEQ		007266	#71-1318 80-1825
P.AER		007302	#71-1322 81-1830
P.AES		007312	#72-1329 81-1831
P.AET		007320	#72-1331 81-1832
P.AEU		007326	#72-1333 81-1833
P.AEV		007334	#72-1335 81-1834
P.AEW		007342	#72-1337 81-1835
P.AEX		007350	#72-1339 81-1836
P.AEY		007356	#72-1341 81-1837
P.AEZ		007364	#72-1343 81-1838
P.AFA		007372	#72-1345 81-1839
P.AFB		007400	#72-1347 81-1840
P.AFL		007406	#72-1349 81-1841
P.AFD		007414	#72-1351 81-1842
P.AFE		007422	#72-1353 81-1843
P.AFF		007430	#72-1355 81-1844
P.AFG		007436	#72-1357 81-1845
P.AFH		007444	#72-1359 81-1846
P.AFI		007500	#72-1369 81-1847
P.AFJ		007534	#72-1379 81-1848
P.AFK		007566	#73-1392 81-1849
P.AFL		007622	#73-1402 81-1850
P.AFM		007662	#73-1413 81-1851
RAS.IN		011666	#74-1453 *155-5684 *155-5704 266-12848 334-16689 335-16697
RD.CS1		014244	#100-2888 150-5408 180-7184
RD.DA		014600	#106-3203 150-5418 180-7198
RD.DAT		011672	#74-1455 *100-2903 100-2904 *103-3061 103-3062 *106-3218 106-3219 *109-3376 109-3377
			*112-3535 112-3536 *115-3695 115-3696 *118-3875 118-3876 *122-4061 122-4062 *125-4197
			125-4198 *128-4330 128-4331 *132-4517 132-4518 *136-4707 136-4708 *140-4913 140-4914
			*143-5071 143-5072 166-6305 169-6555 174-6870 181-7250
RD.DS		017514	#143-5056 151-5477
RD.D1		016366	#131-4487 150-5443

SYMBOL	CROSS REFERENCE VALUE	REFERENCES
WRD.12	= 005760	233-10582 236-10785 236-10800 348-17423 353-17730 354-17749 356-17894 359-18017 #79-1741 161-6023 166-6299 169-6549 174-6864 197-8134 198-8187 199-8208 204-8509 205-8562 206-8583 208-8695 211-8915 211-8967 211-8984 211-9094 215-9330 215-9348 218-9524 227-10146 227-10175 227-10220 233-10603 239-11096 253-12136 272-13143 287-14016 295-14471 296-14497 301-14798 305-15025 312-15409 318-15742 326-16150 326-16182 327-16221 366-18402
WRD.13	= 005770	#79-1742 161-6020 216-9368 277-13457
WRD.14	= 005774	#79-1743 162-6079 189-7700 216-9369 267-12875 287-14019 335-16746 336-16800 352-17662 353-17705
WRD.15	= 006006	#79-1744 223-9766 223-9802 224-9838 225-9875 225-9919
WRD.16	= 006014	#79-1745 223-9783 227-10148
WRD.17	= 006022	#79-1746 225-9872 225-9916 353-17729
WRD.18	= 006034	#79-1747 230-10400
WRD.19	= 006040	#79-1748 193-7881 193-7898 193-7915 197-8132 198-8185 199-8206 199-8246 200-8264 200-8293 204-8507 205-8560 206-8581 206-8621 207-8639 207-8668 208-8693 211-8913 211-8962 211-8982 211-9025 211-9039 211-9068 211-9092 214-9271 215-9327 215-9345 216-9370 223-9798 224-9834 227-10108 227-10124 227-10144 227-10173 227-10218 229-10344 229-10360 229-10376 230-10396 233-10561 233-10580 233-10601 241-11426 295-14468 296-14494 305-15022 312-15407 318-15740 325-16144 326-16180 327-16219 331-16462 347-17377 347-17399 348-17422
WRD.2	= 005634	#79-1731 198-8155 198-8172 200-8268 205-8530 205-8547 207-8643 211-8932 211-8949 211-9043 214-9291 215-9314
WRD.20	= 006046	#79-1749 208-8697 211-9096 236-10802
WRD.21	= 006054	#79-1750 236-10784 236-10787 236-10799
WRD.22	= 006062	#79-1751 253-12138 272-13145 295-14474
WRD.23	= 006076	#79-1752 174-6862 295-14470 296-14496 305-15024 326-16153 326-16185 327-16224 331-16464 362-18195 363-18234
WRD.24	= 006104	#79-1753 247-11771 247-11788 247-11809 248-11830 248-11847 301-14801 305-15028
WRD.25	= 006112	#79-1754 301-14800 305-15027
WRD.26	= 006126	#79-1755 359-18019
WRD.27	= 006134	#79-1756 352-17644 353-17732 354-17748 354-17751
WRD.29	= 006142	#79-1757 352-17642
WRD.5	= 005646	#79-1732 193-7902 198-8189 205-8564 211-8969 215-9332 227-10129 229-10364 233-10584
WRD.30	= 006156	#79-1758 356-17893 356-17896
WRD.31	= 006164	#79-1759 352-17663 353-17706
WRD.32	= 006200	#79-1760 247-11774 247-11791 247-11812 248-11833 248-11850
WRD.33	= 006206	#79-1761 247-11772 247-11789 247-11810 248-11831 248-11848
WRD.34	= 006212	#79-1762 247-11776 247-11793 247-11814 248-11835 248-11852
WRD.35	= 006220	#79-1763 239-11094 241-11428 259-12454 368-18552
WRD.36	= 006226	#79-1764 241-11427
WRD.37	= 006232	#79-1765 161-6021 162-6080 181-7244 259-12453 287-14014 289-14137 361-18159
WRD.38	= 006242	#79-1766 181-7245
WRD.39	= 006250	#79-1767 289-14139 326-16152 326-16184 327-16223 331-16461
WRD.4	= 005654	#79-1733 193-7919 199-8210 206-8585 211-8986 215-9350 227-10177 227-10222 230-10384 233-10605
WRD.40	= 006260	#79-1768 239-11167 239-11182
WRD.41	= 006264	#79-1769 342-17126
WRD.42	= 006300	#80-1774 239-11168 239-11183
WRD.43	= 006306	#80-1775 198-8153 198-8170 205-8528 205-8545 211-8930 211-8947 214-9289 215-9312
WRD.44	= 006316	#80-1776
WRD.45	= 006324	#80-1777 239-11095 253-12135
WRD.46	= 006336	#80-1778 239-11170 239-11185 342-17125

SYMBOL	CROSS REFERENCE	VALUE	REFERENCES
WRD.47	=	006350	#80-1779 239-11184
WRD.48	=	006360	#80-1780 272-13141
WRD.49	=	006370	#80-1781
WRD.5	=	005662	#79-1734 186-7513 247-11811 248-11832 248-11849
WRD.50	=	006400	#80-1782 279-13571 312-15411 318-15744 336-16802
WRD.51	=	006406	#80-1783 239-11169 279-13570
WRD.52	=	006420	#80-1784 166-6296 169-6546 174-6861 174-6863 335-16737
WRD.53	=	006426	#80-1785 187-7568
WRD.54	=	006436	#80-1786 239-11171
WRD.55	=	006446	#80-1787 239-11175
WRD.56	=	006454	#80-1788 166-6295 169-6545 174-6860 277-13463 287-14013 335-16735
WRD.57	=	006462	#80-1789
WRD.58	=	006472	#80-1790
WRD.59	=	006504	#80-1791 348-17425
WRD.6	=	005670	#79-1735 186-7533 186-7553 247-11773 247-11790 362-18194 363-18233
WRD.60	=	006510	#80-1792 335-16736 336-16801
WRD.61	=	006526	#80-1793 208-8698 211-9097
WRD.7	=	005676	#79-1736 186-7512 186-7532 186-7552 187-7569 361-18160 362-18193 363-18232
WRD.8	=	005712	#79-1737 186-7551 363-18231
WRD.9	=	005726	#79-1738 186-7511 186-7531 295-14472 296-14498 362-18192
WRT.CS		014174	#98-2791 146-5210 178-7114 179-7146
WRT.DA		014530	#104-3106 146-5220 179-7124 179-7154
WRT.DS		017512	#141-4970 147-5279
WRT.D1		016302	#129-4373 146-5245
WRT.D2		016620	#133-4565 146-5250
WRT.D3		017136	#137-4764 147-5259
WRT.EE		016240	#126-4247 147-5269
WRT.EL		016176	#123-4112 147-5274
WRT.ER		014352	#101-2948 146-5215 178-7117 179-7150
WRT.E1		015272	#113-3579 146-5230 179-7130 179-7162
WRT.E2		015500	#116-3752 146-5235 179-7133
WRT.MR		014706	#107-3263 146-5225
WRT.PA		015064	#110-3419 146-5240 179-7127 179-7158
WRT.PD		015732	#119-3933 147-5264 341-17051
WRT.RE		017622	#145-5183 166-6261 169-6508 173-6795
WT.DAT		011670	#74-1454 *100-2900 100-2904 *103-3058 103-3062 *106-3215 106-3219 *109-3373 109-3377
			*112-3532 112-3536 *115-3692 115-3696 *118-3872 118-3876 *122-4058 122-4062 *125-4196
			125-4198 *128-4329 128-4331 *132-4516 132-4518 *136-4706 136-4708 *140-4910 140-4914
			*143-5068 143-5072 166-6306 169-6556 181-7251
			#74-1451 *155-5683 *155-5703 335-16698 335-16707 336-16787
W.C.SI		011664	#94-2510 295-14443 301-14790
XOR.LN		013354	#7-13
X\$ALWA	=	000000	#7-13
X\$FALS	=	000040	#7-13
X\$OFFS	=	000400	#7-13
X\$TRUE	=	000020	#7-13
\$END.L		065370	G #370-18693
\$PATCH		002720	G #8-327
\$SAVE2		004006	G 37-128 42-371 #45-61 90-2284 98-2791 100-2888 101-2948 103-3046 104-3106
			106-3203 107-3263 109-3361 110-3419 112-3519 113-3579 115-3679 116-3752 118-3858
			119-3933 122-4044 129-4373 131-4487 133-4565 135-4677 137-4764 139-4880 143-5056
			145-5183 149-5381 192-7852 197-8115 204-8490 211-8893 213-9241 236-10745 334-16687
			365-18365 368-18527 370-18713

ML11
SYMBOL CROSS REFERENCE

CREATED BY MACRO ON 23-OCT-80 AT 09:15

PAGE 20
CREF V01

I 15

SEQ 0398

SYMBOL	VALUE		REFERENCES								
SSAVE3	004022	G	12-143	13-191	18-153	19-207	31-183	32-234	#45-68	81-1862	154-5613
			185-7489	189-7677	227-10094	346-17322	370-18712				
SSAVE4	004040	G	#45-76	94-2511	158-5837	165-6241	218-9503	222-9730	351-17612	370-18712	
SSAVE5	004060	G	38-196	42-371	#45-85	161-6001	169-6491	172-6773	178-7102	232-10533	239-11002
			241-11344	245-11687	252-12038	257-12331	264-12704	270-13033	276-13363	285-13868	293-14342
			299-14690	304-14975	310-15271	316-15631	323-16028	330-16404	340-17002	370-18712	
\$T1	021522		#158-5837	159-5900							
\$T10	026334		#192-7852	194-7957							
\$T11	026744		#197-8115	201-8336							
\$T12	030116		#204-8490	208-8729							
\$T13	031364		#211-8893	211-9131							
\$T14	032632		#213-9241	216-9406							
\$T15	033532		#218-9503	219-9572							
\$T16	033770		#222-9730	226-9961							
\$T17	035134		#227-10094	227-10260							
\$T18	036074		#229-10332	230-10424							
\$T19	036470		#232-10533	234-10648							
\$T2	021730		#161-6001	163-6126							
\$T20	037136		#236-10745	236-10828							
\$T21	037474		#239-11002	239-11236							
\$T22	040650		#241-11344	241-11476							
\$T23	041424		#245-11687	249-11903							
\$T24	042550		#252-12038	254-12170							
\$T25	043326		#257-12331	260-12502							
\$T26	044312		#264-12704	267-12903							
\$T27	045424		#270-13033	272-13173							
\$T28	046220		#276-13363	280-13622							
\$T29	047614		#285-13868	290-14181							
\$T3	022416		#165-6241	167-6357							
\$T30	051352		#293-14342	296-14524							
\$T31	052364		#299-14690	302-14859							
\$T32	053320		#304-14975	306-15085							
\$T33	053752		#310-15271	313-15471							
\$T34	055010		#316-15631	319-15795							
\$T35	055716		#323-16028	328-16284							
\$T36	057244		#330-16404	332-16516							
\$T37	057672		#334-16687	337-16840							
\$T38	060664		#340-17002	343-17181							
\$T39	061716		#346-17322	348-17456							
\$T4	023040		#169-6491	169-6610							
\$T40	062470		#351-17612	354-17784							
\$T41	063424		#356-17873	357-17930							
\$T42	063634		#358-17995	359-18043							
\$T43	064020		#361-18156	363-18267							
\$T44	064516		#365-18365	366-18432							
\$T45	065010		#368-18527	368-18613							
\$T5	023476		#172-6773	175-6940							
\$T6	024370		#178-7102	182-7306							
\$T7	025354		#183-7361	183-7387							
\$T8	025426		#185-7489	187-7593							